KIDBUDDY MANAGEMENT SYSTEM

NUR ADLINA NAJWA BINTI RAZALI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

KIDBUDDY MANAGEMENT SYSTEM

NUR ADLINA NAJWA BINTI RAZALI

This report is submitted in partial fulfillment of the requirements for the Bachelor of Computer Science (Database Management) with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

DECLARATION

I hereby declare that this project report entitled

KIDBUDDY MANAGEMENT SYSTEM

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : ______________________________ Date : 13 SEPTEMBER 2024
(NUR ADLINA NAJWA BINTI RAZALI)

I hereby declare that I have read this project report and found this project report is sufficient in term of the scope and quality for the award of Bachelor of Computer Science (Database Management) with Honours.

SUPERVISOR : ______________________________ Date : 13 SEPTEMBER 2024
(DR NUR ATIKAH BINTI ARBAIN)

DEDICATION

This project is dedicated to my beloved parents, whose unwavering support, encouragement, and sacrifices have made this journey possible. Your love, patience, and belief in my abilities have been my greatest motivation, and I am deeply grateful for the countless ways you have supported me throughout this academic endeavor. Your constant encouragement has given me the strength to persevere and strive for excellence, and for that, I am eternally thankful.

To my professors and mentors, your invaluable guidance, wisdom, and dedication to my learning have been instrumental in shaping my academic journey. Your passion for teaching and commitment to nurturing our intellectual growth have inspired me to push the boundaries of my knowledge and skills. The insights and feedback you provided have been crucial in the successful completion of this project, and I am profoundly grateful for your time and effort.

Lastly, to my friends and classmates, this journey would not have been the same without you. Your camaraderie, support, and the countless shared experiences have made this period of my life truly memorable. From late-night study sessions to moments of triumph and laughter, your presence has made this journey not only bearable but enjoyable. Thank you for being there every step of the way and for the friendships that I will cherish forever.

To all of you, this project is a testament to the incredible support system that has surrounded me. Each of you has played a significant role in my academic and personal growth, and I am honored to dedicate this work to you.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to Madam Nur Atikah binti Arbain for her invaluable assistance and guidance in completing this project successfully. Her expertise, patience, and encouragement have been instrumental in shaping this work. From the initial stages to the final touches, her insightful suggestions and constructive criticism have greatly enriched the quality of this project. Her dedication to fostering a supportive learning environment has been truly inspiring, and I am deeply thankful for her unwavering support.

I am profoundly grateful to my beloved parents for their unwavering support and motivation throughout this journey. Their faith in me has been my greatest source of strength, providing me with the confidence to overcome challenges and strive for excellence. Their constant encouragement and sacrifices have made this academic endeavour possible, and I am forever indebted to them for their love and dedication.

Additionally, I would like to extend my heartfelt thanks to my professors and mentors. Your wisdom, insightful feedback, and dedication to my academic growth have been invaluable. Your commitment to teaching and your passion for your respective fields have ignited a similar passion within me. The knowledge and skills I have acquired under your guidance have significantly enhanced the quality of this project, and I am truly grateful for your contributions.

Finally, I would like to thank my friends and classmates for their camaraderie and support. The shared experiences, late-night study sessions, and moments of joy and stress have made this journey memorable. Your friendship and encouragement have been a source of great comfort and motivation, and I am thankful for the bonds we have formed. The collective effort and shared understanding among us have not only made this academic journey bearable but also enjoyable.

To all of you, this project is a testament to the incredible support and guidance I have received. Each of you has played a significant role in my academic and personal growth, and I am honoured to acknowledge your contributions.

ABSTRACT

This project focuses on the development of a comprehensive child day care management system designed to streamline and enhance the operational efficiency of day care centres. The primary problem addressed is the lack of an integrated platform that effectively manages bookings, activity tracking, notifications, and payments. To solve this, the proposed system incorporates modules for registration, login, booking management, activity tracking, an administrative dashboard, a parent portal, a caregiver interface, a staff portal, payment processing, and training management. The development of this project involved requirements gathering, system and database design, implementation, and testing. Key results demonstrate that the system significantly improves communication between parents, caregivers, and administrators, optimizes scheduling and resource allocation, and enhances overall user satisfaction. The system's user-friendly interface and robust functionality make it a valuable tool for day care centres seeking to improve their service delivery and operational workflows. The proposed system is built using HTML for the front-end and PHP for the back end, with MySQL as the database, managed via phpMyAdmin and facilitated by XAMPP software, alongside PL/SQL for complex database operations, including stored procedures, triggers, and events to automate actions such as rejecting overly long booking requests. Additionally, Power BI is used to create an interactive administrative dashboard, by retrieving data from MySQL database. The database backup is performed by storing backup databases into Microsoft Azure. This system has been undergoing a testing phase, during which all functionalities were tested before deployment to end users. User acceptance testing has been conducted with real users through a survey to obtain feedback. The survey is divided into three parts: design interface, system functionality, and workflow of the system. The results of the survey indicate that users are generally satisfied with the system, highlighting the intuitive design interface, reliable functionality, and smooth workflow. Minor suggestions for improvement were noted and addressed, leading to enhancements that further improved the overall user experience.

ABSTRAK

Projek ini memberi tumpuan kepada pembangunan sistem pengurusan jagaan harian kanak-kanak yang komprehensif yang direka untuk menyelaras dan meningkatkan kecekapan operasi pusat jagaan harian. Masalah utama yang ditangani ialah kekurangan platform bersepadu yang menguruskan tempahan, penjejakan aktiviti, pemberitahuan dan pembayaran dengan berkesan. Untuk menyelesaikannya, sistem yang dicadangkan menggabungkan modul untuk pendaftaran, log masuk, pengurusan tempahan, penjejakan aktiviti, papan pemuka pentadbiran, portal induk, antara muka penjaga, portal kakitangan, pemprosesan pembayaran dan pengurusan latihan. Pembangunan projek ini melibatkan pengumpulan keperluan, reka bentuk sistem dan pangkalan data, pelaksanaan dan ujian. Keputusan utama menunjukkan bahawa sistem ini meningkatkan komunikasi antara ibu bapa, penjaga dan pentadbir dengan ketara, mengoptimumkan penjadualan dan peruntukan sumber serta meningkatkan kepuasan pengguna secara keseluruhan. Antara muka mesra pengguna sistem dan fungsi yang mantap menjadikannya alat yang berharga untuk pusat jagaan harian yang ingin meningkatkan penyampaian perkhidmatan dan aliran kerja operasi mereka. Sistem yang dicadangkan dibina menggunakan HTML untuk bahagian hadapan dan PHP untuk bahagian belakang, dengan MySQL sebagai pangkalan data, diuruskan melalui phpMyAdmin dan difasilitasi oleh perisian XAMPP, bersama PL/SQL untuk operasi pangkalan data yang kompleks, termasuk prosedur tersimpan, pencetus, dan acara untuk mengautomasikan tindakan seperti menolak permintaan tempahan yang terlalu lama. Selain itu, Power BI digunakan untuk mencipta papan pemuka pentadbiran interaktif, dengan mendapatkan semula data daripada pangkalan data MySQL. Sandaran pangkalan data dilakukan dengan menyimpan pangkalan data sandaran ke dalam Microsoft Azure. Sistem ini telah menjalani fasa ujian, di mana semua fungsi telah diuji sebelum penggunaan kepada pengguna akhir. Ujian penerimaan pengguna telah dijalankan dengan pengguna sebenar melalui tinjauan untuk mendapatkan maklum balas. Tinjauan dibahagikan kepada tiga bahagian: antara muka reka bentuk, kefungsian sistem dan aliran kerja sistem. Hasil tinjauan menunjukkan bahawa pengguna secara amnya berpuas hati dengan sistem, menyerlahkan antara muka reka bentuk intuitif, fungsi yang boleh dipercayai dan aliran kerja yang lancar. Cadangan kecil untuk penambahbaikan telah diambil perhatian dan ditangani, yang membawa kepada penambahbaikan yang meningkatkan lagi keseluruhan pengalaman pengguna.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

FYP	**-**	**Final Year Project**
SDLC	**-**	**Software Development Life Cycle**
DBLC	**-**	**Database Life Cycle**

Chapter 1: INTRODUCTION

1.1 Introduction

In the modern, fast-paced world, the need for reliable and trustworthy childcare services is more prevalent than ever. Platforms have emerged to meet this demand, connecting parents with caregivers who can provide childcare services on demand. However, there are still challenges to be addressed in this sector, particularly in terms of operational efficiency and communication between all parties involved.

This project focuses on the development of a comprehensive child day care management system designed to streamline the process of finding and booking caregivers, enhancing communication between all parties involved, and improving overall user satisfaction. The system incorporates various modules, including registration, login, booking management, activity tracking, administrative dashboard, parent portal, caregiver interface, staff portal, payment processing, and training management.

The development process involved rigorous stages, including requirements gathering, system design, implementation, and testing. The results demonstrate that the system significantly improves communication between parents, caregivers, and administrators, optimizes scheduling and resource allocation, and enhances overall user satisfaction. With its user-friendly interface and robust functionality, this system is set to be a valuable tool for parents seeking reliable childcare services and caregivers looking to offer their services more efficiently.

1.2 Problem statement(s)

While childcare facilities play a crucial role in supporting working parents by providing a safe and nurturing environment for their children, traditional methods of managing these facilities often fall short in addressing key challenges:

- **Manual processes and communication gaps:** Traditional childcare management systems rely on manual processes and fragmented communication channels, leading to inefficiencies, errors, and communication gaps between caregivers and parents. This lack of streamlined communication can result in misunderstandings, missed updates, and a lack of transparency regarding a child's care.

- **Difficulty in accessing reliable childcare services:** Parents frequently face challenges in finding and booking qualified caregivers, particularly in urgent situations. The lack of a reliable platform for on-demand childcare makes it difficult for parents to ensure their children are cared for when needed, adding stress and uncertainty.

- **Lack of comprehensive reporting:** Many existing childcare management systems do not offer detailed reporting features, limiting administrators' ability to track caregiver performance, booking history, and other key metrics. This absence of data-driven insights hinders the optimization of operations and the improvement of service quality.

1.3 Objectives

This project embarks on the following objectives:

- **Streamline administrative processes:** This objective directly addresses the problem of manual processes and communication gaps by automating administrative tasks. By streamlining processes such as scheduling and billing, the system reduces inefficiencies and errors in childcare facility management.

- **Facilitate on-demand childcare solutions:** Develop a platform that enables parents to easily find and book qualified caregivers for short-term childcare needs. This objective aims to provide convenience and peace of mind to parents, ensuring their children are in safe hands during emergency situations or when immediate care is required.

- **Provide a comprehensive reporting solution:** Implement features that offer detailed reports on caregiver performance, booking history, and administrative activities. This objective focuses on providing data-driven insights to help administrators optimize operations, improve service quality, and make informed decisions.

1.4 Scope

- **User demographics:** The primary users of this system are parents and caregivers. Parents are those who are seeking reliable childcare services, while caregivers are individuals who provide these services. The system is designed to facilitate communication and transactions between these two user groups.

- **Geographical scope:** Initially, the system will cater to users within a specific geographical area (e.g., a city or region). This will allow for a more manageable rollout and the ability to provide localized support and services. However, the system is designed to be scalable and could potentially be expanded to cover larger geographical areas in the future.

- **Platform:** The system will be a web-based platform, accessible via any device with an internet connection and a web browser. This ensures maximum accessibility for users, as they can access the system from home, work, or on the go.

- **Services offered:** The system will offer a range of services, including caregiver booking, activity tracking, payment processing, and communication between parents and caregivers. These services are designed to address the key challenges identified in the problem statement.

- **Training management:** The system will also include a module for training management. This will provide resources and support for caregivers to enhance their skills and knowledge, ultimately leading to improved service quality.

1.5 Project Significance

The significance of this project lies in its potential to transform the way childcare services are managed and delivered.

- **Parents:** Parents will benefit from a streamlined process of finding and booking caregivers. The system's user-friendly interface and robust functionality will provide parents with real-time updates and timely notifications about their child's care, reducing anxiety and uncertainty. The system will also offer a secure platform for processing payments, enhancing the trust and confidence of parents in using the service.

- **Caregivers:** Caregivers will have an efficient platform to offer their services. The system will provide them with a convenient way to manage bookings, communicate with parents, and receive payments. The training management module will also provide resources and support for caregivers to enhance their skills and knowledge, leading to improved service quality.

- **Administrators:** Administrators will gain a comprehensive system to monitor activities and communication logs, ensuring compliance with regulations and maintaining quality standards. The system will also provide valuable insights into the operational efficiency of the service, helping administrators make informed decisions to improve service delivery.

- **Staff:** Staff members manage interactions between parents and caregivers, facilitating bookings and ensuring smooth communication. They also oversee caregiver training programs, scheduling sessions to enhance caregiver skills and knowledge, thereby further improving service quality and satisfaction.

1.6 Expected Output

The expected outputs from this project are as follows:

- **Output 1:** A comprehensive dashboard for parents to find and book caregivers, track activities, and process payments. This dashboard will provide real-time updates and notifications about their child’s care.

- **Output 2:** A user-friendly interface for caregivers to manage bookings, communicate with parents, and receive payments. The system will also provide resources and support for caregivers to enhance their skills and knowledge.

- **Output 3:** An administrative portal for monitoring activities and communication logs, ensuring compliance with regulations, and maintaining quality standards. The portal will also provide valuable insights into the operational efficiency of the service.

- **Output 4:** A training management module that provides resources and support for caregivers to enhance their skills and knowledge, leading to improved service quality.

1.7 Conclusion

In conclusion, this chapter has outlined the background, objectives, problem statement, scope, significance, and expected outputs of the project. The project aims to develop a comprehensive child day care management system that streamlines the process of finding and booking caregivers, enhances communication between all parties involved, and improves overall user satisfaction. The next chapter will delve into the literature review, exploring existing research and technologies related to childcare management systems. This will provide a solid foundation for the subsequent design and development of the system.

CHAPTER 2: PROJECT METHODOLOGY AND PLANNING

2.1 Introduction

This chapter outlines the project methodology and planning, focusing on the Database Life Cycle (DBLC) phases related to the project. The DBLC is a series of steps, followed by database designers to create and maintain a database. The methodology will guide the development process, ensuring that the project is systematic, organized, and well-documented.

2.2 Project Methodology

The development of the child day care management system followed a structured methodology rooted in the principles of both the Software Development Life Cycle (SDLC) and Database Life Cycle (DBLC). The project began with a thorough requirement gathering phase, engaging stakeholders such as administrators, parents, caregivers, and staff to understand the operational needs surrounding booking management, activity tracking, notifications, and payments. This initial phase laid the groundwork for the system design, where detailed architectural diagrams and database schemas were created using MySQL with phpMyAdmin, emphasizing user-friendly interfaces tailored for different user roles.

During the implementation phase, the system was developed using PHP and JavaScript, with PL/SQL integrated for database automation and complex operations. The database design focused on creating normalized schemas, ensuring data integrity and efficient query performance. Comprehensive testing was conducted, including unit, integration, and user acceptance testing, to validate the system's reliability, security, and performance. Additionally, database testing ensured data accuracy and consistency, with ongoing maintenance involving regular backups and performance tuning managed through Microsoft Azure. This approach, combining SDLC and DBLC, ensured that both the application and data management aspects of the system were robust, scalable, and aligned with the project’s objectives.

2.2.1 Database Initial Study

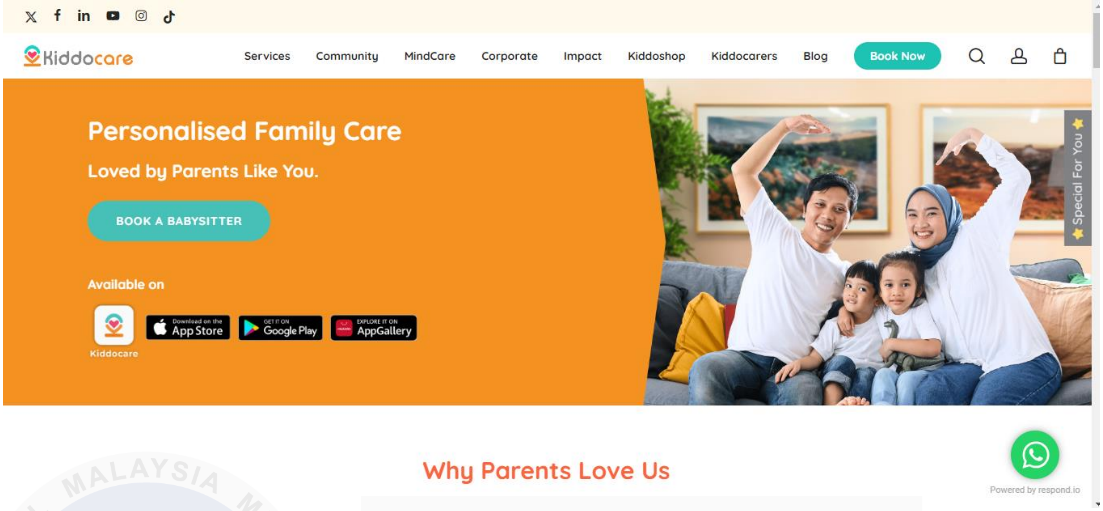


Figure 2.1: Kiddocare Homepage

Kiddocare is an online platform designed to connect parents with qualified caregivers for childcare needs. It offers a range of services including babysitting, nanny placements, and daycare options. Parents can search for caregivers based on criteria such as location, availability, and specific childcare requirements. Kiddocare provides profiles of caregivers with reviews and ratings, facilitating informed decision-making. The platform aims to simplify the process of finding reliable childcare solutions through its user-friendly interface and secure communication tools, ensuring peace of mind for parents.

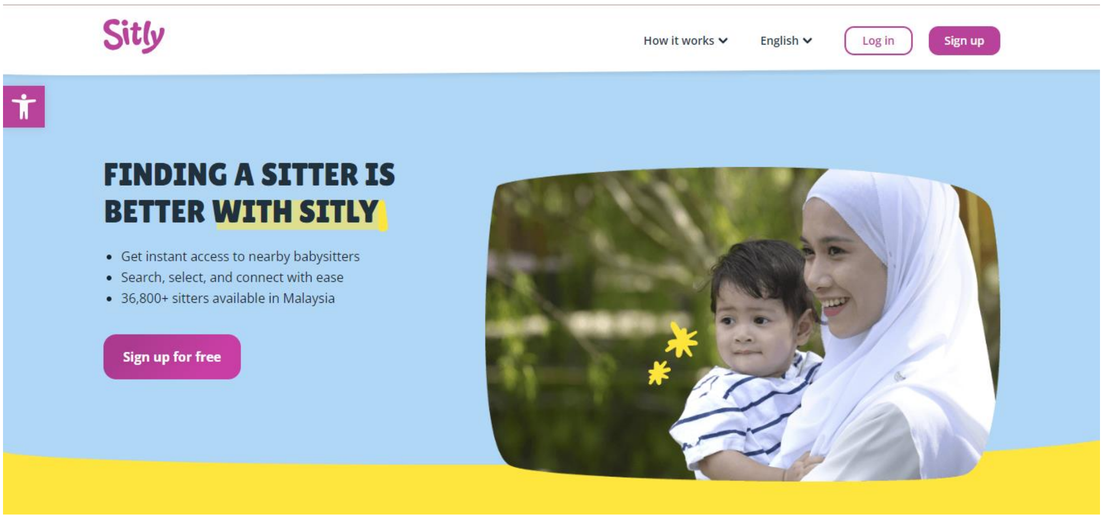

Sitly
How it works
English
Log in
Sign up
FINDING A SITTER IS BETTER WITH SITLY
Get instant access to nearby babysitters
Search, select, and connect with ease
36,800+ sitters available in Malaysia
Sign up for free

Figure 2.2 : Sitly Homepage

Sitly is a service connecting parents with babysitters and childminders. It operates as a matching platform where parents can find caregivers based on location, availability, and specific care preferences. With caregiver profiles detailing experience and reviews, Sitly facilitates direct communication between parents and potential caregivers through its intuitive interface and mobile app, ensuring convenience and safety in childcare arrangements.

2.2.2 Database Design

The second phase of this project focuses on the design of the database model to support the operational requirements and objectives of the childcare management system. This phase comprises several critical sub-phases: conceptual design, logical design, and physical design. In the conceptual design stage, an entity-relationship diagram (ERD) is crafted, and the normalization process ensures the ERD aligns effectively with the system requirements. Moving to the logical database design, definitions and business logic are refined and validated, culminating in the development of a relational data model that specifies data structures and queries based on system requirements. Finally, the physical design phase translates the logical model into concrete implementation, detailing tables, columns, indexes, sequences, and constraints. This phase organizes attributes from the logical database model to accurately represent core business rules and establish comprehensive data relationships essential for the system's functionality and performance.

2.2.3 Implementation and Loading

The implementation and loading phase of this project involves the actual development and deployment of the childcare management system. It encompasses several key activities aimed at translating the designed database and system requirements into a functional application. During implementation, the system's components, including modules for registration, login, booking management, activity tracking, administrative dashboard, parent portal, caregiver interface, staff portal, payment processing, and training management, are developed according to the specifications defined in the previous phases. This phase includes coding, debugging, and integration of various system functionalities to ensure they operate seamlessly. Additionally, data migration and loading processes are executed to populate the database with initial data, ensuring the system is ready for operational use. Throughout this phase, rigorous testing and quality assurance procedures are conducted to validate the system's performance, functionality, and adherence to requirements before proceeding to the next stages of deployment and evaluation.

2.2.4 Testing and Evaluation

The testing and evaluation phase is crucial in ensuring the reliability, functionality, and performance of the childcare management system. Comprehensive testing methodologies, including unit testing, integration testing, system testing, and user acceptance testing, are employed to identify and resolve any issues or bugs. This phase validates that all system components, from registration and login processes to booking management, notifications, and payment processing, perform as intended and meet specified requirements. User feedback and testing results are analyzed to refine and optimize the system's performance and user experience. Evaluation criteria include usability, efficiency, scalability, and security to ensure the system's readiness for deployment and operational use.

2.2.5 Operation

The operation phase marks the deployment and commencement of the childcare management system's live operation. During this phase, the system is launched into production following successful testing and evaluation. Operational activities include system monitoring, user support, and ensuring continuous availability and performance. Users, including parents, caregivers, and administrative staff, engage with the system to utilize its functionalities such as booking childcare services, tracking activities, making payments, and accessing reports. Monitoring tools and procedures are implemented to detect and address any operational issues promptly, ensuring uninterrupted service delivery and user satisfaction.

2.2.6 Maintenance and Evolution

Maintenance and evolution focus on sustaining and enhancing the childcare management system post-deployment. Regular maintenance activities, including updates, patches, and bug fixes, are performed to address software issues, security vulnerabilities, and user feedback. Additionally, the system evolves through planned enhancements and feature updates to adapt to changing user needs, technological advancements, and regulatory requirements. This phase involves collaboration between developers, stakeholders, and end-users to prioritize and implement system improvements while ensuring minimal disruption to ongoing operations. Continuous evaluation and feedback loops inform iterative development cycles aimed at optimizing system performance, expanding functionality, and enhancing overall user satisfaction and system longevity.

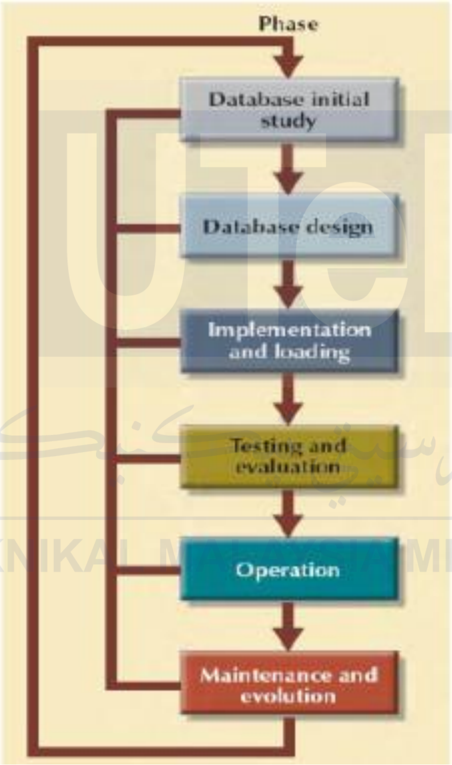


Figure 2.3 : Database Life Cycle

2.3 Project Schedule and Milestone

Table 2.1 : Gantt Chart

Activities	Month (2024)						
	3	4	5	6	7	8	9
Database Initial Study	🟩						
Database Design		🟩	🟩				
Implementation and Loading			🟩	🟩	🟩		
Testing and Evaluation						🟩	
Operation						🟩	
Maintenance and Evolution							🟩

The table 2.1 above outlines the project timeline for the development and deployment of a database system over several months in 2024. The project begins in March with a database initial study, where the requirements and scope of the database are thoroughly examined to ensure all necessary data and functionalities are considered. This phase continues into April.

Following this, the database design phase takes place in April and May. During this time, the database's structure is carefully designed, including the creation of tables, relationships, and schema definitions to support the system's required functionalities.

From May to July, the project enters the implementation and loading phase. In this phase, the database is built and populated with initial data, and the necessary configurations are made to ensure the system is ready for operation.

The testing and evaluation phase runs from June to August. During this period, the system undergoes rigorous testing to verify that it functions as intended and meets user requirements. Any issues discovered during testing are addressed to refine the system.

In July and August, the database moves into the operation phase, where it becomes fully operational. Users begin using the system, and any final adjustments are made to ensure smooth operation.

Finally, the maintenance and evolution phase start in August and continues beyond. This phase involves ongoing support, updates, and enhancements to ensure the database remains effective and up to date with evolving user needs and technological advancements.

2.4 Conclusion

In conclusion, the development of the childcare management system has followed a structured approach guided by the Database Life Cycle (DBLC), ensuring systematic planning, design, implementation, testing, deployment, and maintenance. The project addressed critical operational needs such as booking management, activity tracking, notifications, and payments through thorough requirements gathering and robust system design. Utilizing MySQL with phpMyAdmin for database management and PHP with JavaScript for frontend and backend development, the system integrates PL/SQL for automation and Power BI for data visualization. Testing methodologies validated system reliability and security, leading to a successful deployment that enhances communication among parents, caregivers, and administrators while optimizing service delivery and operational workflows. Ongoing maintenance and evolution strategies ensure the system's adaptability, responsiveness to user feedback, and continuous improvement, positioning it as a valuable tool for childcare centers seeking enhanced efficiency and user satisfaction.

CHAPTER 3: ANALYSIS

3.1 Introduction

This chapter delves into the analysis phase of the project, providing a detailed examination of the current system, proposed improvements, and requirements of the to-be system. The analysis will be developed through a systematic approach, ensuring a thorough understanding of the system’s needs and potential solutions.

3.2 Problem Analysis

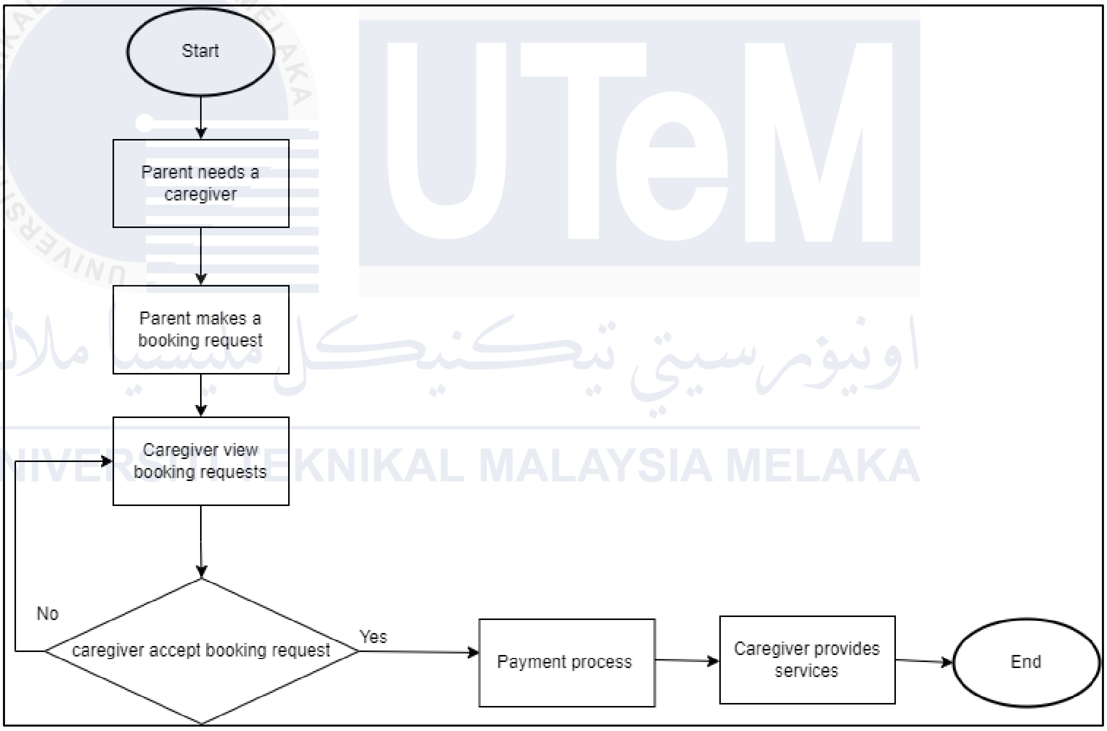

Start
Parent needs a caregiver
Parent makes a booking request
Caregiver view booking requests
caregiver accept booking request
No
Yes
Payment process
Caregiver provides services
End

Figure 3.1 : Flowchart of current system

Based on figure 3.1, the process begins when a parent identifies the need for a caregiver. This need triggers the parent to make a booking request, which could involve calling, emailing, or visiting different childcare facilities. Once the booking request is made, caregivers view these requests. At this point, a decision is made by the caregivers. If a caregiver accepts the booking request, the process moves forward. If not, the process loops back, and other caregivers view the booking request. Once a caregiver accepts the booking request, the parent makes a payment. This payment could be made in person, via bank transfer, or through another manual method. After the payment process, the caregiver provides the service, taking care of the child as per the booking. This marks the end of the process. Throughout this process, there are several manual steps and decision points, which can lead to inefficiencies and communication gaps. The proposed system aims to automate these steps, improving efficiency and communication between parents, caregivers, and administrators.

3.3 The proposed improvement / solutions

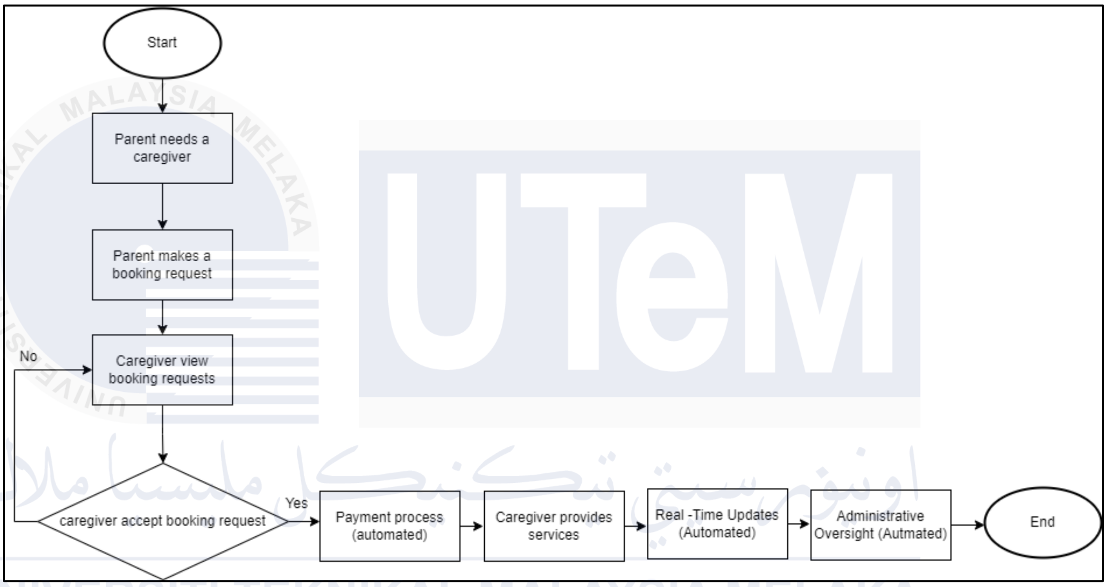

Start
Parent needs a caregiver
Parent makes a booking request
Caregiver view booking requests
No
caregiver accept booking request
Yes
Payment process (automated)
Caregiver provides services
Real -Time Updates (Automated)
Administrative Oversight (Autmated)
End

Figure 3.2 : Flowchart of Proposed System

As shown in Figure 3.2, the process begins when a parent identifies the need for a caregiver. This need triggers the parent to make a booking request using the system's automated booking function. Once the booking request is made, it becomes visible to all caregivers who can view these requests. At this point, a decision is made by the caregivers. If a caregiver accepts the booking request, the process moves forward. If not, the process loops back, and other caregivers can view and accept the booking request. Once a caregiver accepts the booking request, the parent makes a payment using the system's secure and automated payment processing function. After the payment process, the caregiver provides the service, taking care of the child as per the booking. During the service, the system sends real-time updates and notifications to the parent about their child's activities and well-being. This ensures that parents are always informed about their child's status. In addition, the system provides administrators with tools to track activities and interactions, ensuring compliance with regulations and maintaining quality standards. This marks the end of the process. The proposed system aims to automate these steps, improving efficiency and communication between parents, caregivers, and administrators.

3.4 Requirement analysis of the to-be system

The requirement analysis will outline both the system requirements for each user and the functional needs.

3.4.1 Functional Requirement (Process Model)

- **User Registration and Login:** The system will facilitate a seamless registration process for both parents and caregivers. During registration, parents will provide essential information such as their name, email, phone number, and address, which the system will use to manage their interactions and booking activities with caregivers. On the other hand, caregivers will also provide their name, email, phone number, and relevant qualifications or certifications. They may additionally set their availability and agree to the platform's terms and conditions during registration. This information is securely stored within the system to ensure user data protection. Once registered, users can log in using their email and password, with the system verifying their credentials against the stored data. Upon successful login, users will gain access to their respective interfaces, enabling parents to manage bookings and caregivers to handle their services efficiently.

- **Booking Management:** The system allows parents to easily search for available caregivers by filtering criteria such as availability, location, and caregiver qualifications. Once a suitable caregiver is found, parents can initiate a booking request, specifying details like the date, time, and duration of care required. The system will then notify the selected caregiver, who can accept or decline the booking based on their availability. Upon acceptance, the booking is confirmed, and both parties receive notifications detailing the care arrangement. The system also allows parents to view the status of their booking requests and provides caregivers with an organized view of upcoming commitments, ensuring clear communication and minimizing scheduling conflicts. This module also tracks booking history, allowing parents to easily rebook a caregiver and enabling caregivers to manage their schedules effectively.

- **Training Management:** The system will include a dedicated training management module designed to track and document the training activities of caregivers. This module will allow caregivers to enroll in mandatory training sessions that are necessary for their verification and qualification on the platform. The system will record the details of each training session, including the caregiver's attendance and completion status. Administrators can access this information to ensure that all caregivers meet the required standards and that their training certifications are up to date.

- **Payment Processing:** The system will streamline the payment process by calculating the total cost of services based on the booking details, such as the duration and type of care provided. Once a booking is completed, the system will automatically process the payment from the parent to the caregiver through a secure payment gateway. The platform may also deduct a service fee or commission as per its policies. The system will ensure transparency by providing detailed payment receipts to both parents and caregivers, which can be accessed through their respective dashboards. Additionally, the system will track all payment transactions, allowing users to review their payment history and ensuring that all financial interactions are accurately recorded for future reference.

- **Administrative Oversight**: The system will offer a comprehensive administrative portal that allows administrators to monitor all activities within the platform. This includes tracking user registrations, bookings, training sessions, and payment transactions. The system will maintain detailed logs of all interactions and activities, enabling administrators to ensure compliance with regulations and uphold quality standards. The administrative portal will also provide tools for generating reports and analytics, offering insights into operational efficiency and helping administrators make informed decisions to improve service delivery. By centralizing oversight functions, the system empowers administrators to maintain control over the platform’s operations, ensuring that both parents and caregivers receive the highest level of service.

3.4.1.1 Data Flow Diagram (DFD)

1. **Context Diagram**

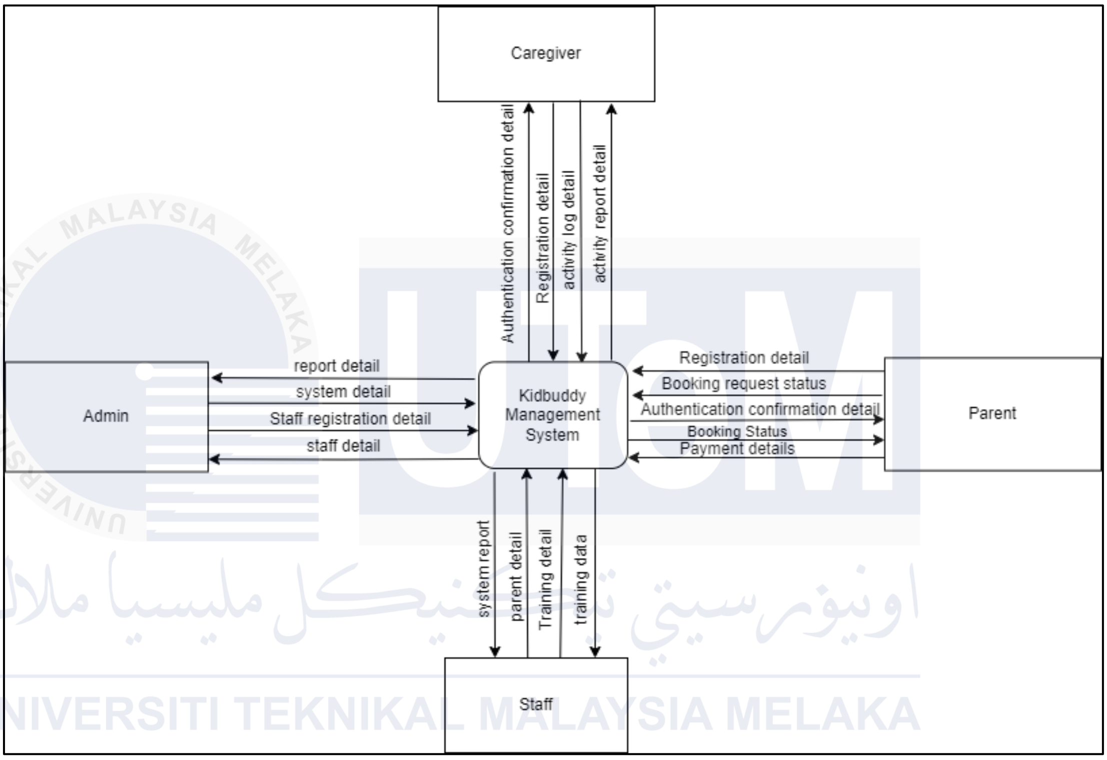


Figure 3.3 : Context Diagram

As shown in figure 3.3, the context diagram provides a high-level overview of the interactions between the Kidbuddy management system and its external entities, including Parents, Caregivers, Staff, and Admin. The diagram depicts how data flows between the system and these key stakeholders, illustrating the essential inputs and outputs that facilitate the system's operation.

Parents primarily engage with the Kidbuddy management system for tasks such as registration, booking requests, and payment processing. They provide the system with their registration details and booking requests, while the system, in turn, delivers authentication confirmations, updates on booking statuses, and payment details back to the parents. This interaction ensures that parents can effectively manage their childcare needs through the system.

Caregivers interact with the system by submitting their registration details and activity logs. The system confirms their authentication and provides them with activity reports, thereby enabling caregivers to keep track of their engagements and responsibilities. This exchange of information is crucial for maintaining the accuracy and reliability of the caregiver's activities within the system.

Staff members use the system to manage parent details, training schedules, and generate system reports. The system sends them the necessary training details and schedules, while staff input updated parent and training information, ensuring that all data within the system remains current and accurate. This interaction supports the system's ability to manage and coordinate various activities effectively.

The admin users have a supervisory role, interacting with the system to input staff registration details and authenticate staff members. In return, they receive comprehensive system details and various reports that allow them to monitor and manage the overall operation of the Kidbuddy Management System. This interaction is vital for the administration to maintain oversight and ensure that the system operates smoothly and efficiently.

In summary, the context diagram highlights the essential data exchanges between the Kidbuddy Management System and its external entities, showcasing how parents, caregivers, staff, and administrators are connected through this centralized platform to facilitate the system's effective functioning.

2. Data Flow Diagram (DFD) Level 1

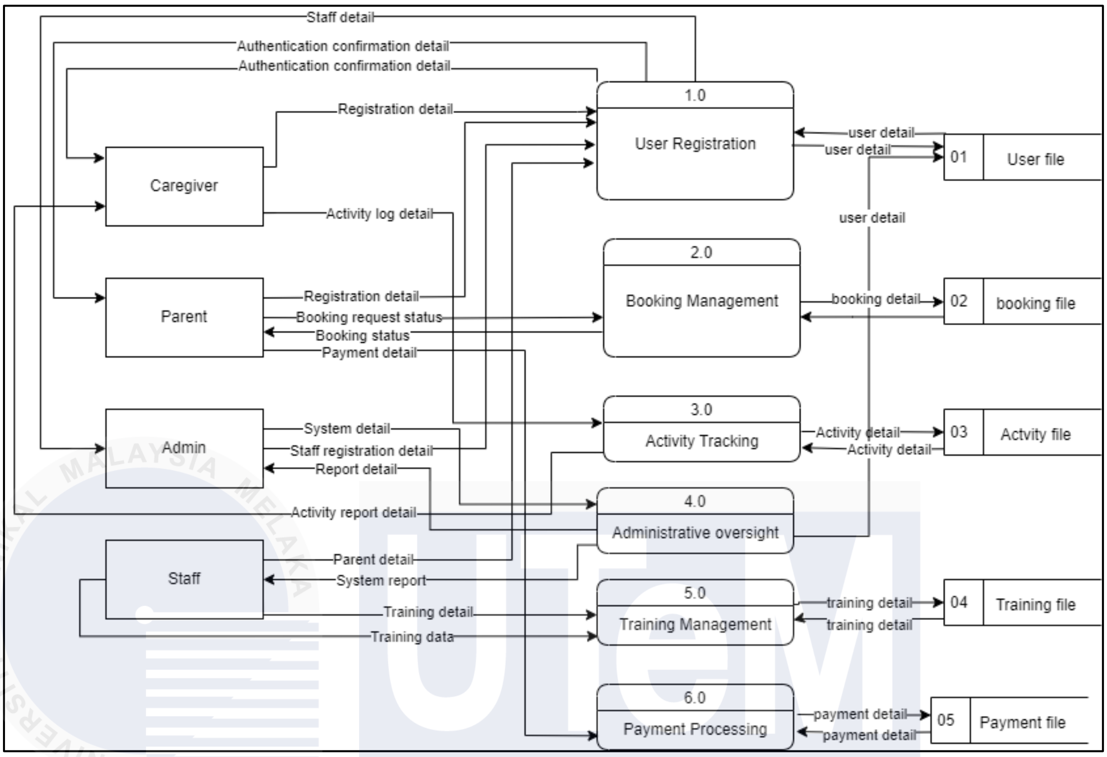


Figure 3.4 : Data Flow Diagram (DFD) Level 1

Based on figure 3.4, the diagram illustrates a system architecture designed to manage various operations for users, including caregivers, parents, admins, and staff. It is structured around six interconnected modules. The user registration module is central to the system, handling the initial registration of all user types. It collects and processes user details, storing them securely in a user file. Following registration, the booking management module comes into play, particularly for parents and caregivers. This module oversees the entire booking process, allowing parents to make requests, confirm bookings, and track status updates. All relevant booking information is recorded in a booking file for future reference.

Parallel to these, the activity tracking module is responsible for monitoring and recording the activities of users across the system. Whether it's logging the availability of caregivers or tracking user interactions, this module ensures that all activities are meticulously documented in an activity file. This tracking supports the administrative oversight module, which provides admins with the tools they need to monitor user activities and generate necessary reports for system governance.

The system also includes a training management module dedicated to staff training activities. It schedules and records training sessions, ensuring that all training details are preserved in a training file. Lastly, the payment processing module handles financial transactions within the system, such as payments for bookings and training sessions. It manages payment details and confirmations, storing them in a payment file.

3.4.2 Non-functional requirement

Non-functional requirements define the qualities and attributes of a system that ensure its overall effectiveness, usability, and reliability. These requirements are critical for delivering a system that not only functions correctly but also meet user expectations in terms of performance, security, and user experience. The non-functional requirements for the Kidbuddy Management System can be categorized into two main areas which is quality requirements and performance requirements.

3.4.2.1 Quality Requirements:

- **Accuracy:** The system's accuracy is critical in ensuring the integrity of its operations. It must accurately record and process all user data, including registration details, booking information, training records, and payment transactions. For instance, when a parent books a caregiver, the system should precisely capture the details such as date, time, duration, and specific requirements without any errors. Similarly, payment calculations should be accurate, reflecting the correct service charges, taxes, and any applicable fees. This precision ensures that users can trust the system to handle sensitive and important information, minimizing the risk of disputes due to incorrect data handling. To achieve this, the system should incorporate robust validation mechanisms that check for data consistency and completeness before recording any information.

- **Security:** The security of user data is paramount, especially when dealing with personal information and financial transactions. The system must implement secure login mechanisms, such as two-factor authentication (2FA), to prevent unauthorized access. User passwords should be stored using strong encryption techniques, and sensitive data, such as payment information, should be transmitted over encrypted channels (e.g., using SSL/TLS). Additionally, the system should be designed to protect against common security threats such as SQL injection, cross-site scripting (XSS), and data breaches. Regular security audits and updates should be conducted to ensure that the system remains resilient against evolving threats. By prioritizing security, the system can foster trust among users, ensuring that their personal and financial information is protected.

- **Reliability:** The system's reliability is essential for maintaining user confidence and ensuring continuous operation. It should be designed to handle high availability with minimal downtime, meaning that users should be able to access the system and perform their tasks whenever needed. This requires the system to be built on a stable and scalable architecture that can handle varying loads without crashing or slowing down. Additionally, the system should include a robust backup and recovery mechanism. In case of any failures, such as server crashes or data corruption, the system should be able to quickly recover without data loss. Automated backups should be scheduled regularly, and recovery processes should be tested to ensure they work effectively in a real-world scenario. By ensuring reliability, the system can minimize disruptions and maintain seamless user experiences.

- **Usability:** Usability focuses on the user experience, ensuring that the system is easy to use for all types of users, including parents, caregivers, and administrators. The interface should be intuitive, with a clean and logical layout that guides users through the system's features effortlessly. For example, the booking process should be straightforward, with clear instructions and minimal steps to complete a booking. Navigation should be consistent across the platform, allowing users to easily find the features they need without getting lost or confused. Additionally, the system should offer responsive design, meaning it should be fully functional and visually appealing on various devices, including smartphones, tablets, and desktops. User feedback should be collected and analyzed to continuously improve the system's usability, addressing any pain points and ensuring that the platform remains user-friendly as new features are added. By focusing on usability, the system can ensure that users, regardless of their technical proficiency, can efficiently accomplish their tasks.

3.4.2.2 Performance Requirements:

- **Response Time:** The system's response time is a critical aspect of user experience, directly impacting how users perceive the efficiency and reliability of the platform. The system should be designed to respond quickly to user queries and actions, minimizing any delays that could frustrate or inconvenience users. For instance, when a parent makes a booking request, the system should process this request instantly, updating the status and notifying the caregiver in real-time. Similarly, when users navigate through different sections of the platform or perform tasks such as logging in, submitting forms, or generating reports, these actions should be completed within seconds. Achieving optimal response times requires efficient coding practices, optimized database queries, and the use of caching strategies to reduce server load and accelerate data retrieval. By ensuring that the system responds quickly, the platform can maintain a smooth and seamless user experience, which is essential for user satisfaction and retention.

- **Scalability:** Scalability refers to the system's ability to grow and adapt to increased demand over time. As the user base expands or as more features are added to the platform, the system should be able to scale up without compromising performance. This could involve handling a larger number of simultaneous users, processing more data, or integrating with additional services. To achieve scalability, the system should be built on a flexible architecture that allows for easy expansion, such as using cloud-based infrastructure that can dynamically allocate resources based on demand. Additionally, modular design principles should be employed, allowing new components or services to be added without disrupting existing functionality.

Scalability also includes the ability to efficiently manage increased data traffic, such as more frequent booking requests or larger volumes of training data. By designing for scalability, the system ensures that it can support future growth, accommodating more users, features, and data without degradation in performance.

- **Capacity:** Capacity is the system's ability to handle the expected volume of data and user load both at present and in the future. This includes the capability to store and manage large amounts of data, such as user accounts, booking requests, activity logs, and training records. The system should be designed to efficiently handle this data, with a database architecture that supports rapid read and write operations even under heavy load. Furthermore, the system should be capable of supporting multiple users accessing the platform simultaneously without performance issues. This involves ensuring that the server infrastructure can handle concurrent sessions, preventing slowdowns or crashes during peak usage times. Load testing should be conducted regularly to identify and mitigate potential bottlenecks. Additionally, the system should be equipped with sufficient storage and processing power to manage large datasets, with provisions for scaling up as data volumes increase. By ensuring adequate capacity, the system can maintain high performance and reliability even as the user base and data volumes grow.

3.4.3 Other Requirement

Other requirements in a project specification include various criteria beyond the software and hardware needs. These often cover performance metrics, such as system response time and efficiency, as well as security measures for protecting data and ensuring safe access. Compliance with relevant regulations and standards is crucial, alongside usability requirements for ease of use and accessibility. Reliability and availability expectations address system uptime and fault tolerance, while maintainability criteria ensure the system can be easily updated and repaired. Compatibility with other systems and software, along with the need for adequate training and documentation, also falls under this category.

3.4.3.1 Software Requirements

Table 3.1 : Table Software Requirements

Software	Benefit	Reason
Microsoft Visual Studio Code	Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript and has a rich ecosystem of extensions for other languages	Visual Studio Code provides a robust and convenient environment for coding, debugging, and version control
XAMPP	XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, and interpreters for scripts written in the PHP and Perl programming languages.	XAMPP provides an easy-to-use control panel to manage your Apache Server, MySQL databases, and PHP scripts.
Microsoft Azure	Microsoft Azure is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through Microsoft-managed data centers.	Microsoft Azure provides a range of cloud services, including those for computing, storage and networking.
Power BI	Power BI is a powerful business analytics tool developed by	Power BI provides interactive

Power BI	Microsoft. It enables users to visualize and analyze data with enhanced capabilities for creating interactive reports and dashboards.	visualizations and deep data analysis capabilities for insightful decision-making.

3.4.3.2 Hardware Requirements:

Table 3.2 : Table Hardware Requirements

Hardware	Specification	Reason
Laptop HP	Windows 11 Home Single Language	Windows 11 is the latest operating system from Microsoft, offering improved performance, security, and a user-friendly interface
	Intel(R) Core (TM) i3-8130U CPU @ 2.20GHz 2.21 GHz	The Intel Core i3 processor provides sufficient processing power for developing and testing your system. It can handle multiple tasks and processes without lagging, ensuring a smooth development experience.
	4.00 GB (3.87 GB usable)	4GB of RAM is enough for running the necessary software for development like Visual Studio Code, XAMPP, and web browsers for testing. It allows for efficient multitasking and quick access to data and applications.

	64-bit operating system, x64-based processor	A 64-bit operating system can handle large amounts of memory more efficiently than a 32-bit operating system. This is beneficial for running applications that require a lot of memory, such as database management systems and web servers.

3.5 Conclusion

In conclusion, this chapter has analyzed the current childcare management system, identified key challenges, and proposed a new system to address these issues. The proposed system's functional and non-functional requirements have been outlined, focusing on automating processes, enhancing communication, and improving oversight. The software and hardware requirements for the system have been specified, providing a solid foundation for the design and development of the system. The proposed system aims to improve the efficiency and effectiveness of childcare services, benefiting parents, caregivers, and administrators alike.

CHAPTER 4: DESIGN

4.1 Introduction

This chapter outlines the architecture and database design for the child day care management system. It provides an in-depth look at the system's architecture, including layers, frameworks, and design patterns, followed by a detailed discussion of the database design process, encompassing conceptual, logical, and physical design phases. The chapter concludes with the design and explanation of the graphical user interface (GUI), ensuring alignment with the functional and non-functional requirements.

4.2 System Architecture

The architecture view of the child day care management system is designed to ensure scalability, maintainability, and efficiency. The system adopts a three-tier architecture comprising the presentation layer, business logic layer, and data layer.

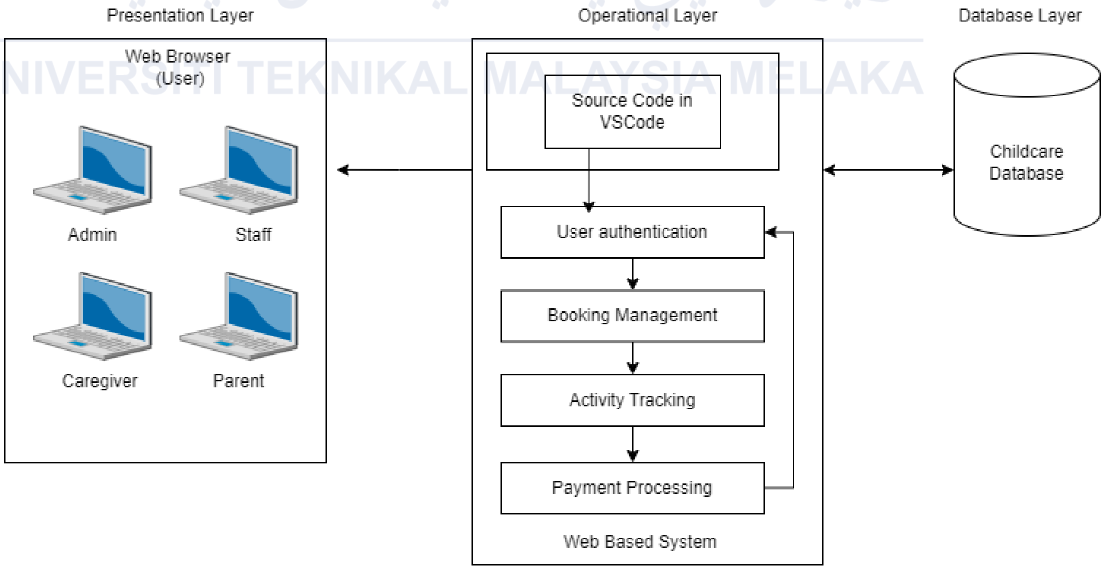


Figure 4.1 : Kidbuddy System Architecture

- Presentation Layer: This layer includes the user interfaces for different users (parents, caregivers, administrators, and staff). It is developed using HTML, CSS, and JavaScript, providing a responsive and intuitive user experience.

- Operational Layer: This layer contains the core functionalities of the system, including user authentication, booking management, activity tracking, payment processing, and training management. It is implemented using PHP and integrated with PL/SQL for database operations.

- Data Layer: This layer is responsible for data storage and management. MySQL with phpMyAdmin is used for database management, ensuring data integrity, security, and efficient data retrieval.

4.3 Database Design

The database design process involves three main phases: conceptual design, logical design, and physical design. Each phase aims to ensure the database structure supports the system's operational requirements and objectives.

4.3.1 Conceptual Design

4.3.1.1 Normalization

1. First Normal Form (1NF)

- In INF, the data might be stored in a large table where some attributes contain repeating groups or sets of values.

- Repeating Groups**:** ChildInfo, BookingInfo, TrainingInfo, and PaymentInfo contain multiple values in a single cell.

Table 4.1 : 1NF Caregiver Table

CaregiverID	CarerName	CarerEmail	ChildID	BookingID	TrainingID	PaymentID

2. First Normal Form (1NF) to Second Normal Form (2NF)

a. Identifying Dependencies in 1NF:

- **Composite Key**: In 1NF, CaregiverID and ChildID together can be used as a composite key in some cases (e.g., identifying which child is cared for by which caregiver).

- **Partial Dependency:** A partial dependency occurs when a non-key attribute is dependent on part of the composite primary key.

b. Conversion to Second Normal Form (2NF):

- Remove partial dependencies by splitting the table into smaller tables where non-key attributes depend on the entire primary key.

- Create new tables to eliminate partial dependencies.

i. Caregiver Table

Table 4.2 : 2NF Caregiver Table

CaregiverID	CarerName	CarerEmail

ii. Child Table

Table 4.3 : 2NF Child Table

ChildID	ChildName	ChildDOB	ChildAllergies	ParentID

iii. Booking Table

Table 4.4 : 2NF Booking Table

BookingID	ChildID	CaregiverID	Booking Date	Booking Time	Transportation	BookingStatus

iv. Training Table

Table 4.5 : 2NF Training Table

TrainingID	CaregiverID	TrainingName	TrainingDate	TrainingStatus

v. Payment Table

Table 4.6 : 2NF Payment Table

PaymentID	BookingID	Amount	PaymentMethod	PaymentStatus	PaymentDate

- **No Partial Dependencies:** All non-key attributes are fully dependent on the entire primary key.

- **Smaller Tables:** The data is organized into multiple tables to ensure that partial dependencies are removed.

3. Second Normal Form (2NF) to Third Normal Form (3NF)

a. Identifying Transitive Dependencies in 2NF:

- Transitive Dependency: A transitive dependency occurs when a non-key attribute depends on another non-key attribute rather than directly on the primary key.

- For example, we had a ParentAddress in the Child table, it would be a transitive dependency since ParentAddress depends on ParentID, which is a non-key attribute.

vi. Caregiver Table

Table 4.7 : 3NF Caregiver Table

CaregiverID	CarerName	CarerEmail

vii. Child Table

Table 4.8 : 3NF Child Table

ChildID	ChildName	ChildDOB	ChildAllergies	ParentID

viii. Parent Table

Table 4.9 : 3NF Parent Table

ParentID	ParentName	ParentEmail	ParentPhone	ParentAddress

ix. Booking Table

Table 4.10 : 3NF Booking Table

BookingID	ChildID	CaregiverID	Booking Date	Booking Time	Transportation	BookingStatus

x. Training Table

Table 4.11 : 3NF Training Table

TrainingID	CaregiverID	TrainingName	TrainingDate	TrainingStatus

xi. Payment Table

Table 4.12 : 3NF Payment Table

PaymentID	BookingID	Amount	PaymentMethod	PaymentStatus	PaymentDate

- **No Transitive Dependencies:** All attributes are directly dependent on the primary key.

- Based on the results of the normalization process, it can be concluded that the ERD is free of transitive dependencies and is well-structured.

4.3.1.2 Entity Relationship Diagram (ERD)

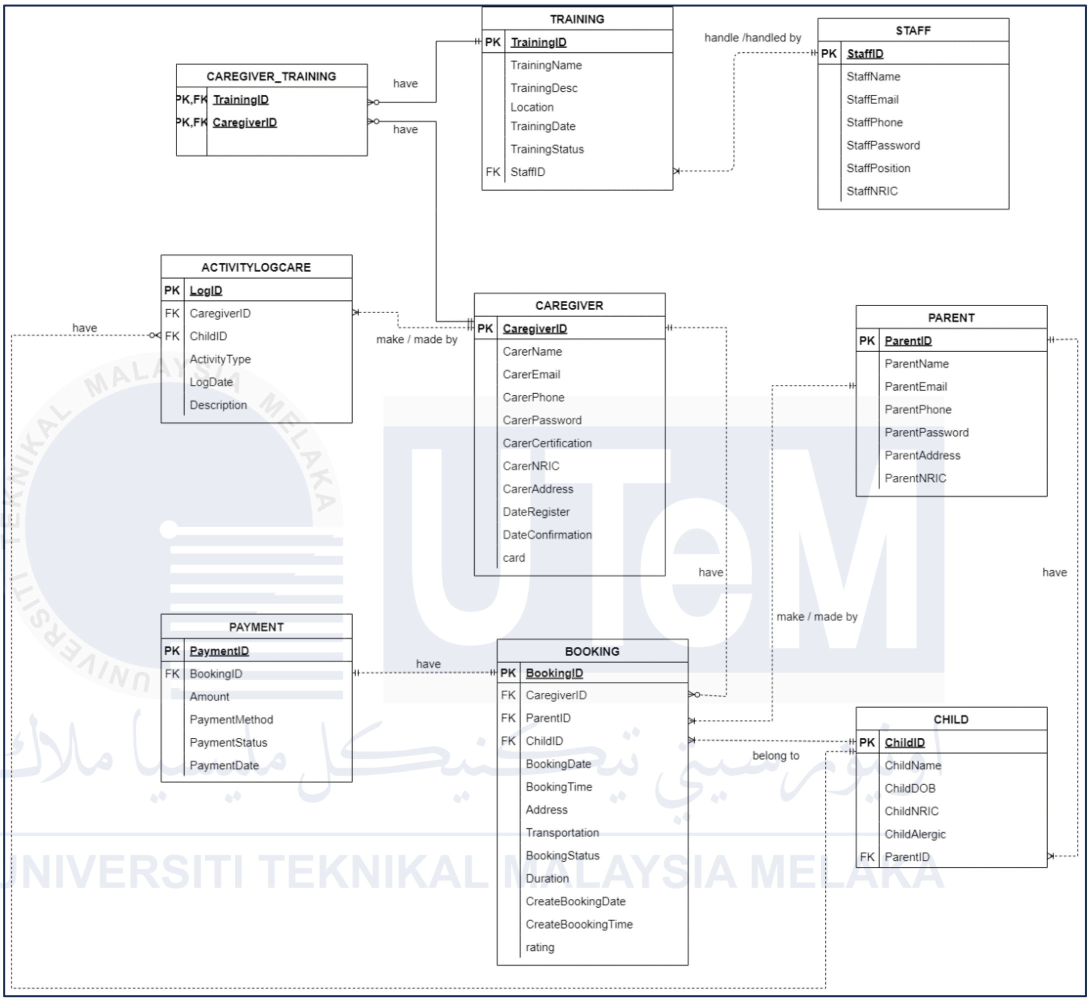


Figure 4.2 : Entity Relationship Diagram

4.3.1.3 Business Rules

i. A parent may have zero or many children, and each child belongs to one parent.

ii. A caregiver may have zero or many bookings assigned to them, and each booking is assigned to one caregiver.

iii. A parent can make zero or many booking requests, and each booking request is assigned to one parent.

iv. A caregiver can log zero or many activities for children, and each activity log is created by one caregiver.

v. A child may have zero or many activity logs, and each activity log is assigned to one child.

vi. A parent may view zero or many activity logs, and each activity log is viewable by one parent.

vii. Each payment is associated with one booking, and one booking may have zero or one payment.

viii. Each caregiver may attend zero or many training sessions, and each training session may have zero or many caregivers.

ix. A training session is scheduled by one staff member, and each staff member may schedule zero or many training sessions.

x. A child may have zero or many allergies recorded, and each allergy is recorded for one child.

xi. An administrator can monitor zero or many bookings, and each booking is monitored by one administrator.

xii. A caregiver may have zero or many certifications, and each certification is assigned to one caregiver.

xiii. Each user (Parent, Caregiver, Staff, Administrator) must have a unique email and NRIC for registration.

xiv. A caregiver may have zero or many availability slots, and each availability slot is assigned to one caregiver.

4.3.2 Logical Design

The second phase of database design is the logical design phase. During this phase, relationships between local logical data are established, normalized, and tested against user transactions. Additionally, the constraints for each piece of local logical data are evaluated.

4.3.2.1 Data Dictionary for Entity Relationship Diagram

Table 4.13 : Parent Table

Parent							
Attribute Name	**Description**	**Data type and size**	**Null**	**Default Value**	**Unique**	**PK or FK**	**Reference table**
ParentID	Parent ID	varchar(200)	NO	-	YES	PK	-
ParentName	Name of Parent	varchar(200)	NO	-	NO	-	-
ParentEmail	Email of Parent	varchar(200)	NO	-	NO	-	-
ParentPhone	Phone Number	varchar(50)	NO	-	NO	-	-
ParentPassword	Password	varchar(200)	NO	-	NO	-	-
ParentAddress	Address	varchar(200)	YES	NULL	NO	-	-
ParentNRIC	NRIC	varchar(12)	NO	-	NO	-	-

Table 4.14 : Child Table

Child							
Attribute Name	**Description**	**Data type and size**	**Null**	**Default Value**	**Unique**	**PK or FK**	**Reference table**
ChildID	Child ID	varchar(200)	NO	-	YES	PK	-
ChildName	Name of Child	varchar(200)	NO	-	NO	-	-
ChildDOB	Date of Birth	date	YES	NULL	NO	-	-
ChildNRIC	NRIC	varchar(12)	NO	-	NO	-	-
ChildAllergic	Allergies	varchar(200)	NO	-	NO	-	-
ParentID	Parent ID	varchar(200)	NO	-	NO	FK	parent

Table 4.15 : Staff Table

Staff							
Attribute Name	**Description**	**Data type and size**	**Null**	**Default Value**	**Unique**	**PK or FK**	**Reference table**
StaffID	Staff ID	varchar(200)	NO	-	YES	PK	-
StaffName	Name of Staff	varchar(200)	NO	-	NO	-	-

StaffEmail	Email of Staff	varchar(200)	NO	-	NO	-	-
StaffPhone	Phone Number	varchar(12)	NO	-	NO	-	-
StaffPassword	Password	varchar(200)	NO	-	NO	-	-
StaffPosition	Position of Staff	varchar(200)	NO	'Staff'	NO	-	-
StaffNRIC	NRIC	varchar(12)	NO	-	NO	-	-

Table 4.16 : Caregiver Table

Caregiver							
Attribute Name	**Description**	**Data type and size**	**Null**	**Default Value**	**Unique**	**PK or FK**	**Reference table**
CaregiverID	Caregiver ID	varchar(200)	NO	-	YES	PK	-
CarerName	Name of Caregiver	varchar(200)	NO	-	NO	-	-
CarerEmail	Email of Caregiver	varchar(200)	NO	-	NO	-	-
CarerPhone	Phone Number	varchar(50)	NO	-	NO	-	-

CarerAddress	Address of Caregiver	varchar(200)	NO	-	NO	-	-
CarerPassword	Password	varchar(200)	NO	-	NO	-	-
CarerCertification	Certification	varchar(200)	YES	NULL	NO	-	-
CarerNRIC	NRIC	varchar(12)	NO	-	NO	-	-
DateRegister	Date caregiver register	varchar(200)	NO	-	NO	-	-
DateConfirmation	Date caregiver become caregiver	varchar(200)	NO	-	NO	-	-
card	Card Number	Int (20)	YES	-	N0	-	-

Table 4.17 :Booking Table

Booking							
Attribute Name	**Description**	**Data type and size**	**Null**	**Default Value**	**Unique**	**PK or FK**	**Reference table**
BookingID	Booking ID	varchar(200)	NO	-	YES	PK	-
ChildID	Child ID	varchar(200)	NO	-	NO	FK	child

CaregiverID	Caregiver ID	varchar(200)	YES	NULL	NO	FK	caregiver
BookingDate	Date of Booking	varchar(200)	NO	-	NO	-	-
BookingTime	Time of Booking	varchar(200)	NO	-	NO	-	-
Address	Booking Address	varchar(200)	NO	-	NO	-	-
Transportation	Transportation Details	varchar(200)	NO	-	NO	-	-
BookingStatus	Status of Booking	varchar(200)	NO	'Payment Pending'	NO	-	-
ParentID	Parent ID	varchar(200)	NO	-	NO	FK	parent
IsSeenByCaregiver	Seen by Caregiver Flag	tinyint(1)	YES	0	NO	-	-
Duration	Duration of Booking	int(11)	YES	NULL	NO	-	-
CreateBookingDate	Date of Booking Creation	date	YES	NULL	NO	-	-
CreateBookingTime	Time of Booking Creation	time	YES	NULL	NO	-	-

rating	Rating booking	Int(20)	NO	NULL	NO	-	-

Table 4.18 : Payment Table

Payment							
Attribute Name	**Description**	**Data type and size**	**Null**	**Default Value**	**Unique**	**PK or FK**	**Reference table**
PaymentID	Payment ID	varchar(200)	NO	-	YES	PK	-
BookingID	Booking ID	varchar(200)	NO	-	NO	FK	bookings
Amount	Payment Amount	varchar(200)	NO	-	NO	-	-
PaymentMethod	Method of Payment	varchar(200)	NO	-	NO	-	-
PaymentStatus	Status of Payment	varchar(200)	NO	-	NO	-	-

Table 4.19 : ActivityLogCare Table

ActivityLogCare							
Attribute Name	**Description**	**Data type and size**	**Null**	**Default Value**	**Unique**	**PK or FK**	**Reference table**
LogID	Activity Log ID	varchar(200)	NO	-	YES	PK	-
CaregiverID	Caregiver ID	varchar(200)	NO	-	NO	FK	caregiver
ChildID	Child ID	varchar(200)	NO	-	NO	FK	child
ActivityType	Type of Activity	varchar(200)	NO	-	NO	-	-
LogDate	Date of Log	varchar(200)	NO	-	NO	-	-
Description	Description of Activity	varchar(200)	NO	-	NO	-	-

Table 4.20 : Training Table

Training							
Attribute Name	**Description**	**Data type and size**	**Null**	**Default Value**	**Unique**	**PK or FK**	**Reference table**
TrainingID	Training ID	varchar(200)	NO	-	YES	PK	-

TrainingName	Name of Training	varchar(200)	NO	-	NO	-	-
TrainingDesc	Training Description	varchar(200)	NO	-	NO	-	-
Location	Training Location	varchar(200)	NO	-	NO	-	-
TrainingDate	Date of Training	varchar(200)	NO	-	NO	-	-
TrainingStatus	Status of Training	varchar(200)	NO	'Coming Soon'	NO	-	-
StaffID	Staff ID	varchar(200)	NO	-	NO	FK	staff

Table 4.21 : Training_Caregiver Table

Training_Caregiver Table							
Attribute Name	**Description**	**Data type and size**	**Null**	**Default Value**	**Unique**	**PK or FK**	**Reference table**
CaregiverID	Caregiver ID	varchar(200)	NO	-	NO	PK, FK	caregiver
TrainingID	Training ID	varchar(200)	NO	-	NO	PK, FK	training

4.3.2.2 Query Design

There are many query designs that can be used to create different types of outcomes. Each query has its own requirements, reasons and purpose. Some of them are shown in Table 4-1 about the Query design.

Table 4.22 : SQL Table

Type of SQL	SQL Statements	Explanation
Simple sql	SELECT * FROM caregiver;	Retrieve all caregiver information.

Join Table	SELECT b.ChildID, c.ChildName, b.CaregiverID, b.BookingDate, b.BookingTime, b.Address, b.Transportation, p.ParentName, b.BookingStatus FROM bookings b
	INNER JOIN parent p ON b.ParentID = p.ParentID
	INNER JOIN child c ON b.ChildID = c.ChildID
	WHERE b.ParentID = '{$_SESSION["user_id"]}'";
		Retrieve booking data along with associated parent and child information for a specific parent.

Aggregate and Order	SELECT caregiver.CaregiverID, caregiver.CarerName, COALESCE(SUM(payment.Amount), 0) AS TotalAmountReceived
	FROM caregiver
	LEFT JOIN bookings ON caregiver.CaregiverID = bookings.CaregiverID
	LEFT JOIN payment ON bookings.BookingID = payment.BookingID
		Total amount paid to caregivers based on bookings made by parents.

	WHERE bookings.BookingStatus = 'Confirmed'
	GROUP BY caregiver.CaregiverID, caregiver.CarerName
	ORDER BY TotalAmountReceived DESC;	

4.3.3 Physical Design

The choice of Database Management System (DBMS) depends on factors like scalability, performance, ease of use, and compatibility with your application's technology stack. For a childcare management system, which likely involves transactional operations and relational data, commonly used DBMS options include MySQL. It is an open-source and widely used for relational databases, known for stability and performance.

4.3.3.1 Usage of Trigger

Table 4.23 : Trigger in Database Table

Trigger	Database Table	Query	Explanation

BEFORE INSERT	Parent	CREATE TRIGGER `ParentIDincrement` BEFORE INSERT ON `parent`
		FOR EACH ROW BEGIN
		INSERT INTO sequence_parent VALUES (NULL);
		SET NEW.ParentID = CONCAT ('P',LPAD (LAST_INSERT_ID (),2,'0'));
			Automatically generates a unique ParentID for each new parent record inserted into the database,ensuring uniqueness and

		END	conformity to a specific format.

BEFORE INSERT	Staff	CREATE TRIGGER
		`beforeinsertStaffID` BEFORE INSERT ON `staff`
		 FOR EACH ROW BEGIN
		INSERT into sequence_staff VALUES (NULL);
		SET NEW.StaffID=CONCAT('S',LPAD (LAST_INSERT_ID (),2,'0'));
		END
			Ensures each staff member inserted into the database receives a unique identifier(StaffID) formatted consistently and automatically.

BEFORE INSERT	Caregiver	CREATE TRIGGER
		`CaregiverIDincrement` BEFORE INSERT ON `caregiver`
		FOR EACH ROW BEGIN
		INSERT INTO sequence_caregiver VALUES (NULL);
		SET NEW.CaregiverID = CONCAT ('G',LPAD (LAST_INSERT_ID (),2,'0'));
		END
			Automatically generates a unique CaregiverID for each new parent record inserted into the database,ensuring uniqueness.

4.3.3.2 Usage of Procedures

Table 4.24 : Stored Procedure in Database Table

Procedures	Query	Explanation
GetParentDetails	DELIMITER $$	Provides a mechanism to fetch detailed information about a

	CREATE DEFINER=`root`@`localhost`
	PROCEDURE `GetParentDetails`(IN
	`parentId` INT)
	BEGIN
	SELECT * FROM parent WHERE
	ParentID = parentId;
	END$$
	DELIMITER ;
		specific parent based on their ParentID.

GetPendingBookings	DELIMITER $$
	CREATE DEFINER=`root`@`localhost`
	PROCEDURE `GetPendingBookings`()
	BEGIN
	SELECT * FROM bookings WHERE
	BookingStatus = 'Request Pending';
	END$$
	DELIMITER ;
		Retrieves all booking records that are currently pending approval or processing based on the BookingStatus condition.

Parent_Login	DELIMITER $$
	CREATE DEFINER=`root`@`localhost`
	PROCEDURE `Parent_Login`(IN
	`p_username` VARCHAR(255), IN
	`p_password` VARCHAR(255))
	BEGIN
	DECLARE user_count INT;
	
	 -- Check if user is a parent
	SELECT COUNT(*) INTO user_count
	FROM parent
	WHERE ParentEmail = p_username
	AND ParentPassword = p_password;
	 IF user_count = 1 THEN
	
	-- Login successful
	SELECT 'Login successful' AS message;
		Validates parent credentials during login and provides feedback on the login status.

	ELSE
	-- Login failed
	SELECT 'Invalid email or password' AS message;
	END IF;
	END$$
	DELIMITER ;	

4.3.3.3 Usage of Event

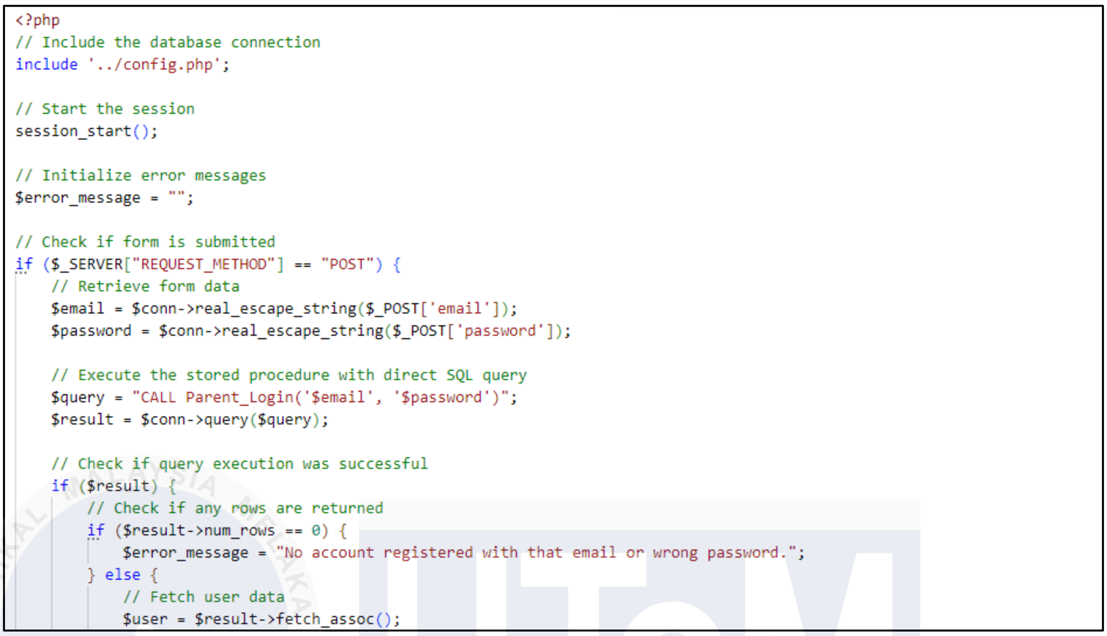
Table 4.25 : Event in Database Table

Event	Query	Explanation
auto_reject_pending_bookings	CREATE DEFINER=\`root\`@\`localhost\` EVENT \`auto_reject_pending_bookings\` ON SCHEDULE EVERY 1 MINUTE STARTS '2024-08-10 14:09:10' ON COMPLETION NOT PRESERVE ENABLE DO BEGIN UPDATE bookings SET BookingStatus = 'Canceled' WHERE BookingStatus = 'Request Pending' AND TIMESTAMPDIFF(MINUTE, CONCAT(CreateBookingDate, ' ', CreateBookingTime), NOW()) > 1440; END	This event query automatically cancels pending bookings that have been waiting for more than 60 minutes. It updates the BookingStatus to 'Canceled' for entries with BookingStatus 'Request Pending' created today, using TIMESTAMPDIFF to check the time difference between booking creation and current time. If this difference is over 60 minutes,

		the booking is canceled.

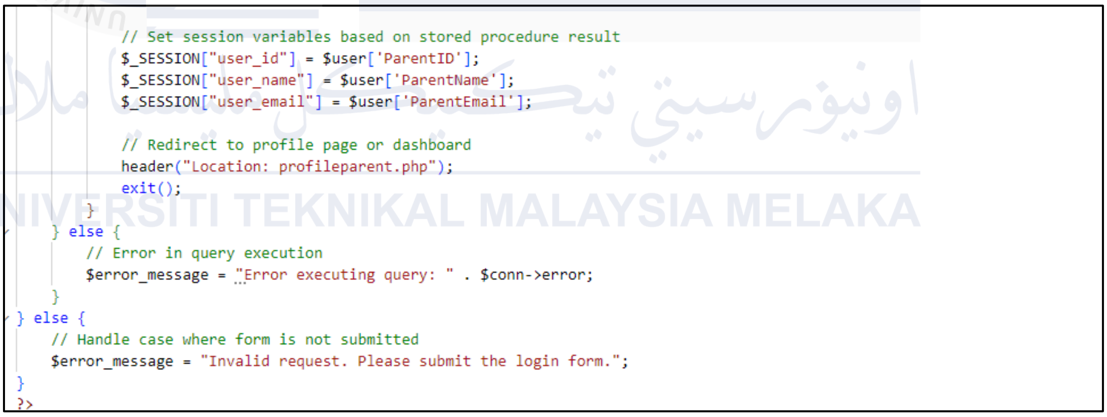
auto_cancel_passed_bookings	CREATE DEFINER=`root`@`localhost` EVENT `auto_cancel_passed_bookings` ON SCHEDULE EVERY 1 MINUTE STARTS '2024-08-10 14:09:10' ON COMPLETION NOT PRESERVE ENABLE DO BEGIN
	UPDATE bookings
	SET BookingStatus = 'Canceled'
	WHERE BookingStatus = 'Request Pending' AND BookingDate = CURDATE() AND BookingTime < CURTIME();
	END
		This event is designed to automatically cancel any bookings that are still in the 'Request Pending' state after their scheduled time has passed. For example, if a user made a booking request for a specific time today but the caregiver did not accept or confirm the request, and the scheduled time has now passed, the system will automatically change the status of that booking to 'Canceled'.

4.3.3.4 Security Mechanism



```
<?php
// Include the database connection
include '../config.php';

// Start the session
session_start();

// Initialize error messages
$error_message = "";

// Check if form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve form data
    $email = $conn->real_escape_string($_POST['email']);
    $password = $conn->real_escape_string($_POST['password']);

    // Execute the stored procedure with direct SQL query
    $query = "CALL Parent_Login('$email', '$password')";
    $result = $conn->query($query);

    // Check if query execution was successful
    if ($result) {
        // Check if any rows are returned
        if ($result->num_rows == 0) {
            $error_message = "No account registered with that email or wrong password.";
        } else {
            // Fetch user data
            $user = $result->fetch_assoc();
```

Figure 4.3 : Validation Email and password for student

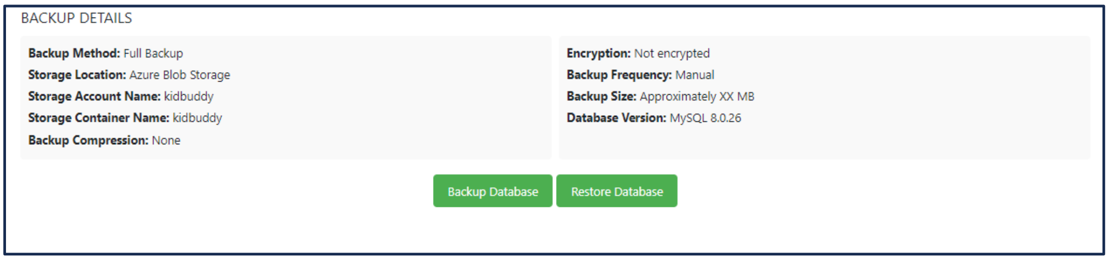


```
            // Set session variables based on stored procedure result
            $_SESSION["user_id"] = $user['ParentID'];
            $_SESSION["user_name"] = $user['ParentName'];
            $_SESSION["user_email"] = $user['ParentEmail'];

            // Redirect to profile page or dashboard
            header("Location: profileparent.php");
            exit();
        }
    } else {
        // Error in query execution
        $error_message = "Error executing query: " . $conn->error;
    }
} else {
    // Handle case where form is not submitted
    $error_message = "Invalid request. Please submit the login form.";
}
?>
```

Figure 4.4 : Validation Email and password for student

The system checks the username and password with the help of a security mechanism and identifies the user (parent) is getting to the system and navigates to the user page according to his role. For example, if a user enters an email and password for a parent account, they are logged in as a parent.

4.3.3.5 Database Contingency

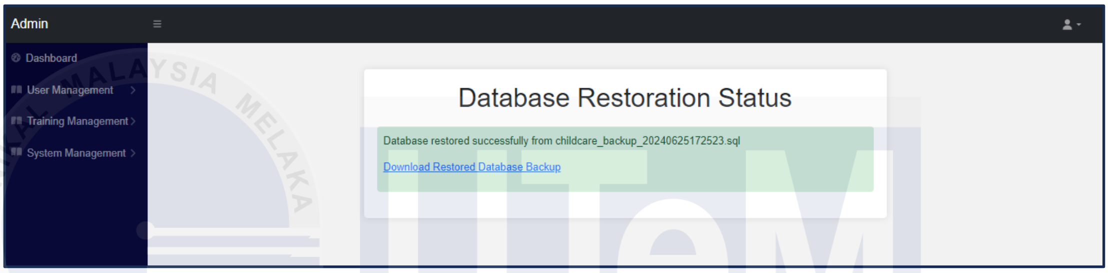


BACKUP DETAILS

Backup Method: Full Backup
Storage Location: Azure Blob Storage
Storage Account Name: kidbuddy
Storage Container Name: kidbuddy
Backup Compression: None

Encryption: Not encrypted
Backup Frequency: Manual
Backup Size: Approximately XX MB
Database Version: MySQL 8.0.26

Figure 4.5 : Database Backup and Restore



Database Restoration Status

Database restored successfully from childcare_backup_20240625172523.sql

Download Restored Database Backup

Figure 4.6 : Database in PhpMyAdmin restored successfully

The database contingency plan includes a robust backup and recovery mechanism to ensure data integrity and availability. I utilize automated backups to Azure Blob Storage, where full backups of the MySQL database are periodically created and stored securely. In the event of data loss or corruption, the latest backup can be restored using a streamlined process that involves fetching the most recent backup file from Azure Blob Storage and restoring it to the MySQL database. This mechanism provides a reliable way to recover from unexpected failures, ensuring minimal downtime and data loss. Regular testing of the backup and recovery process is conducted to validate its effectiveness and make improvements as needed.

4.4 Graphical User Interface (GUI) Design

Graphical user interface design encompasses the overall process of planning how users interact with the system and what inputs and outputs the system handles. It includes screens for user navigation within the system, as well as forms and interfaces for data input and output.

4.4.1 Navigation Design

Navigation design refers to how users interact with and move through a software application or website. Effective navigation design ensures that users can easily find the information or features they need without becoming confused or frustrated. It involves organizing content, creating logical pathways, and providing clear visual cues that guide users through the system.

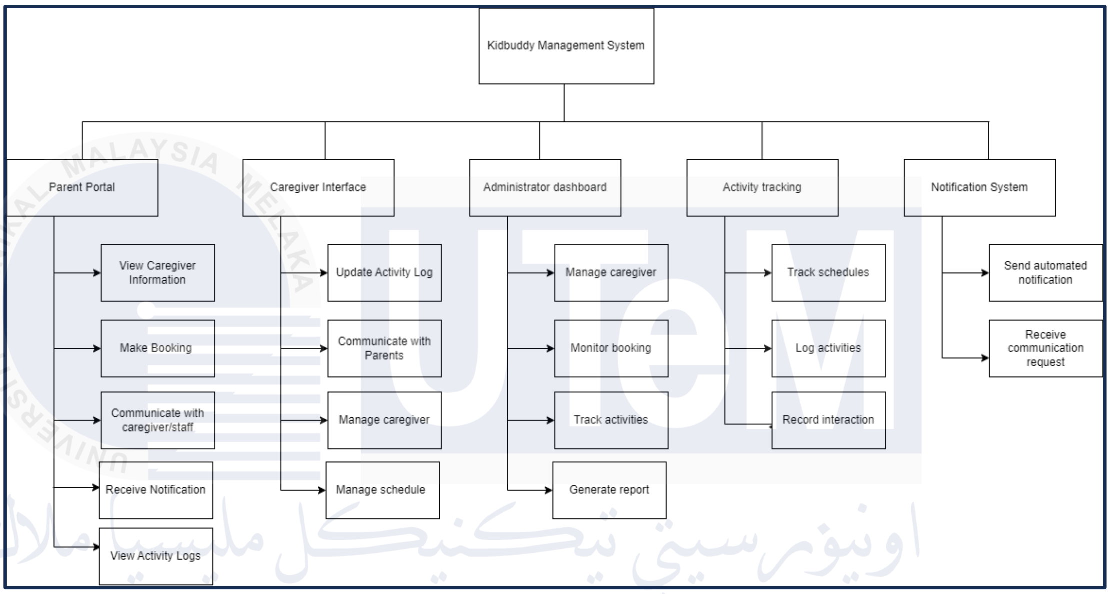


Figure 4.7 : Navigation Design for Kidbuddy

The navigation design of the kidbuddy management system is structured to cater to different user roles, ensuring an intuitive and user-friendly experience. The system is divided into five main sections: Parent Portal, Caregiver Interface, Administrator Dashboard, Activity Tracking, and Notification System. Each section is tailored to the specific needs of the users, providing them with easy access to relevant features.

The parent portal allows parents to view caregiver information, make bookings, communicate with caregivers or staff, receive notifications, and view activity logs. This design ensures that parents can quickly and efficiently manage all aspects of their child's care from one central location. The flow is logical, guiding parents from viewing caregivers to making bookings and managing communications seamlessly.

The caregiver interface focuses on the caregivers' daily tasks, enabling them to update activity logs, communicate with parents, manage their schedules, and maintain their profiles. The navigation is designed to facilitate the efficient management of caregiving tasks, ensuring that caregivers can keep their information up-to-date and easily accessible.

The administrator dashboard centralizes control and monitoring functions, allowing administrators to manage caregiver profiles, monitor bookings, track activities, and generate reports. This design provides administrators with a comprehensive overview of the system, enabling them to manage operations effectively and make informed decisions.

The activity tracking section is crucial for maintaining the system’s accuracy and accountability. It allows for the tracking of schedules, logging of activities, and recording of interactions, ensuring that all activities within the system are documented and can be reviewed when necessary.

Finally, the notification system ensures that users stay informed and engaged by sending automated notifications and managing communication requests. This system is designed to keep users updated in real-time, enhancing the overall user experience by ensuring timely responses and notifications.

Overall, the navigation design of the kidbuddy management system is well-organized and user-centric, providing clear pathways for each user role to access the features they need. This design supports the system's goals of being efficient, reliable, and easy to use, ultimately contributing to a positive user experience.

4.4.2 Input and Output System Design

The input and output system design of the kidbuddy management system is carefully structured to ensure smooth and efficient data handling, which is essential for the overall functionality and usability of the system.

The input system is designed to gather data from various users, including parents, caregivers, and administrators, in a manner that is both intuitive and secure. For parents, the input system allows them to enter details such as booking requests, communication messages, and feedback. The system is designed to capture these inputs accurately, with forms that guide the user through each step, ensuring that all necessary information is provided. Caregivers, on the other hand, input data related to activity logs, schedules, and communication with parents. The system is designed to accommodate these inputs through user-friendly interfaces that allow quick updates and modifications. For administrators, the input system involves entering and managing data related to caregiver profiles, bookings, and system configurations. This system is equipped with validation mechanisms that check for consistency and completeness before data is processed, minimizing errors and ensuring the integrity of the information.

The output system is designed to provide users with meaningful and actionable information based on the data processed within the system. For parents, the output system generates real-time notifications, booking confirmations, and access to caregiver profiles and activity logs. This ensures that parents have all the information they need to make informed decisions regarding their child’s care. Caregivers receive outputs such as updated schedules, communication logs, and notifications about bookings, allowing them to manage their tasks effectively. Administrators, on the other hand, receive outputs in the form of reports, booking summaries, and system alerts, which help them monitor system performance and manage operations efficiently. The system is designed to deliver these outputs in a clear and organized manner, with dashboards and reports that are easy to read and understand.

4.4.2.1 Display of Information

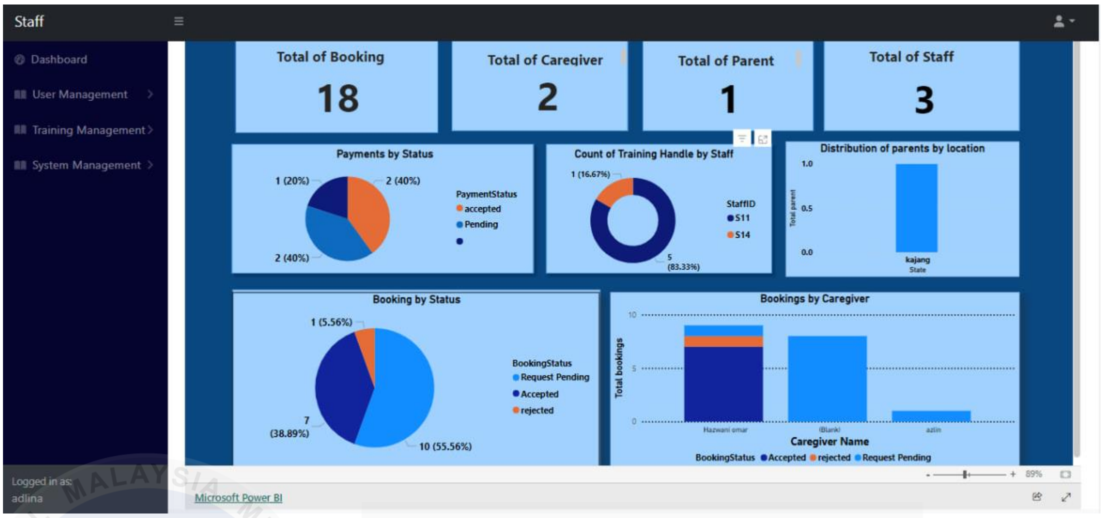


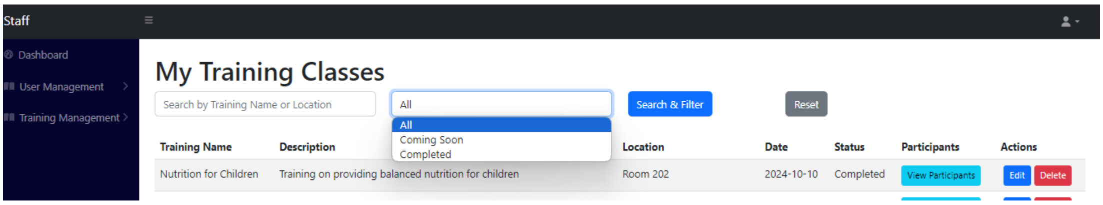
Figure 4.8 : Admin Dashboard Page Using Power BI

Based on figure 4.7, the admin dashboard page, powered by Power BI, offers administrators a powerful and interactive platform for overseeing and managing different aspects of the system. The dashboard serves as a centralized hub, where administrators can quickly access and interpret critical information, enabling informed decision-making and efficient management of operations.

Dashboard widgets are key components of this interface, designed to present key metrics and summaries in a visually intuitive manner. These widgets function as snapshots of the system’s performance, highlighting important data points such as user activity, booking statistics, caregiver performance, financial metrics, and system health. By consolidating this information into easily digestible formats, widgets allow administrators to monitor the system's status at a glance, without the need to delve into detailed reports.

Power BI’s dynamic capabilities allow these widgets to be interactive. Administrators can drill down into specific data points for a more detailed analysis, filter results by various criteria, and even customize the dashboard layout to suit their specific needs.

This interactivity not only enhances user engagement but also empowers administrators to take a proactive approach to system management.



Training Name	Description	Location	Date	Status	Participants	Actions
Nutrition for Children	Training on providing balanced nutrition for children	Room 202	2024-10-10	Completed	View Participants	Edit Delete

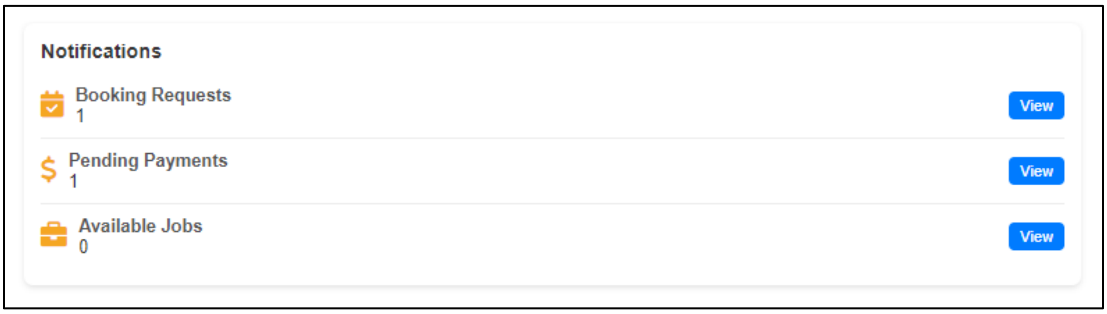
Figure 4.9 : Customizable Views

Based on figure 4.8, customizable views are a powerful feature that significantly enhances the usability and flexibility of a system by allowing users to personalize how information is displayed according to their specific needs and preferences. These views enable users to focus on the most relevant data for their roles and tasks, ensuring that they can interact with the system in a way that best supports their work.

One of the key aspects of customizable views is the ability to **sort** information. Sorting allows users to organize data in a particular order, such as alphabetically, numerically, or by date. For example, an administrator might sort a list of caregivers by performance rating or a list of bookings by date to quickly identify upcoming appointments. This capability ensures that the most important or relevant data is always at the user's fingertips.

Another critical feature is filtering, which enables users to narrow down large datasets to focus on specific criteria. For instance, a user could filter the list of available caregivers to show only those with specific qualifications or availability. By applying filters, users can quickly isolate the data that matters most to them, reducing information overload and improving efficiency.

4.4.2.2 Feedback and Notification



Notifications

Booking Requests
1

Pending Payments
1

Available Jobs
0

Figure 4.10 : Notification of Request Booking

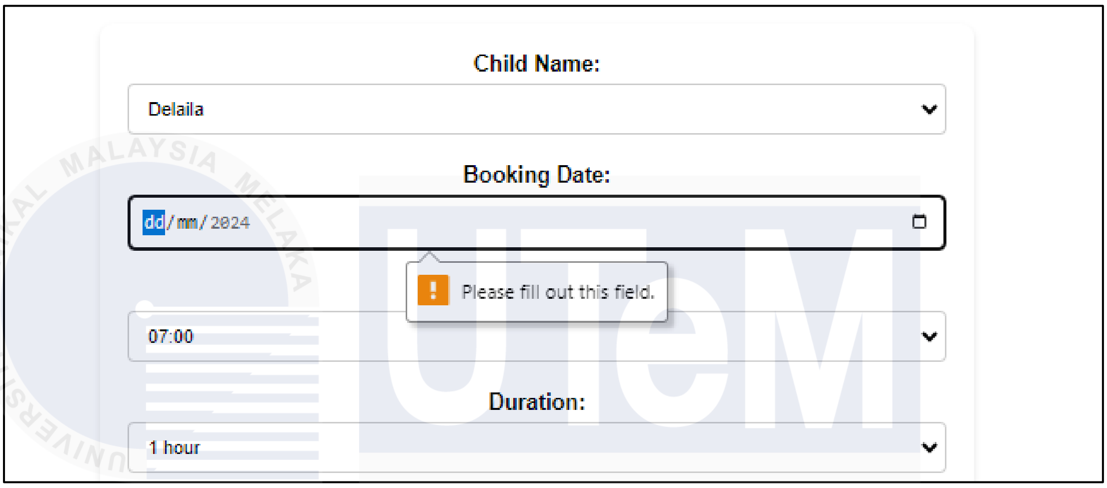
Based on figure 4.9, real-time updates and notifications are pivotal features in a graphical user interface, designed to ensure that users receive the most current information instantly. These features significantly enhance the responsiveness and interactivity of the system, making it more efficient and user-friendly.

When a system incorporates real time updates, any changes or new data such as booking requests, payment status, or job availability are immediately displayed on the user's screen. For example, when a parent submits a new booking request, the system automatically updates the caregiver's dashboard to reflect this new opportunity without requiring the caregiver to refresh the page manually. This immediate reflection of changes allows caregivers to respond promptly to new job opportunities, ensuring they don't miss out on potential work. The ability to see how many jobs are available at any given moment empowers caregivers to manage their schedules more effectively and make informed decisions about accepting new bookings.

Real-time notifications play an equally important role by alerting users to critical events as they happen. These notifications can include appointment reminders, payment confirmations, or system alerts that require immediate attention. For instance, a caregiver might receive a real-time notification when a booking is confirmed or when a payment has been processed. Similarly, parents can be notified instantly when their booking request is accepted or if there is any change in the caregiver's availability.

The inclusion of real-time updates and notifications in the system keeps users constantly engaged and informed, reducing the likelihood of missed information or delays in action. This immediacy not only improves user satisfaction but also enhances the overall efficiency of the system, as it allows all parties involved to stay aligned and act quickly on the most current data. Whether it's ensuring that a caregiver is aware of a new job opportunity or alerting a parent to a booking confirmation, real-time updates and notifications make the system more dynamic, interactive, and user centric.

4.4.2.3 Error Handling



Child Name:

Delaila

Booking Date:

dd/mm/2024

Please fill out this field.

07:00

Duration:

1 hour

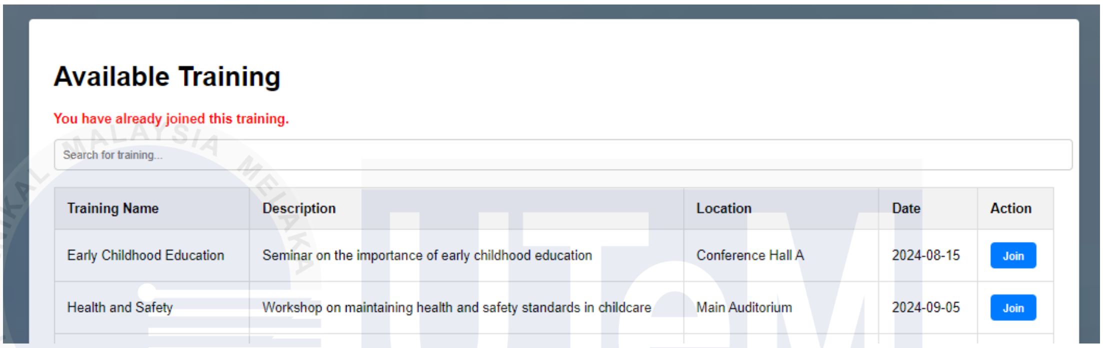
Figure 4.11 : Error Handling at Booking Page

Based on figure 4.10, error handling at booking page illustrates the system’s approach to managing user input errors on the booking page. This figure specifically addresses the scenario where a user attempts to submit the form without completing all the required fields.

In this context, the figure demonstrates how the system effectively handles incomplete submissions. When a user leaves one or more required fields blank and clicks the "Submit" button, the system triggers an error handling mechanism designed to alert the user to the omission. The error message, prominently displayed on the page, reads "Please fill out this field," clearly indicating which fields need to be completed before the form can be submitted successfully.

The error message is strategically positioned near the relevant fields to ensure users can easily identify and correct their mistakes. This approach minimizes confusion and guides users toward resolving the issue. By providing immediate and specific feedback, the system helps users understand what needs to be fixed, thereby improving the overall usability of the form and reducing the likelihood of submission errors.

Additionally, the use of such error handling features enhances user experience by preventing incomplete or incorrect data from being processed. This not only ensures the accuracy and completeness of the information collected but also contributes to a smoother and more efficient booking process.



Available Training

You have already joined this training.

Training Name	Description	Location	Date	Action
Early Childhood Education	Seminar on the importance of early childhood education	Conference Hall A	2024-08-15	Join
Health and Safety	Workshop on maintaining health and safety standards in childcare	Main Auditorium	2024-09-05	Join

Figure 4.12 : Enrollment Status Verification

Based on figure 4.11, enrolment status verification is a crucial feature implemented to enhance the efficiency and organization of the training management system by preventing caregivers from enrolling in the same training session more than once. This feature plays a significant role in ensuring that training schedules remain streamlined and that resources are utilized effectively.

When a caregiver attempts to enrol in a training session they have previously completed, the system's enrolment status verification mechanism activates. It checks the caregiver’s existing training records to determine whether they have already participated in the specific session. If the system identifies that the caregiver is already enrolled, it generates an informative error message, such as "You have already joined this training." This message is displayed prominently, alerting the caregiver to their existing enrolment status.

By implementing this feature, the system prevents duplicate enrolments, which can lead to inefficiencies such as redundant training sessions and unnecessary administrative overhead. This not only helps in maintaining a well-organized training schedule but also ensures that caregivers engage with a diverse range of training opportunities rather than repeating the same sessions. The clear and direct feedback provided by the system helps caregivers quickly understand and correct their actions, thereby improving their overall experience and satisfaction with the training process.

Moreover, this feature contributes to the efficient management of training resources and schedules, allowing administrators to better allocate training sessions and monitor caregiver progress. It helps in avoiding overbooking of training sessions and ensures that each session is attended by the intended participants, enhancing the overall quality and effectiveness of the training program.

4.4.2.4 User Registration and Login Module

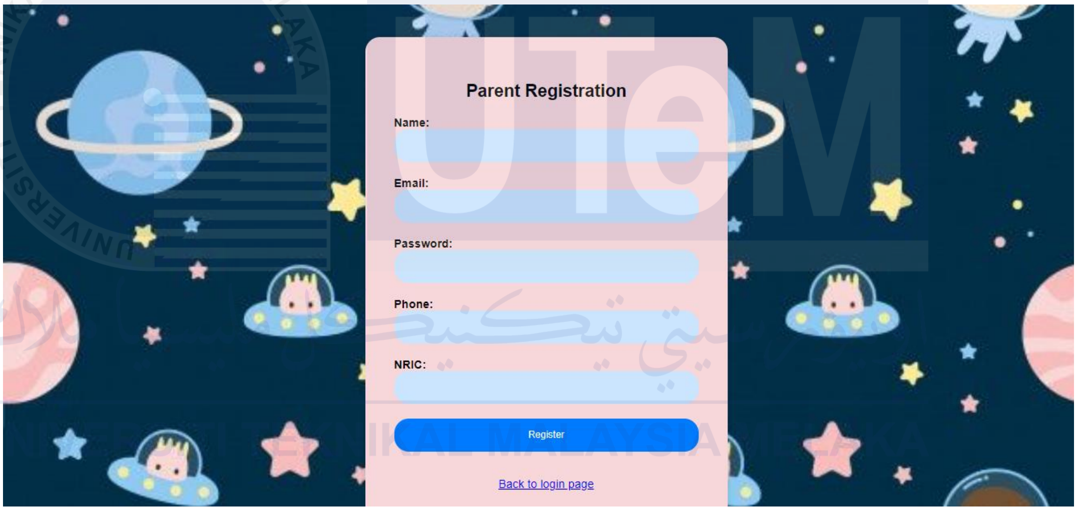
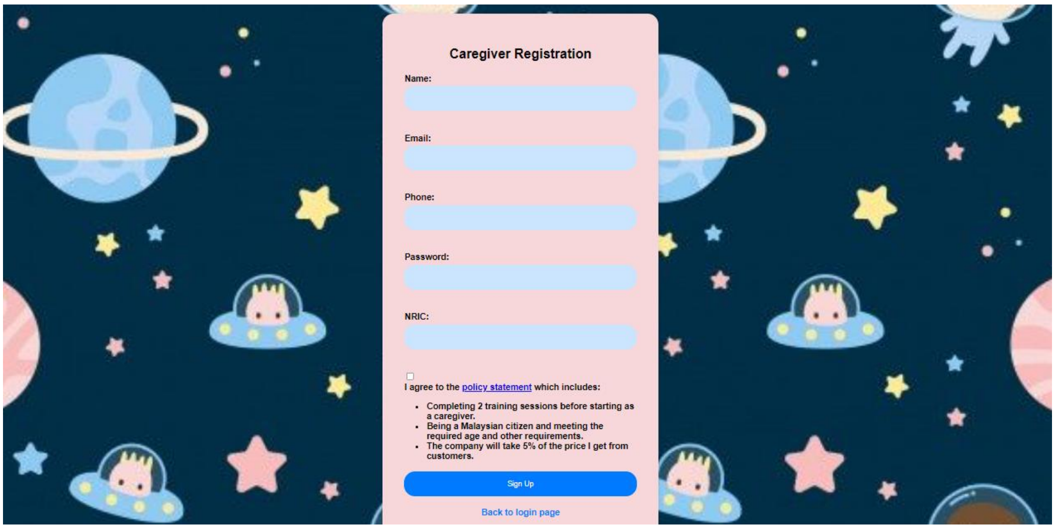


Figure 4.13 : Parent Registration Page



□ I agree to the policy statement which includes:

- Completing 2 training sessions before starting as a caregiver.
- Being a Malaysian citizen and meeting the required age and other requirements.
- The company will take 5% of the price I get from customers.

Figure 4.14 : Caregiver Registration Page

Based on figure 4.12 and figure 4.13, the parent and caregiver registration page are a central component of the child day care management system, serving as the gateway for new users both parents and caregivers to create accounts and gain access to the platform. This page is crucial for onboarding new users and ensuring that they are seamlessly integrated into the system.

The registration page is designed with user experience in mind, focusing on simplicity and efficiency to streamline the account creation process. For new parents, the registration form typically requires essential information such as their name, contact details, address, and any specific preferences or requirements related to childcare. Similarly, caregivers are required to provide their personal details, qualifications, certification information, and availability.

To enhance the registration process, the page incorporates real-time validation and error handling. As users input their details, the system checks for common errors such as missing required fields, incorrect email formats, or weak passwords. Immediate feedback is provided through validation messages, helping users correct any issues before submission. For example, if a user leaves a required field blank and attempts to submit the form, an error message such as "Please fill out this field" will prompt them to complete the missing information.

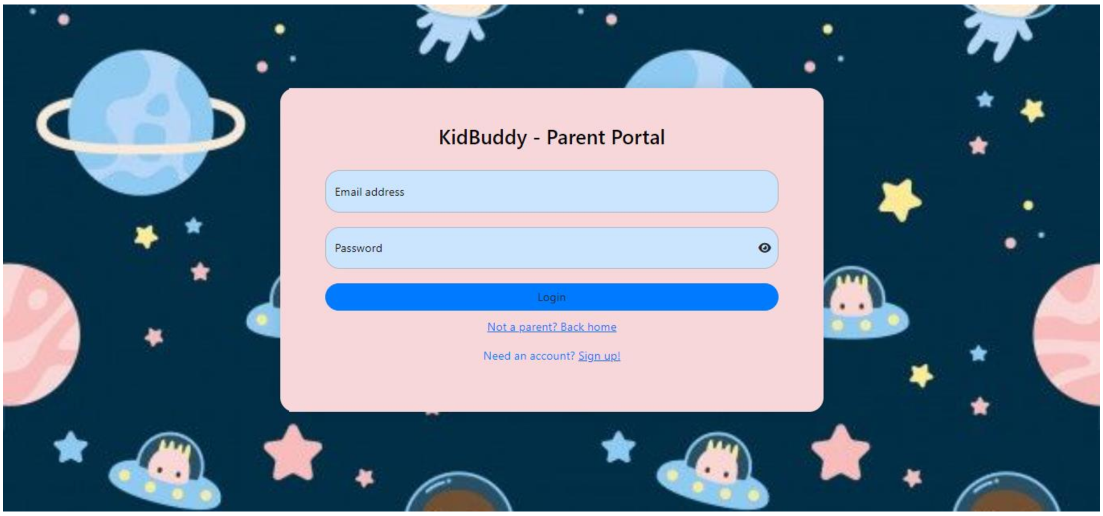
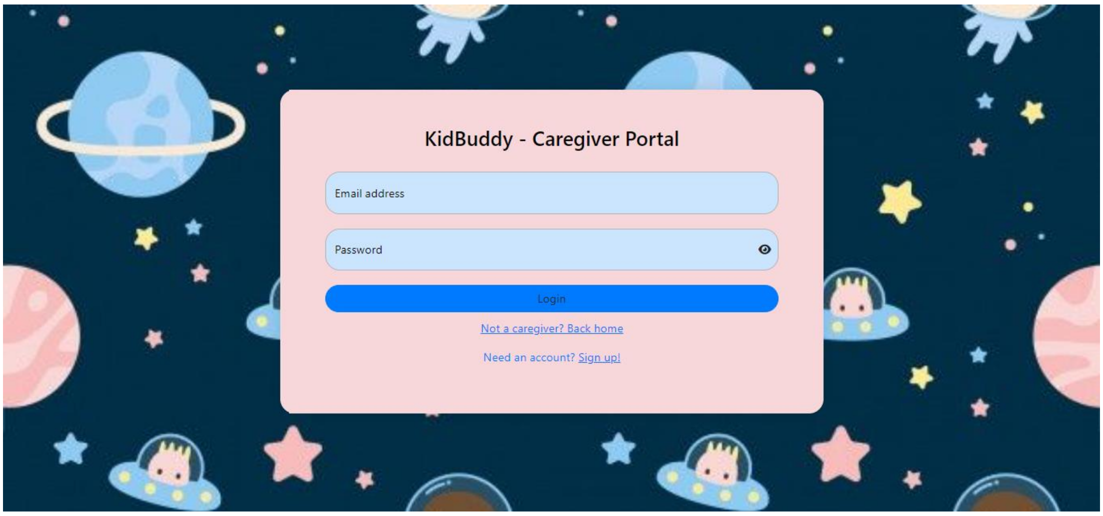


Figure 4.15 : Parent Login Page



KidBuddy - Caregiver Portal

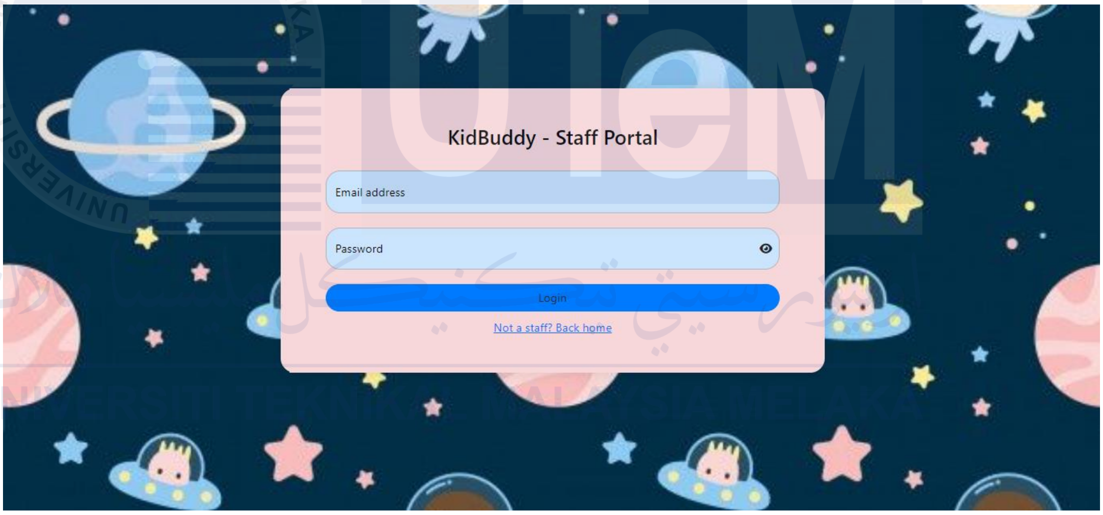
Email address

Password

Not a caregiver? Back home

Need an account? Sign up!

Figure 4.16 : Caregiver Login Page



KidBuddy - Staff Portal

Email address

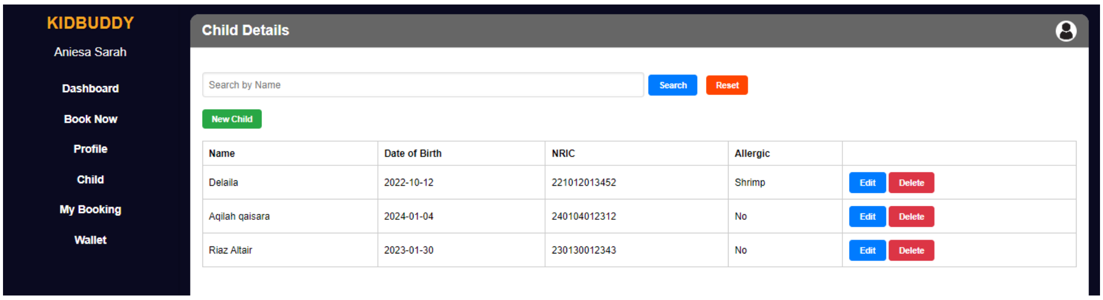
Password

Not a staff? Back home

Figure 4.17 : Staff Login Page

Based on figure 4.14, figure 4.15 and figure 4.16, the parent, caregiver, and staff login page are a critical interface within the child day care management system, providing secure access to the platform for different types of users. This page is designed to ensure that only authorized users can log in and access their respective dashboards and functionalities.

4.4.2.5 Parent Module



Name	Date of Birth	NRIC	Allergic	
Delaila	2022-10-12	221012013452	Shrimp	Edit Delete
Aqilah qaisara	2024-01-04	240104012312	No	Edit Delete
Riaz Altair	2023-01-30	230130012343	No	Edit Delete

Figure 4.18 : Child Details Page

Based on figure 4.17, the child details page is a crucial component of the child day care management system, meticulously crafted to offer a comprehensive overview of each child registered in the system. This page serves as a centralized hub where parents, caregivers, and administrative staff can access detailed information about a child, including personal data, health records, dietary preferences, and any special needs or requirements. By providing a holistic view of the child's profile, the child details page ensures that all parties involved in the child's care are well-informed and can tailor their approach to meet the child's unique needs. This feature not only enhances the efficiency of the day care management process but also contributes to the overall safety and well-being of the children under the center's care.

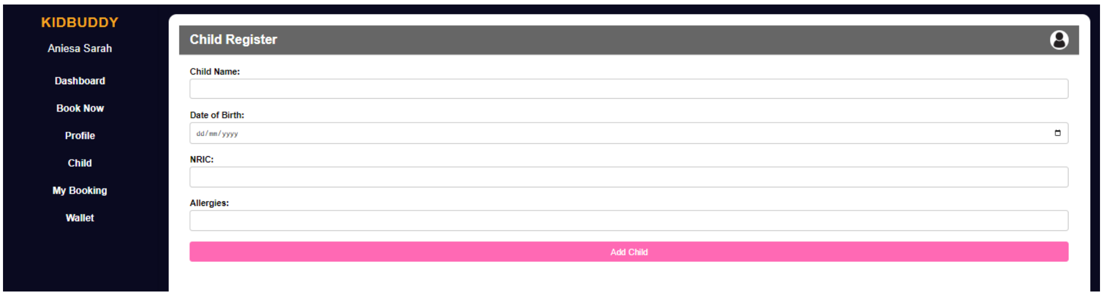
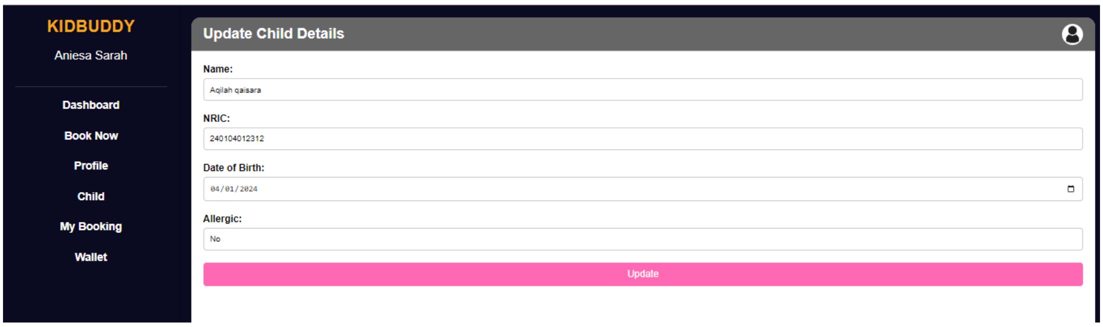


Figure 4.19 : Add Child Information Page

Based on figure 4.18, the add child information page plays a pivotal role in the child day care management system, serving as the primary interface for parents to register new children into the system. This page is thoughtfully designed to guide parents through the process of entering essential details about their child, including personal information, health conditions and any specific care instructions. This functionality not only simplifies the enrollment process for parents but also equips caregivers with the information they need to ensure the well-being and safety of every child from the moment they are registered.



Update Child Details

Name: Aqilah qaisara

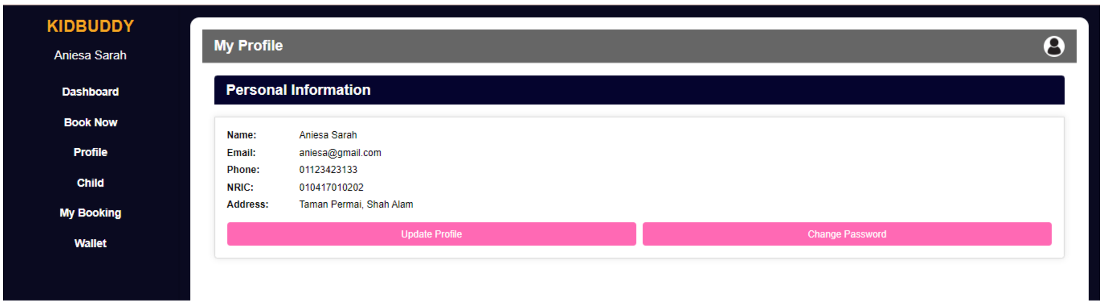
NRIC: 240104012312

Date of Birth: 04/01/2024

Allergic: No

Figure 4.20 : Update Child Information Page

Based on figure 4.19, the update child information page is an integral feature of the child day care management system, designed to empower parents with the ability to easily edit and update their child's details as needed. This functionality is vital for maintaining accurate and current information within the system, ensuring that caregivers are always informed with the most up-to-date data regarding each child's health, dietary preferences, and any special requirements. By enabling parents to promptly revise their child's information, the system helps prevent potential miscommunication or oversight, which could impact the quality of care provided. This feature not only enhances the responsiveness of the care provided but also reinforces the commitment to personalized and effective childcare, where each child's unique needs are consistently met.



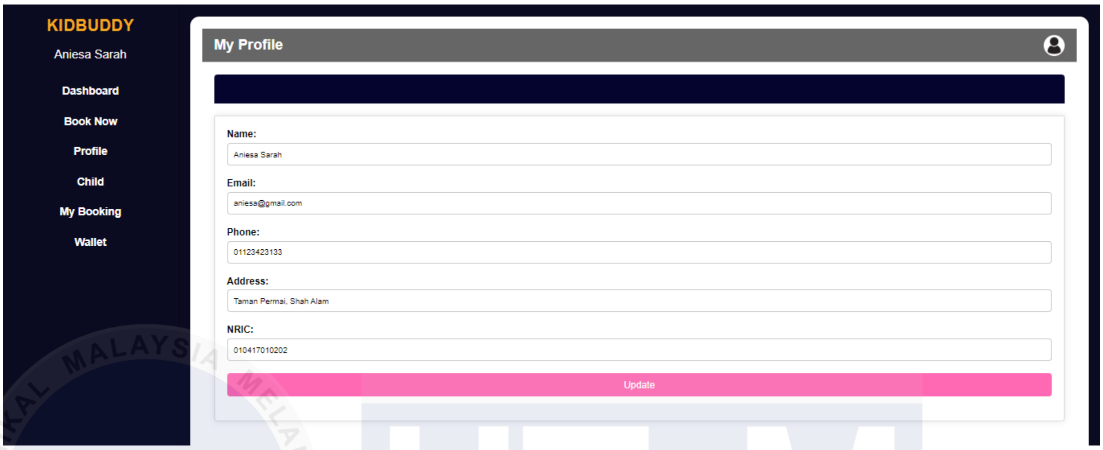
My Profile

Personal Information

Name:	Aniesa Sarah
Email:	aniesa@gmail.com
Phone:	01123423133
NRIC:	010417010202
Address:	Taman Permai, Shah Alam

Figure 4.21 : Parent Profile Page

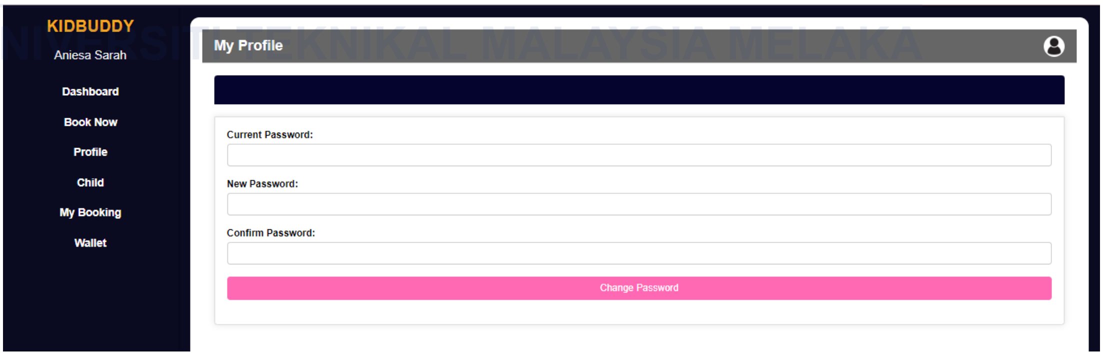
Based on figure 4.20, the parent profile page provides a detailed overview of a parent's or guardian's information and caregiving preferences. It includes personal information such as the parent's name and contact details.



Name:
Aniesa Sarah
Email:
aniesa@gmail.com
Phone:
01123423133
Address:
Taman Permai, Shah Alam
NRIC:
010417010202

Figure 4.22 : Update Profile Parent Page

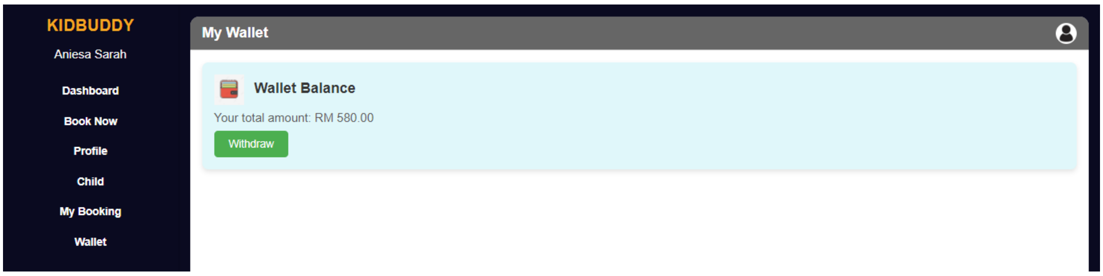
Based on figure 4.21, the update profile parent page allows parents or guardians to modify and update their personal information and caregiving preferences. This page includes fields for updating personal details such as name and contact information.



Current Password:
New Password:
Confirm Password:

Figure 4.23 : Change Password Page

Based on figure 4.22, the change password page enables users to update their account passwords securely. It typically includes fields where users can enter their current password, followed by fields for the new password and a confirmation of the new password.



My Wallet

Wallet Balance

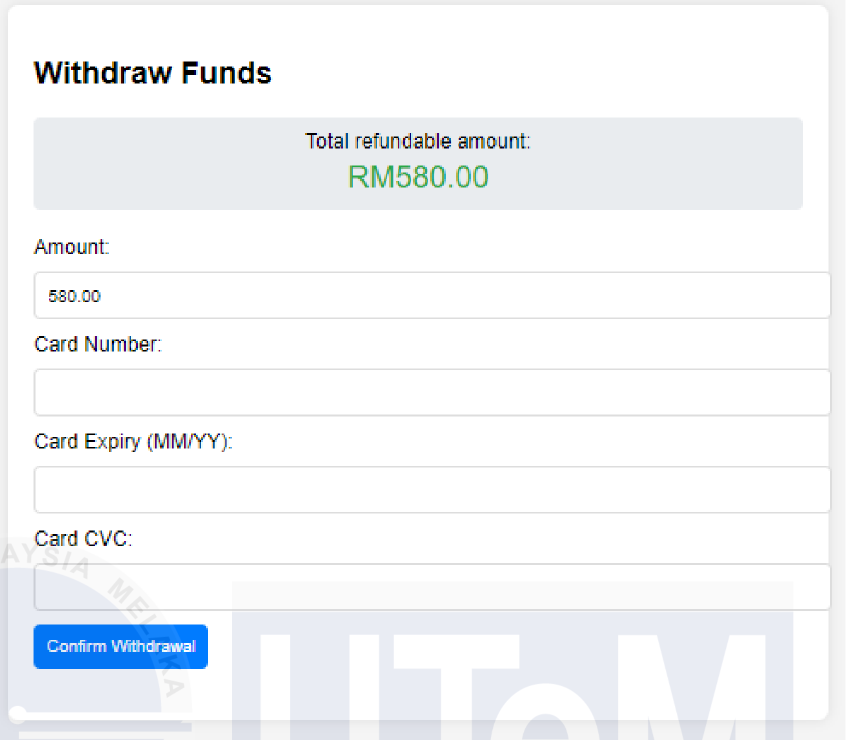
Your total amount: RM 580.00

Figure 4.24 : MyWallet Page

Based on figure 4.23, the "My Wallet" page you've shared is a section of the KidBuddy platform, designed for parents to manage their finances related to bookings. The primary focus of this page is to display the wallet balance, which represents the total amount of money that the parent has paid for bookings that were later canceled. Instead of losing these funds, the money is returned to their wallet within the system.

The page layout is clean and straightforward. At the top, there's a header labeled "My Wallet," indicating the current section. Below it, a light blue box prominently displays the wallet balance, showing the total refundable amount (in this case, RM 580.00) that the parent has available in their wallet. This balance is the cumulative total of all canceled bookings that have been refunded back to the parent’s wallet.

There is also a withdraw button, which allows the parent to withdraw the balance to their bank account or card. This feature ensures that parents can reclaim their funds easily after a booking cancellation. The sidebar on the left provides navigation options, including links to the Dashboard, Book Now, Profile, Child, My Booking, and Wallet sections. The page is designed to be user-friendly, with a clear emphasis on simplicity and accessibility.



Withdraw Funds

Total refundable amount:
RM580.00

Amount:
580.00

Card Number:

Card Expiry (MM/YY):

Card CVC:

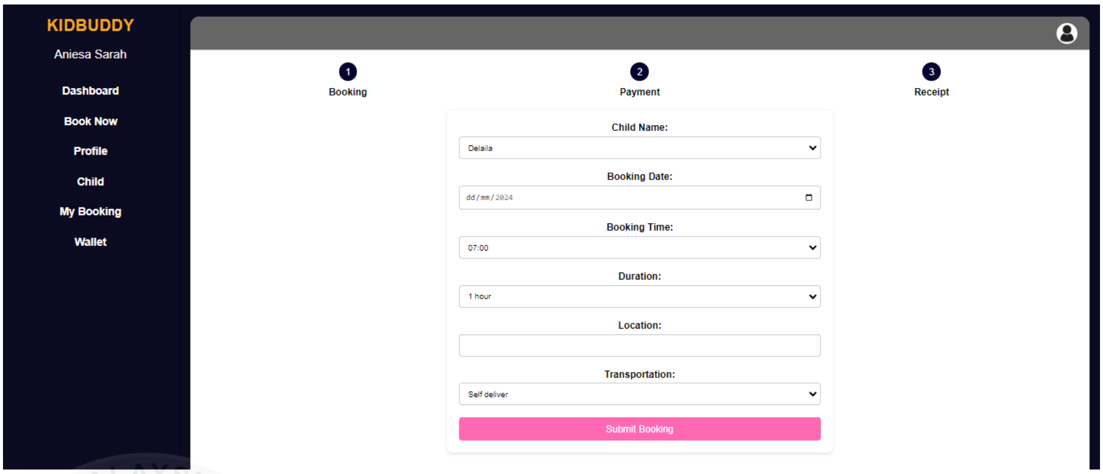
Figure 4.25 : Withdraw funds page

The withdraw funds page is designed to allow users, specifically parents, to withdraw the funds available in their wallet, which were returned to them after booking cancellations. At the top of the page, a clear and prominent label, **"Total refundable amount: RM 580.00"**, informs the user of the exact amount they are eligible to withdraw. This amount reflects the total money refunded to their wallet due to canceled bookings.

Below this, the amount field is automatically populated with the total refundable amount, making it easy for the user to proceed with the full withdrawal. The user is required to enter their card details, including the card number, card expiry (MM/YY), and card CVC (Card Verification Code). These fields ensure that the withdrawal is securely processed to the correct card.

At the bottom of the form, a confirm withdrawal button is provided. Once the user fills in the necessary card details and clicks this button, the system processes the withdrawal request, transferring the specified amount back to the user's card. The design is simple and intuitive, focusing on ease of use and ensuring that the withdrawal process is straightforward and secure.

4.4.2.6 Booking Management Module



Child Name: Delaila

Booking Date: dd/mm/2024

Booking Time: 07:00

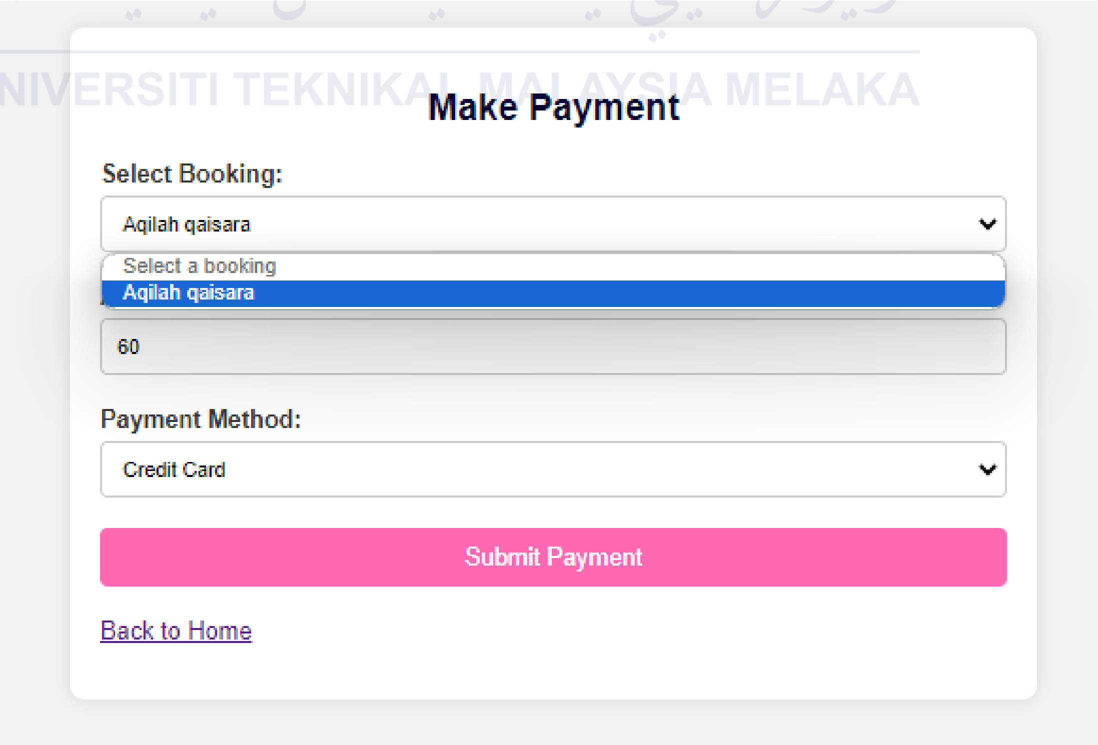
Duration: 1 hour

Location:

Transportation: Self deliver

Figure 4.26 : Create Booking with Random Caregiver Page

Based on figure 4.25, the created booking with random caregiver page provides parents with an easy and efficient way to book childcare services. This page simplifies the process by randomly assigning an available caregiver, making it ideal for parents who do not have specific preferences.



Make Payment

Select Booking: Aqilah qaisara

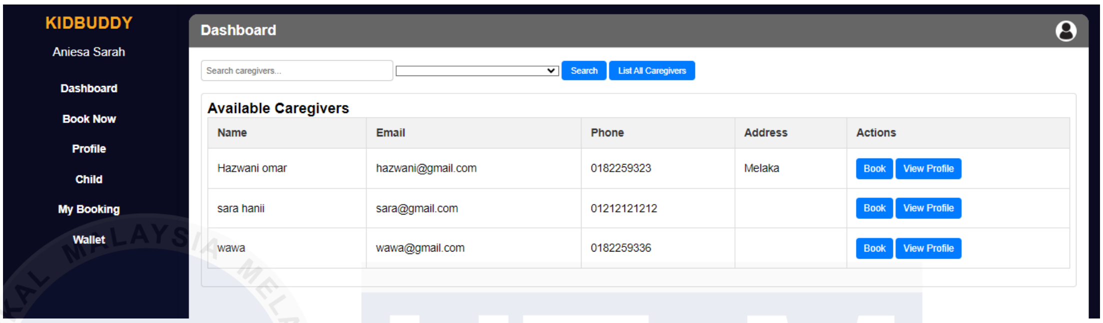
Select a booking
Aqilah qaisara

60

Payment Method: Credit Card

Figure 4.27 : Payment After Fill in The Booking Form Page

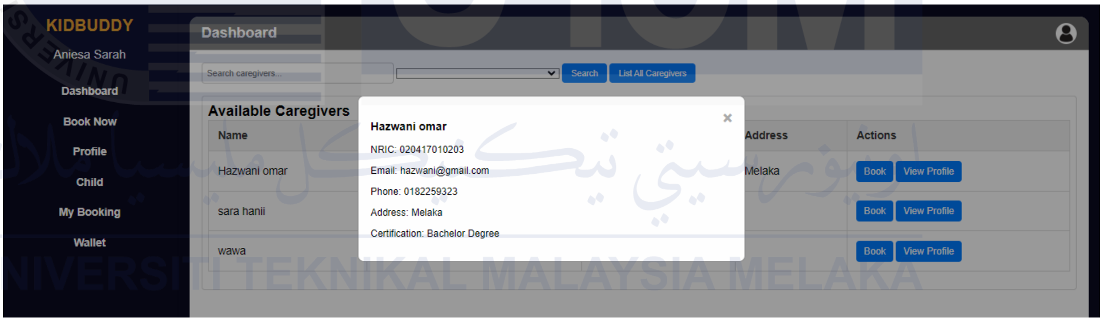
Based on figure 4.26, the payment after filling in the booking form page is designed to streamline the process of completing a childcare booking by facilitating immediate payment once the booking form is filled out. The Amount to be paid based on the duration in the booking form. This page ensures that the transaction is processed efficiently, providing parents with a secure and hassle-free payment experience.



Available Caregivers

Name	Email	Phone	Address	Actions
Hazwani omar	hazwani@gmail.com	0182259323	Melaka	Book View Profile
sara hanii	sara@gmail.com	01212121212		Book View Profile
wawa	wawa@gmail.com	0182259336		Book View Profile

Figure 4.28 : Choose Available Caregiver Page



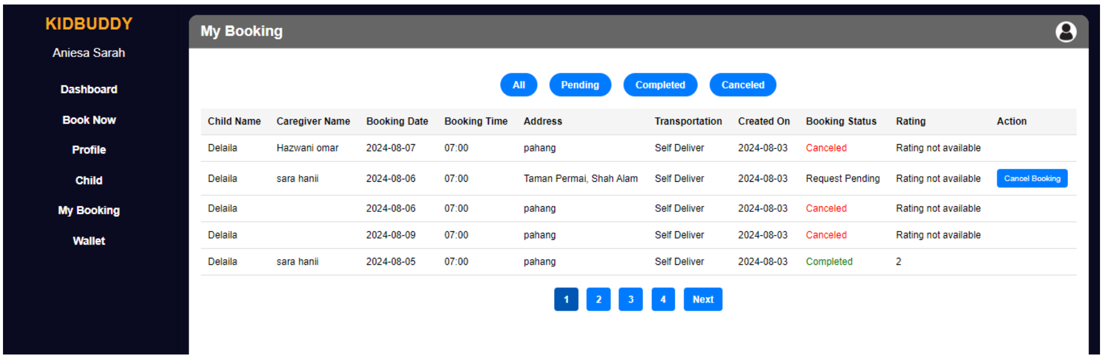
Hazwani omar
NRIC: 020417010203
Email: hazwani@gmail.com
Phone: 0182259323
Address: Melaka
Certification: Bachelor Degree

Available Caregivers

Name	Address	Actions
Hazwani omar	Melaka	Book View Profile
sara hanii		Book View Profile
wawa		Book View Profile

Figure 4.29 : Caregiver Information Page

Based on figure 4.27 and figure 4.28, the choose available caregiver page and caregiver information is designed to provide parents with the flexibility to select a caregiver from a list of available options based on their preferences and the caregiver's availability. This page ensures that parents can make an informed choice by viewing caregiver profiles and selecting the one that best suits their needs.

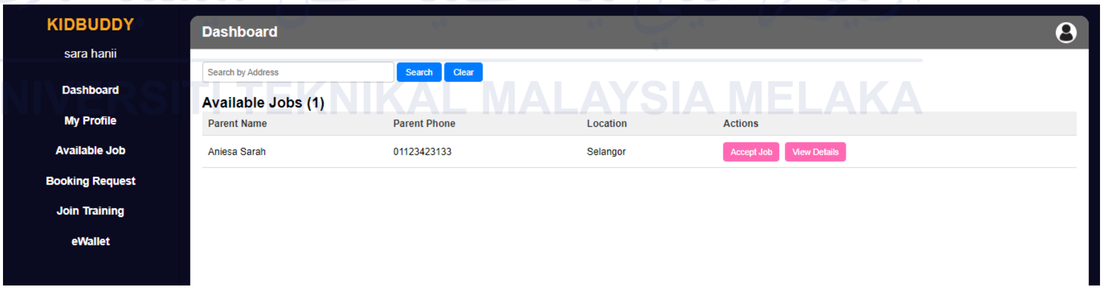


Child Name	Caregiver Name	Booking Date	Booking Time	Address	Transportation	Created On	Booking Status	Rating	Action
Delaila	Hazwani omar	2024-08-07	07:00	pahang	Self Deliver	2024-08-03	Canceled	Rating not available	
Delaila	sara hanii	2024-08-06	07:00	Taman Permai, Shah Alam	Self Deliver	2024-08-03	Request Pending	Rating not available	Cancel Booking
Delaila		2024-08-06	07:00	pahang	Self Deliver	2024-08-03	Canceled	Rating not available	
Delaila		2024-08-09	07:00	pahang	Self Deliver	2024-08-03	Canceled	Rating not available	
Delaila	sara hanii	2024-08-05	07:00	pahang	Self Deliver	2024-08-03	Completed	2	

Figure 4.30 : Booking Details Page

Based on figure 4.29, the booking details page displays the status of caregiver bookings. It provides clear information on whether a booking has been accepted, is pending confirmation from a caregiver, has been rejected by a caregiver, or has been canceled due to prolonged waiting for a response. This page helps parents to track and manage booking requests efficiently, ensuring transparency and clarity in the booking process.

4.4.2.7 Caregiver Module

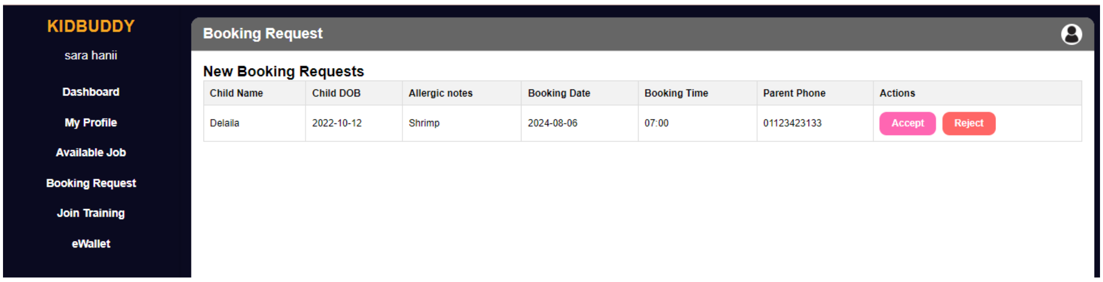


Available Jobs (1)

Parent Name	Parent Phone	Location	Actions
Aniesa Sarah	01123423133	Selangor	Accept Job View Details

Figure 4.31 : Available Job for Caregiver Page

Based on figure 4.30, an available job for caregiver’s page typically lists current job openings specifically for caregivers. This page serves as a platform for caregivers to find employment opportunities that match their skills and preferences.



KIDBUDDY

sara hanii

- Dashboard
- My Profile
- Available Job
- Booking Request
- Join Training
- eWallet

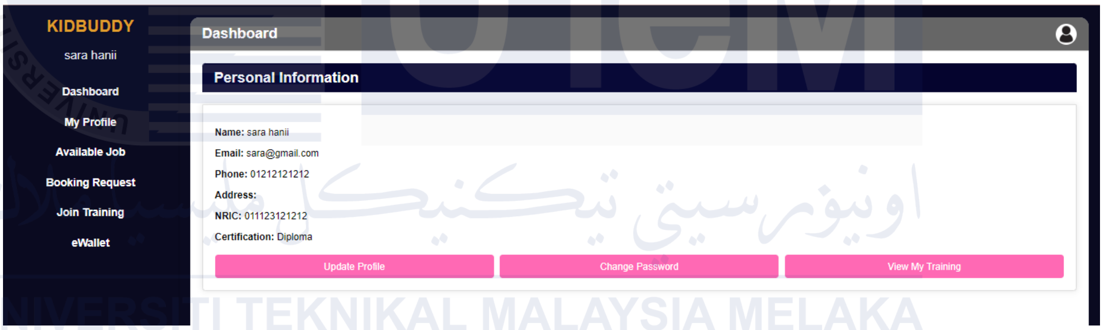
Booking Request

New Booking Requests

Child Name	Child DOB	Allergic notes	Booking Date	Booking Time	Parent Phone	Actions
Delaila	2022-10-12	Shrimp	2024-08-06	07:00	01123423133	Accept Reject

Figure 4.32 : Booking Request from Parent

Based on figure 4.31, booking requests from parent for caregiver page typically notifies caregivers about new requests for caregiving services from parents or guardians. It includes essential details such as the child's information, the booking date, time, and any specific requirements or preferences. This page allows caregivers to review and respond promptly to booking requests.



KIDBUDDY

sara hanii

- Dashboard
- My Profile
- Available Job
- Booking Request
- Join Training
- eWallet

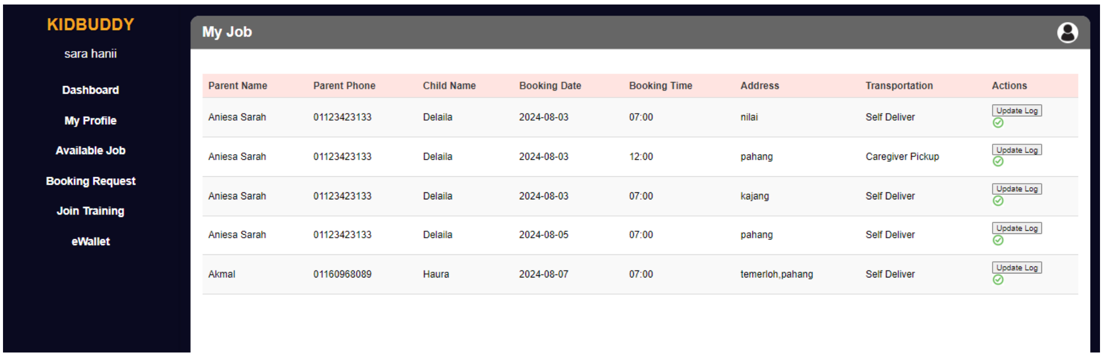
Dashboard

Personal Information

Name: sara hanii
Email: sara@gmail.com
Phone: 01212121212
Address:
NRIC: 011123121212
Certification: Diploma

Figure 4.33 : Profile Caregiver Page

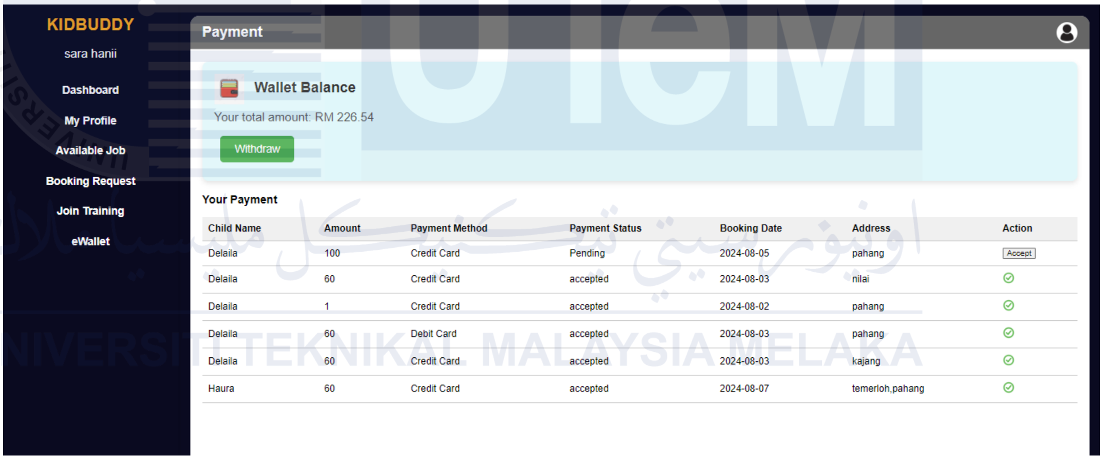
Based on figure 4.32, the profile caregiver page provides detailed information about a caregiver, including their name, phone number, NRIC, certifications, and address. This page is designed to offer a comprehensive overview of the caregiver's credentials and contact information. Additionally, it allows caregivers to update their profile, ensuring that all information remains current and accurate. This functionality helps caregivers maintain an up-to-date and professional profile, which is crucial for building trust and reliability with potential clients.



Parent Name	Parent Phone	Child Name	Booking Date	Booking Time	Address	Transportation	Actions
Aniesa Sarah	01123423133	Delaila	2024-08-03	07:00	nilai	Self Deliver	Update Log ✓
Aniesa Sarah	01123423133	Delaila	2024-08-03	12:00	pahang	Caregiver Pickup	Update Log ✓
Aniesa Sarah	01123423133	Delaila	2024-08-03	07:00	kajang	Self Deliver	Update Log ✓
Aniesa Sarah	01123423133	Delaila	2024-08-05	07:00	pahang	Self Deliver	Update Log ✓
Akmal	01160968089	Haura	2024-08-07	07:00	temerloh,pahang	Self Deliver	Update Log ✓

Figure 4.34 : Caregiver Job History Page

Based on figure 4.33, the caregiver job history page provides a detailed record of a caregiver's past job assignments. This page typically includes information such as the dates of each job, the location, and the child details of each job.



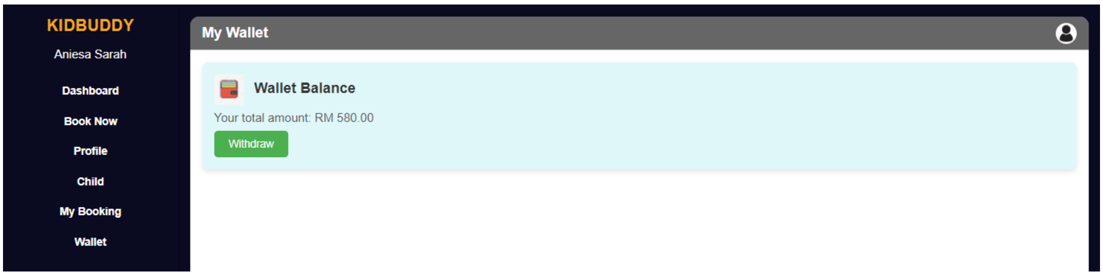
Your total amount: RM 226.54

Your Payment

Child Name	Amount	Payment Method	Payment Status	Booking Date	Address	Action
Delaila	100	Credit Card	Pending	2024-08-05	pahang	Accept
Delaila	60	Credit Card	accepted	2024-08-03	nilai	✓
Delaila	1	Credit Card	accepted	2024-08-02	pahang	✓
Delaila	60	Debit Card	accepted	2024-08-03	pahang	✓
Delaila	60	Credit Card	accepted	2024-08-03	kajang	✓
Haura	60	Credit Card	accepted	2024-08-07	temerloh,pahang	✓

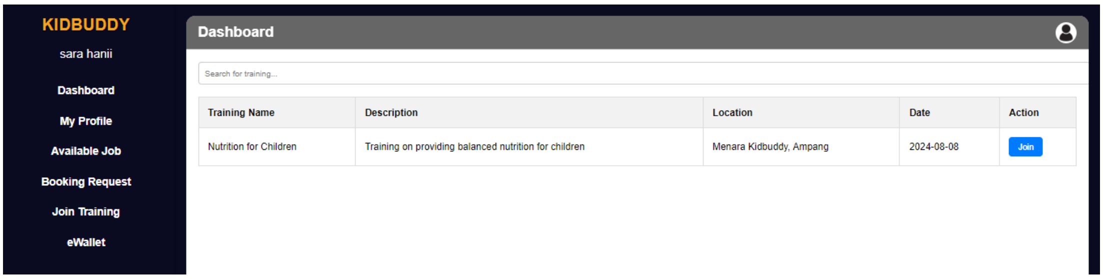
Figure 4.35 : Payout Page

Based on figure 4.34, the payout page provides caregivers with information about their earnings and payment status. This page typically includes details such as the total amount earned, payment method, and the status of each payment such as pending and accepted.



Your total amount: RM 580.00

4.4.2.8 Training Module



Training Name	Description	Location	Date	Action
Nutrition for Children	Training on providing balanced nutrition for children	Menara Kidbuddy, Ampang	2024-08-08	Join

Figure 4.36 : Available Training for Caregiver Page

Based on figure 4.35, the available training for caregiver's page typically offers essential educational resources and training opportunities designed to enhance the skills and knowledge of caregivers. This training is crucial for caregivers to improve their caregiving abilities and ensure they provide high-quality care to their clients.

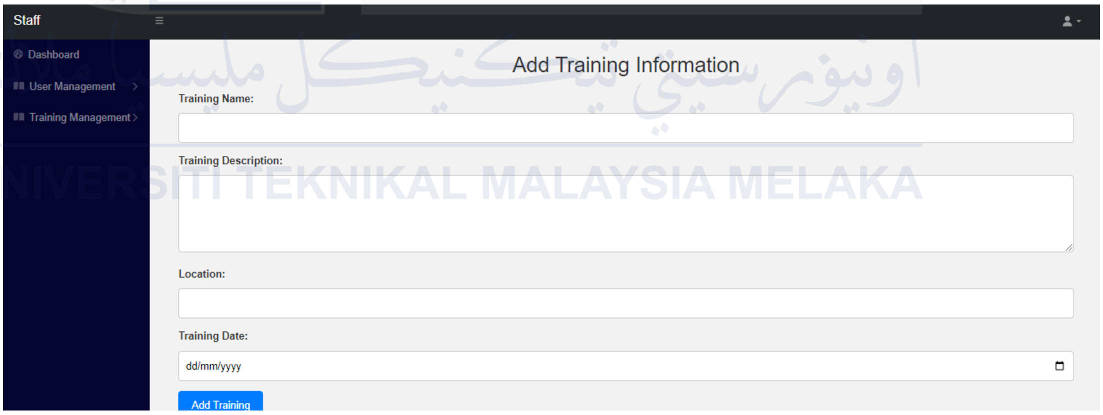
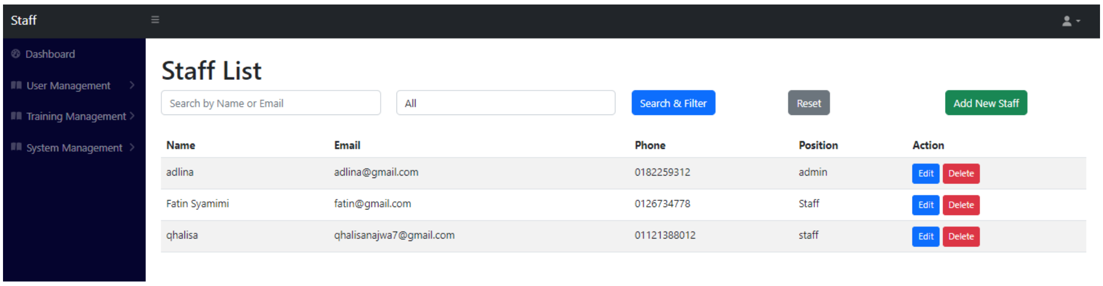


Figure 4.37 : Add Training Information Page

Based on figure 4.36, the added training information page in the staff section allows administrators or staff members to input and update training details for caregivers. This page typically includes fields for entering the training name, description, date, duration, and location.

4.4.2.9 Admin and Staff Module

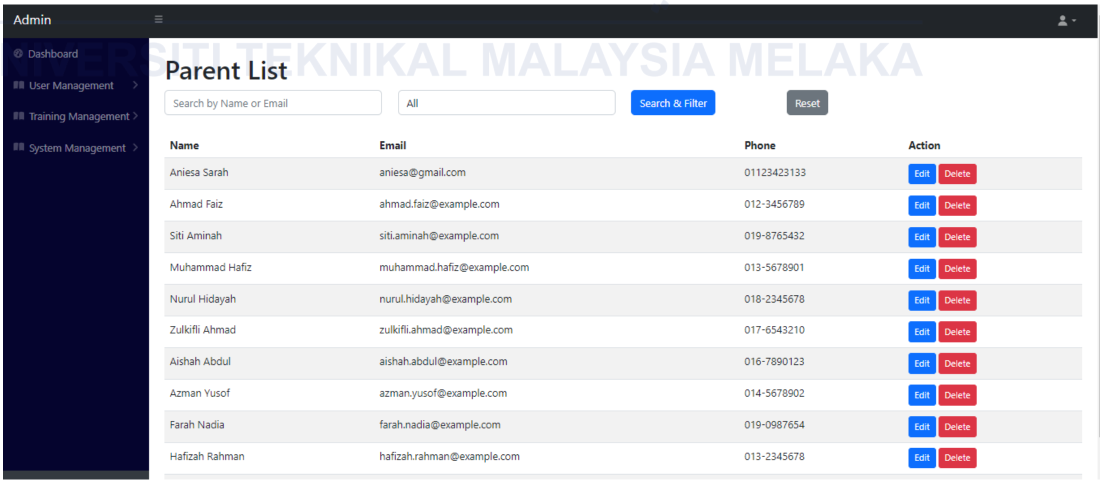


Staff List

Name	Email	Phone	Position	Action
adlina	adlina@gmail.com	0182259312	admin	Edit Delete
Fatin Syamimi	fatin@gmail.com	0126734778	Staff	Edit Delete
qhalisa	qhalisanajwa7@gmail.com	01121388012	staff	Edit Delete

Figure 4.38 : Staff List Page

Based on figure 4.37, the staff list page provides a comprehensive directory of all staff members within an organization, including details such as names, email, position and contact information. It offers functionality for editing and deleting staff information, allowing administrators to keep the directory up-to-date and accurate. Users can edit a staff member's details, such as updating their contact information or position, and remove staff members from the list when they leave the organization. This ensures that the staff directory remains current and reliable, facilitating effective communication and collaboration within the organization.



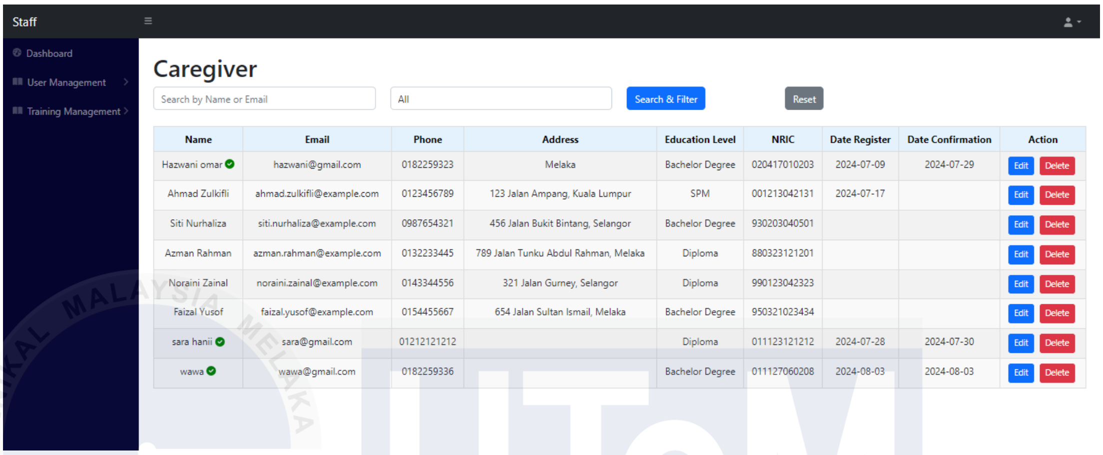
Parent List

Name	Email	Phone	Action
Aniesa Sarah	aniesa@gmail.com	01123423133	Edit Delete
Ahmad Faiz	ahmad.faiz@example.com	012-3456789	Edit Delete
Siti Aminah	siti.aminah@example.com	019-8765432	Edit Delete
Muhammad Hafiz	muhammad.hafiz@example.com	013-5678901	Edit Delete
Nurul Hidayah	nurul.hidayah@example.com	018-2345678	Edit Delete
Zulkifli Ahmad	zulkifli.ahmad@example.com	017-6543210	Edit Delete
Aishah Abdul	aishah.abdul@example.com	016-7890123	Edit Delete
Azman Yusof	azman.yusof@example.com	014-5678902	Edit Delete
Farah Nadia	farah.nadia@example.com	019-0987654	Edit Delete
Hafizah Rahman	hafizah.rahman@example.com	013-2345678	Edit Delete

Figure 4.39 : Parent List Page

Based on figure 4.38, the parent list page provides a detailed directory of all parents or guardians registered within the system. This page typically includes each parent's name, contact information such as phone number, email and address. The page allows administrators to edit and delete entries, ensuring that the information remains accurate and up to date. This functionality helps in managing and maintaining effective communication with parents, ensuring their needs and preferences are well-documented and easily accessible.

.



Caregiver

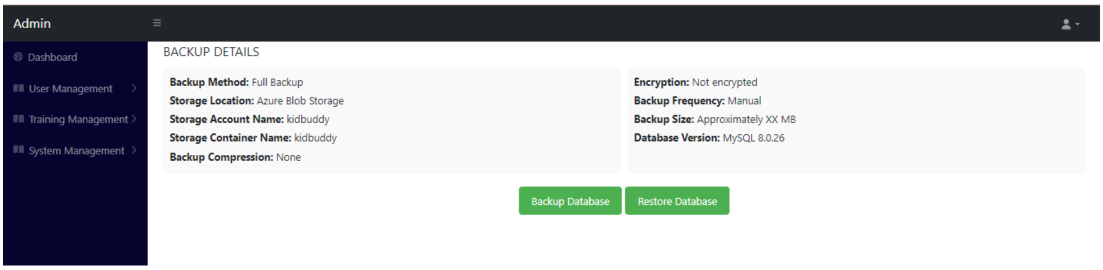
Search by Name or Email

All

Name	Email	Phone	Address	Education Level	NRIC	Date Register	Date Confirmation	Action
Hazwani omar ✓	hazwani@gmail.com	0182259323	Melaka	Bachelor Degree	020417010203	2024-07-09	2024-07-29	Edit Delete
Ahmad Zulkifli	ahmad.zulkifli@example.com	0123456789	123 Jalan Ampang, Kuala Lumpur	SPM	001213042131	2024-07-17		Edit Delete
Siti Nurhaliza	siti.nurhaliza@example.com	0987654321	456 Jalan Bukit Bintang, Selangor	Bachelor Degree	930203040501			Edit Delete
Azman Rahman	azman.rahman@example.com	0132233445	789 Jalan Tunku Abdul Rahman, Melaka	Diploma	880323121201			Edit Delete
Noraini Zainal	noraini.zainal@example.com	0143344556	321 Jalan Gurney, Selangor	Diploma	990123042323			Edit Delete
Faizal Yusof	faizal.yusof@example.com	0154455667	654 Jalan Sultan Ismail, Melaka	Bachelor Degree	950321023434			Edit Delete
sara hanii ✓	sara@gmail.com	01212121212		Diploma	011123121212	2024-07-28	2024-07-30	Edit Delete
wawa ✓	wawa@gmail.com	0182259336		Bachelor Degree	011127060208	2024-08-03	2024-08-03	Edit Delete

Figure 4.40 : Caregiver List Page

Based on figure 4.39, the caregiver list page provides a comprehensive directory of all caregivers registered within the system. This page typically includes each caregiver's name, contact information such as phone number, email and address. Administrators can edit and delete entries, allowing them to keep the information accurate and up to date.



BACKUP DETAILS

Backup Method: Full Backup
Storage Location: Azure Blob Storage
Storage Account Name: kidbuddy
Storage Container Name: kidbuddy
Backup Compression: None

Encryption: Not encrypted
Backup Frequency: Manual
Backup Size: Approximately XX MB
Database Version: MySQL 8.0.26

Figure 4.41 : System Management Page

Based on figure 4.40, the system management page provides administrators with tools to maintain the system's integrity and data security, featuring both backup and restore buttons. The backup button allows administrators to create secure copies of current system data, ensuring critical information is safely stored for recovery in case of data loss or system failure. The restore button enables reverting the system to a previous state using a backup, essential for recovering from errors or data corruption.

4.5 Conclusion

Chapter 4 design has provided a comprehensive design framework for the childcare management system, addressing key components essential for creating an intuitive and efficient user interface. By detailing the navigation flow, we established a clear structure for accessing various sections of the application, ensuring users can seamlessly navigate through the system. The input design emphasized creating user-friendly forms with clear labels and validation mechanisms to minimize errors and enhance data accuracy. Output design focused on presenting critical information through well-designed dashboard widgets and customizable views, enabling users to tailor data displays to their specific needs and preferences. Additionally, the implementation of real-time updates and notifications ensures that users remain informed and engaged with the latest system developments. These design elements collectively aim to provide a robust, user-centric interface that meets both functional and non-functional requirements, ultimately enhancing the overall user experience and operational efficiency of the childcare management system.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

This chapter discusses the implementation phase of the Kidbuddy Management System project. The implementation phase involves translating the design into a functioning system through coding, configuration, and deployment. The expected output of this phase is a fully operational system that meets the defined requirements, including features such as user registration, booking management, activity tracking, payment processing, and communication between parents, caregivers, and staffs.

5.2 Software Development Environment Setup

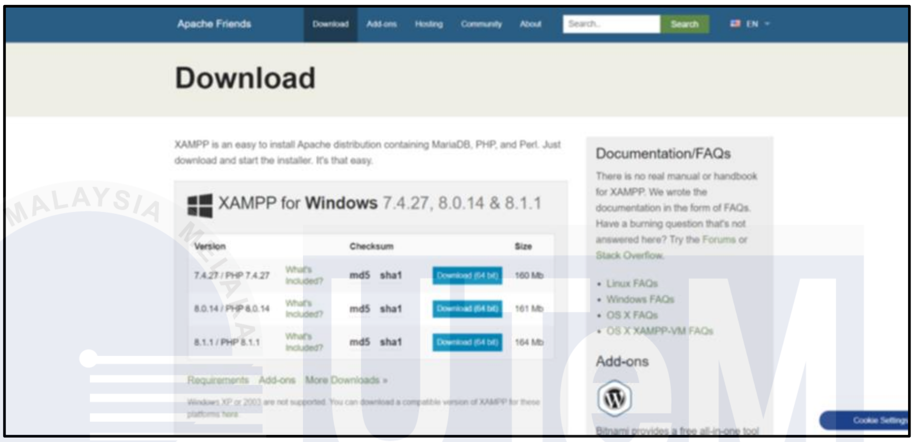
Setting up the software development environment is a critical step in ensuring a smooth and efficient development process. This phase involves configuring the necessary tools and platforms that support the design, implementation, and management of the child day care management system. Key components include XAMPP for local server and database management, Visual Studio Code for code development and debugging, Microsoft Azure for secure cloud-based database backups, and Power BI for data visualization and reporting. Together, these tools create a robust environment that facilitates the seamless integration of development, testing, and deployment activities, ensuring that the project is built on a solid foundation.

5.2.1 System and Database Installation Setup

a. XAMPP

1. Step 1: Download XAMPP Package Installer from the website.

   Link -https://www.microsoft.com/en-us/download/details.aspx?id=58494



Download

XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy.

XAMPP for **Windows** 7.4.27, 8.0.14 & 8.1.1

Version		Checksum			Size
7.4.27 / PHP 7.4.27	What's Included?	md5	sha1	Download (64 bit)	160 Mb
8.0.14 / PHP 8.0.14	What's Included?	md5	sha1	Download (64 bit)	161 Mb
8.1.1 / PHP 8.1.1	What's Included?	md5	sha1	Download (64 bit)	164 Mb

Requirements Add-ons More Downloads »

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms here.

Documentation/FAQs

There is no real manual or handbook for XAMPP. We wrote the documentation in the form of FAQs. Have a burning question that's not answered here? Try the Forums or Stack Overflow.

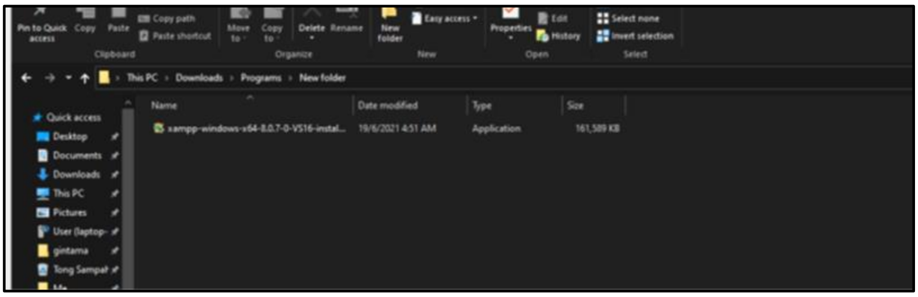
- Linux FAQs
- Windows FAQs
- OS X FAQs
- OS X XAMPP-VM FAQs

Add-ons

Bitnami provides a free all-in-one tool

Figure 5.1 : Download Page

2. Step 2: Locate the download destination folder and after that, make sure antivirus software that in laptop has been turn off before run the XAMPP.

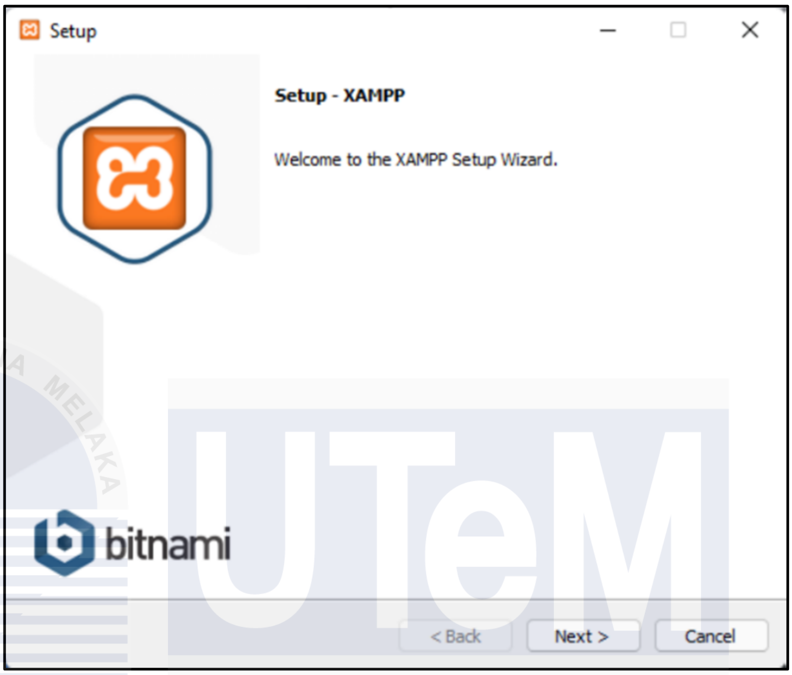


This PC › Downloads › Programs › New folder

Name	Date modified	Type	Size
xampp-windows-x64-8.0.7-0-VS16-instal...	19/6/2021 4:51 AM	Application	161,589 KB

Figure 5.2 : Location of XAMPP application after download

3. Click the installation package until the window above appears and click next.



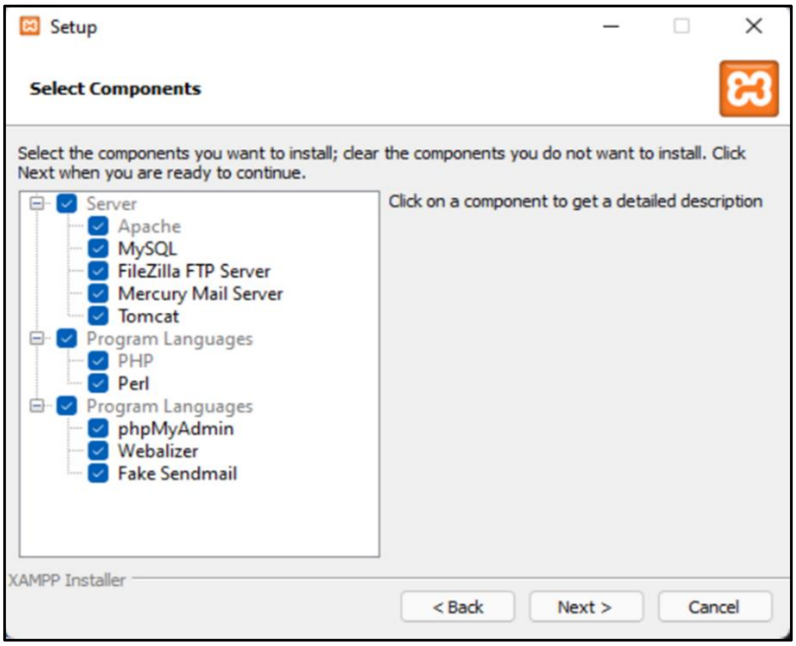
Setup

Setup - XAMPP

Welcome to the XAMPP Setup Wizard.

Figure 5.3 : Setup Page

4. Make sure all the boxes are checked and click next.



Setup

Select Components

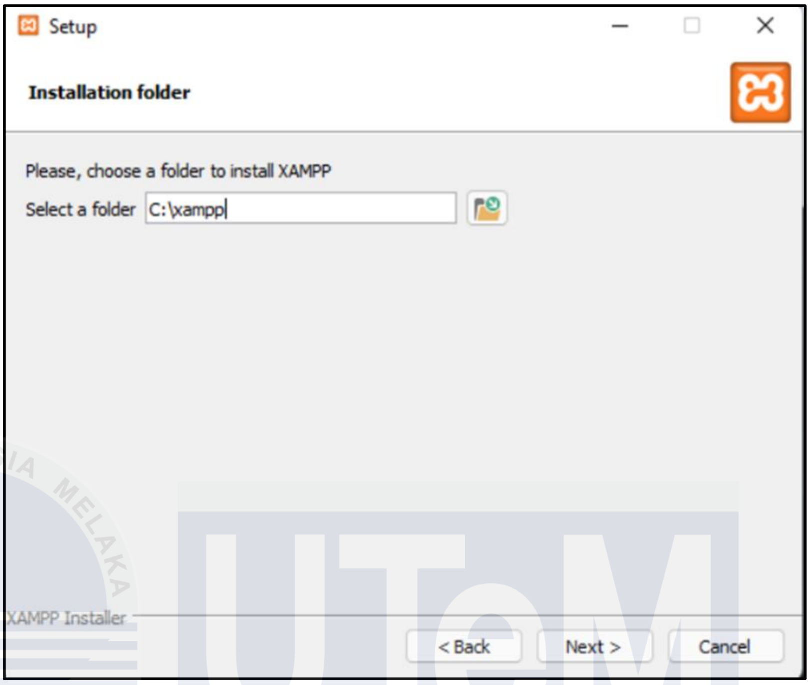
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- ☑ Server
  - ☑ Apache
  - ☑ MySQL
  - ☑ FileZilla FTP Server
  - ☑ Mercury Mail Server
  - ☑ Tomcat
- ☑ Program Languages
  - ☑ PHP
  - ☑ Perl
- ☑ Program Languages
  - ☑ phpMyAdmin
  - ☑ Webalizer
  - ☑ Fake Sendmail

Click on a component to get a detailed description

Figure 5.4 : Select Component Page

5. Choose the file destination folder for the installation and click Next.



Setup

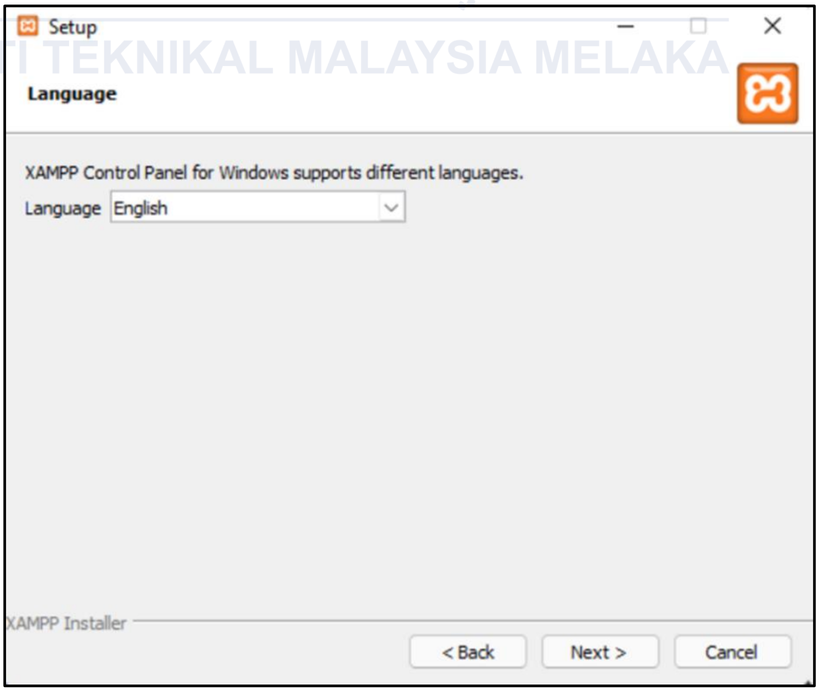
Installation folder

Please, choose a folder to install XAMPP

Select a folder C:\xampp

Figure 5.5 : Installation Folder Page

6. Choose preferred language and click Next.



Setup

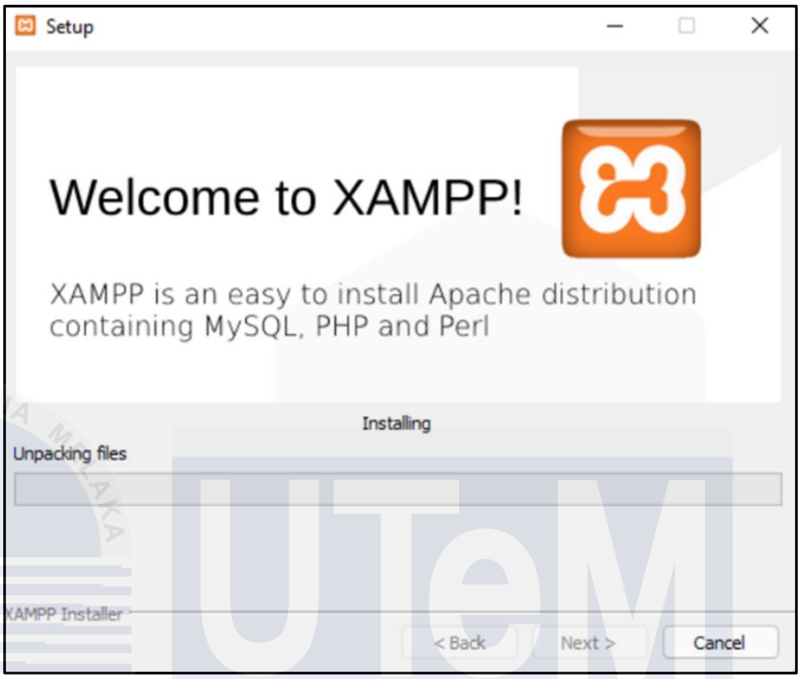
Language

XAMPP Control Panel for Windows supports different languages.

Language English

Figure 5.6 : Language Page

7. The download progress bar will appear indicating the installation process has begun.



Setup

Welcome to XAMPP!

XAMPP is an easy to install Apache distribution containing MySQL, PHP and Perl

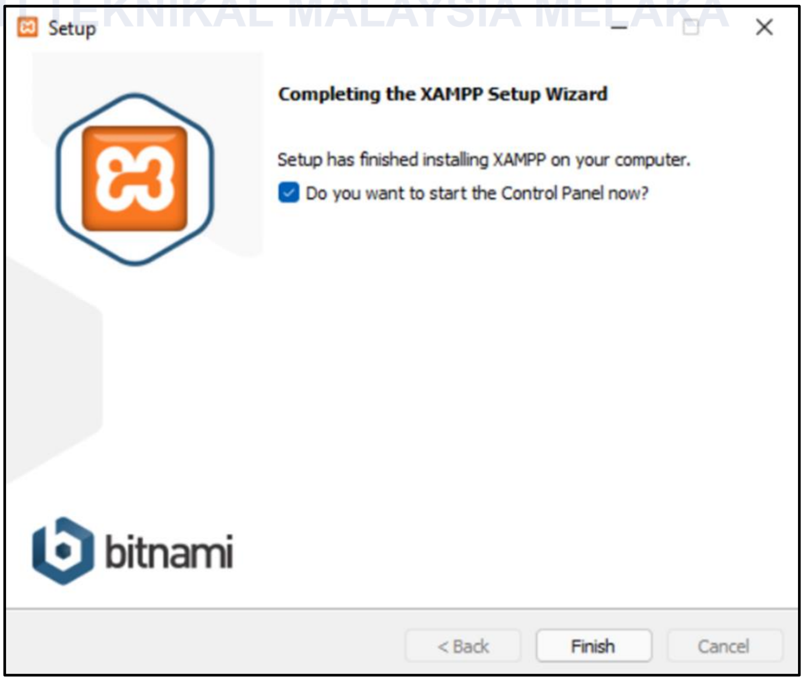
Installing

Unpacking files

XAMPP Installer

Figure 5.7 : Installation in Progress

8. Click Finish.



Setup

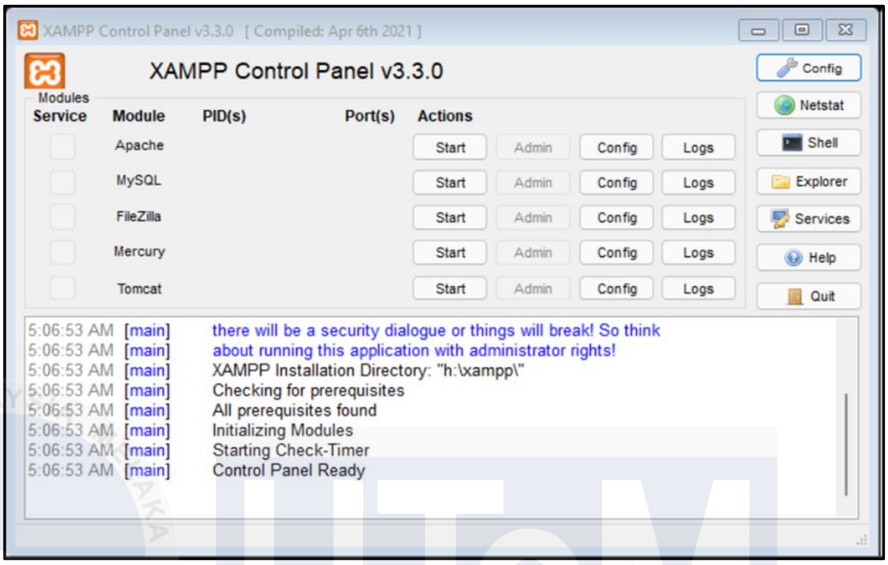
Completing the XAMPP Setup Wizard

Setup has finished installing XAMPP on your computer.

☑ Do you want to start the Control Panel now?

Figure 5.8 : Completing the XAMPP Setup Wizard Page

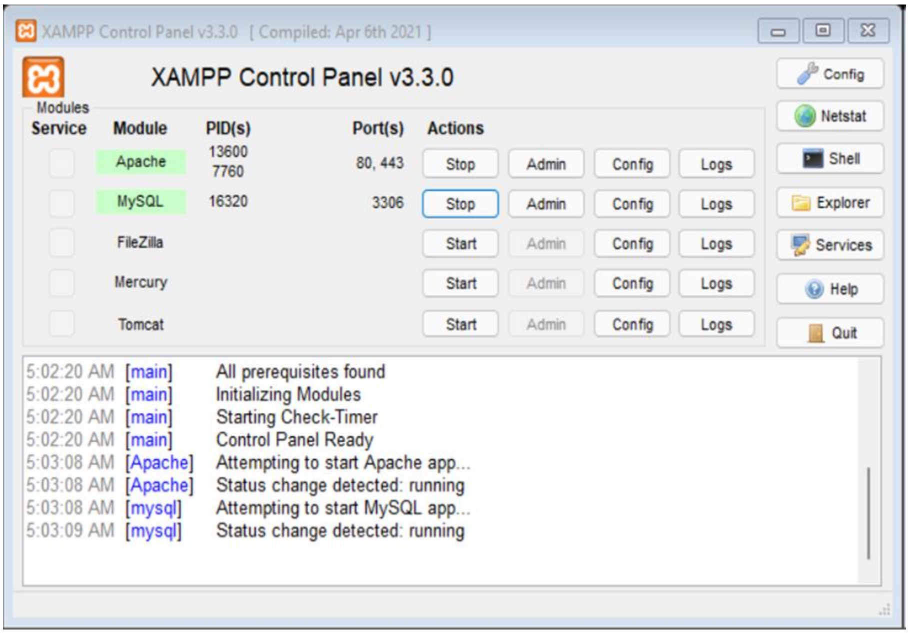
9. After the installation is done. Open XAMPP and the main window will as show below.



```
5:06:53 AM  [main]   there will be a security dialogue or things will break! So think
5:06:53 AM  [main]   about running this application with administrator rights!
5:06:53 AM  [main]   XAMPP Installation Directory: "h:\xampp\"
5:06:53 AM  [main]   Checking for prerequisites
5:06:53 AM  [main]   All prerequisites found
5:06:53 AM  [main]   Initializing Modules
5:06:53 AM  [main]   Starting Check-Timer
5:06:53 AM  [main]   Control Panel Ready
```

Figure 5.9 : Control Panel

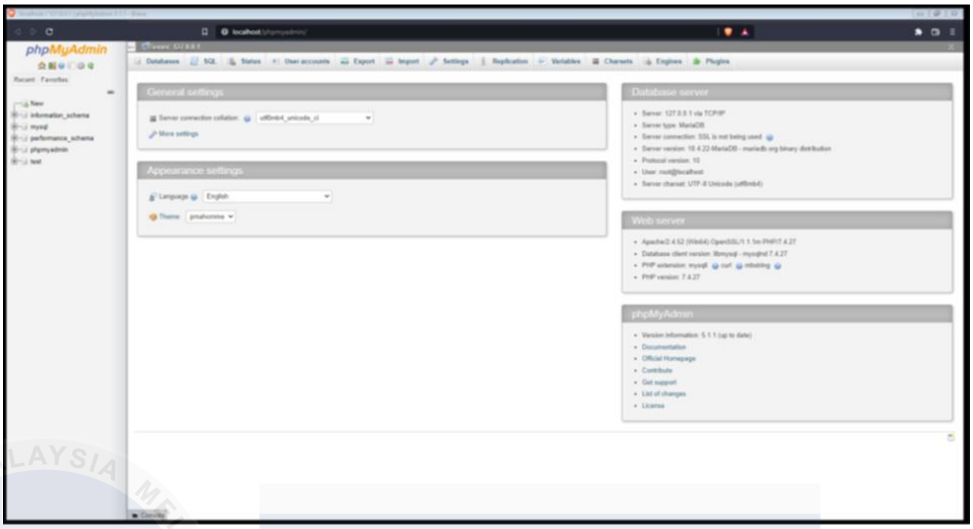
10. Click the start button on Apache and MySQL to start. Click on MySQL Admin button to go to PhpMyAdmin page.



```
5:02:20 AM  [main]     All prerequisites found
5:02:20 AM  [main]     Initializing Modules
5:02:20 AM  [main]     Starting Check-Timer
5:02:20 AM  [main]     Control Panel Ready
5:03:08 AM  [Apache]   Attempting to start Apache app...
5:03:08 AM  [Apache]   Status change detected: running
5:03:08 AM  [mysql]    Attempting to start MySQL app...
5:03:09 AM  [mysql]    Status change detected: running
```

Figure 5.10 : Start MySQL and Apache

11. The localhost will appear on browser as shown above.



General settings

- Server connection collation: utf8mb4_unicode_ci
- More settings

Appearance settings

- Language: English
- Theme: pmahomme

Database server

- Server: 127.0.0.1 via TCP/IP
- Server type: MariaDB
- Server connection: SSL is not being used
- Server version: 10.4.22-MariaDB - mariadb.org binary distribution
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8mb4)

Web server

- Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/7.4.27
- Database client version: libmysql - mysqlnd 7.4.27
- PHP extension: mysqli curl mbstring
- PHP version: 7.4.27

phpMyAdmin

- Version information: 5.1.1 (up to date)
- Documentation
- Official Homepage
- Contribute
- Get support
- List of changes
- License

Figure 5.11 : phpMyAdmin

b. Microsoft Azure

1. Login to microsoft azure

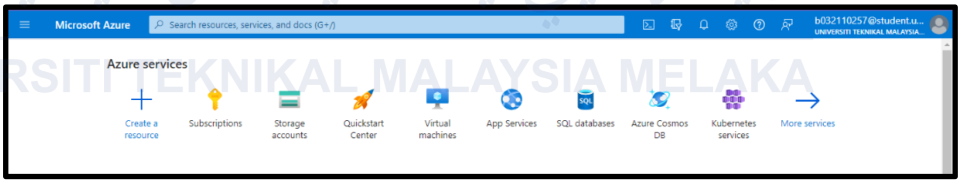


Figure 5.12 : Microsoft Azure homepage

2. Click storage account and create

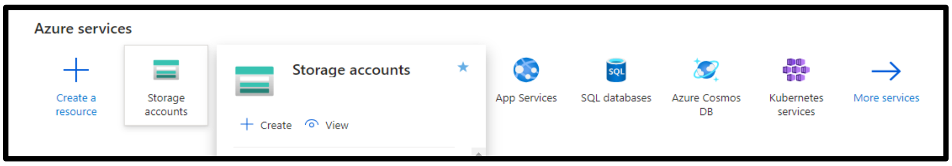
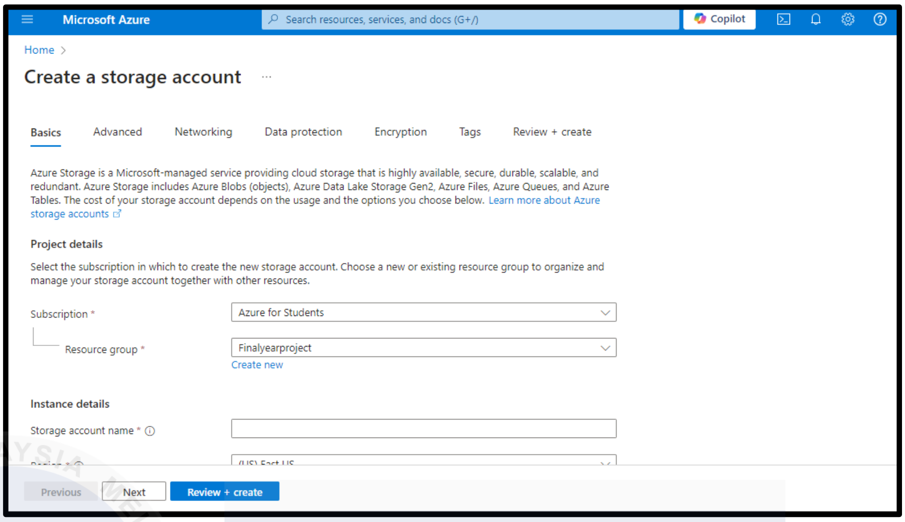


Figure 5.13 : Storage accounts in Azure homepage

3. Fill the details of storage account, next click reviews then create.



Create a storage account

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. Learn more about Azure storage accounts

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription * Azure for Students

Resource group * Finalyearproject

Create new

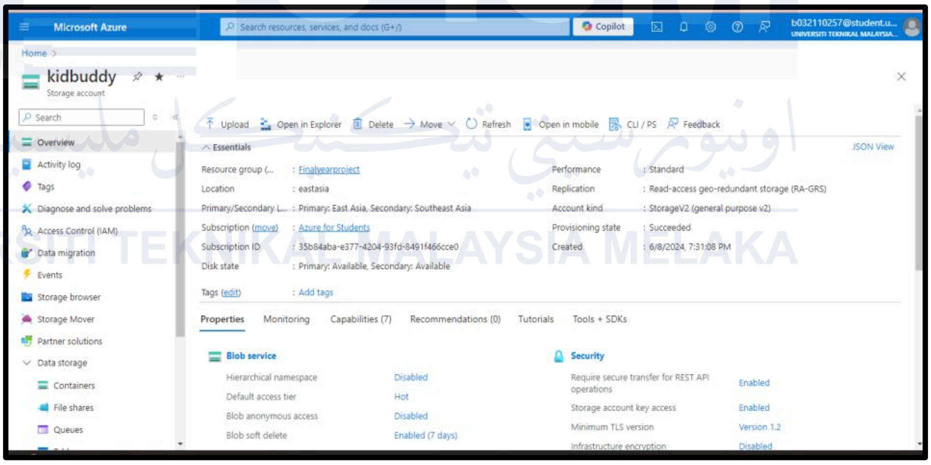
Instance details

Storage account name *

Region * (US) East US

Figure 5.14 : Create storage accounts page

4. The storage account successfully created then click containers under data storage.



kidbuddy
Storage account

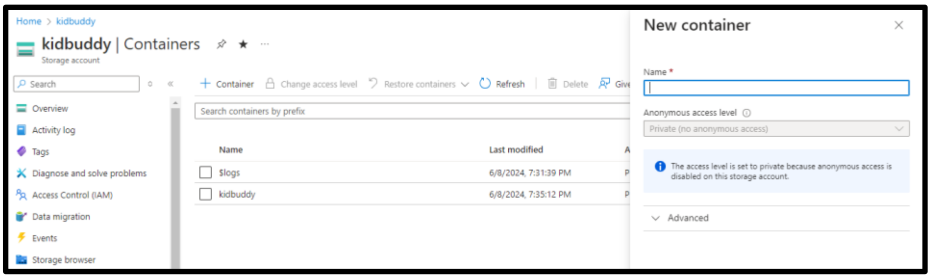
Essentials

Resource group (...	: Finalyearproject	Performance	: Standard
Location	: eastasia	Replication	: Read-access geo-redundant storage (RA-GRS)
Primary/Secondary L...	: Primary: East Asia, Secondary: Southeast Asia	Account kind	: StorageV2 (general purpose v2)
Subscription (move)	: Azure for Students	Provisioning state	: Succeeded
Subscription ID	: 35b84aba-e377-4204-93fd-8491f466cce0	Created	: 6/8/2024, 7:31:08 PM
Disk state	: Primary: Available, Secondary: Available		
Tags (edit)	: Add tags		

Blob service		Security	
Hierarchical namespace	Disabled	Require secure transfer for REST API operations	Enabled
Default access tier	Hot	Storage account key access	Enabled
Blob anonymous access	Disabled	Minimum TLS version	Version 1.2
Blob soft delete	Enabled (7 days)	Infrastructure encryption	Disabled

Figure 5.15 : Overview storage accounts kidbuddy

5. Click create and fill in the details then click create.



kidbuddy | Containers
Storage account

Name	Last modified
$logs	6/8/2024, 7:31:39 PM
kidbuddy	6/8/2024, 7:35:12 PM

New container

Name *

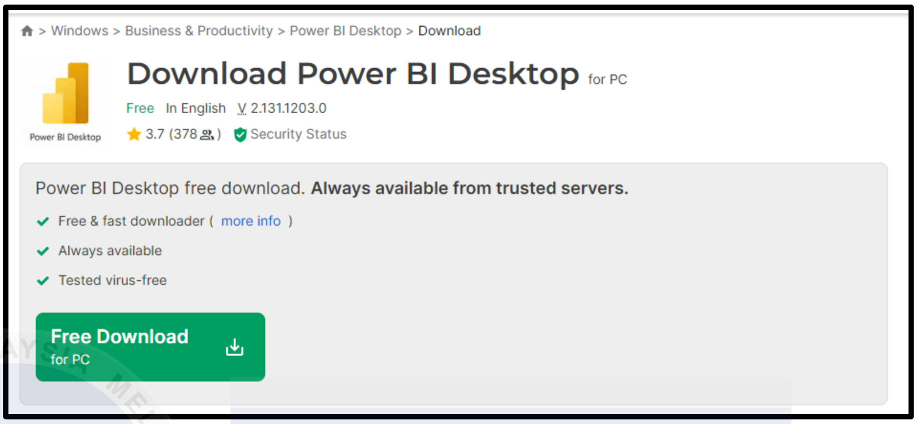
Anonymous access level: Private (no anonymous access)

The access level is set to private because anonymous access is disabled on this storage account.

Figure 5.16 : Containers of kidbuddy

c. Power Bi

1. Download Power Bi Desktop from the website.



Home > Windows > Business & Productivity > Power BI Desktop > Download

Download Power BI Desktop for PC

Free In English V 2.131.1203.0

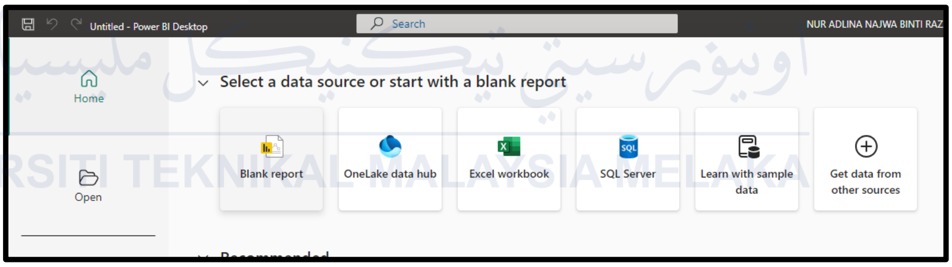
★ 3.7 (378) Security Status

Power BI Desktop free download. **Always available from trusted servers.**

- ✓ Free & fast downloader (more info)
- ✓ Always available
- ✓ Tested virus-free

Figure 5.17 : Download Power Bi page

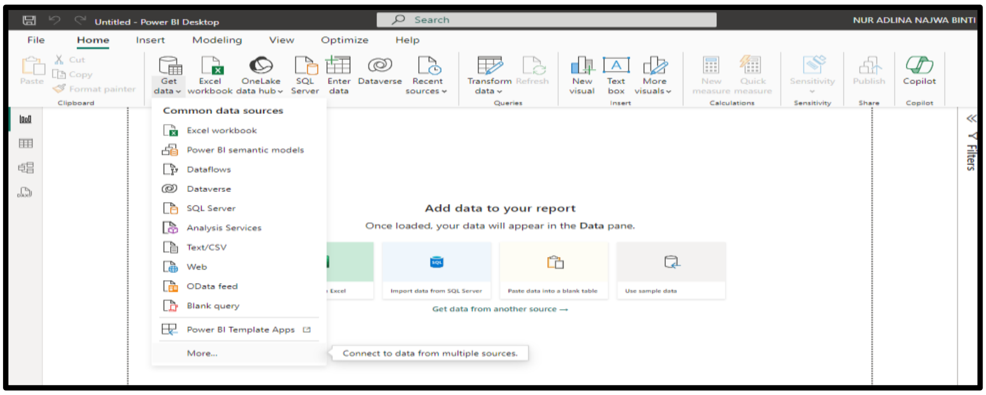
2. Click blank report to start creating a dashboard



Select a data source or start with a blank report

Figure 5.18 : Power BI Desktop homepage

3. Go to home then choose get data, click more



Common data sources

- Excel workbook
- Power BI semantic models
- Dataflows
- Dataverse
- SQL Server
- Analysis Services
- Text/CSV
- Web
- OData feed
- Blank query
- Power BI Template Apps
- More...

Connect to data from multiple sources.

Add data to your report

Once loaded, your data will appear in the **Data** pane.

Excel
Import data from SQL Server
Paste data into a blank table
Use sample data
Get data from another source →

Figure 5.19 : Get data in the home

4. Choose Database then click MySQL then click connect

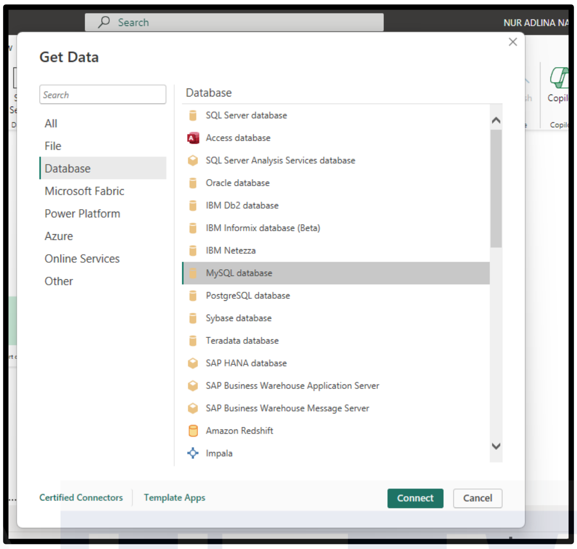
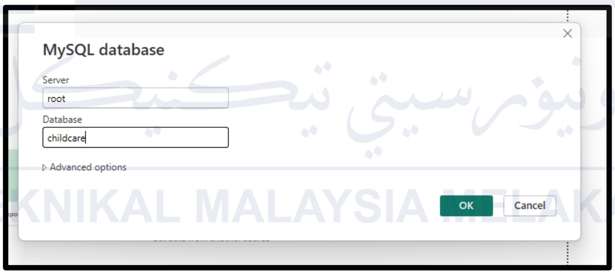


Figure 5.20 : Choose MySQL database to import database from MySQL

5. Insert server name and database name then click ok

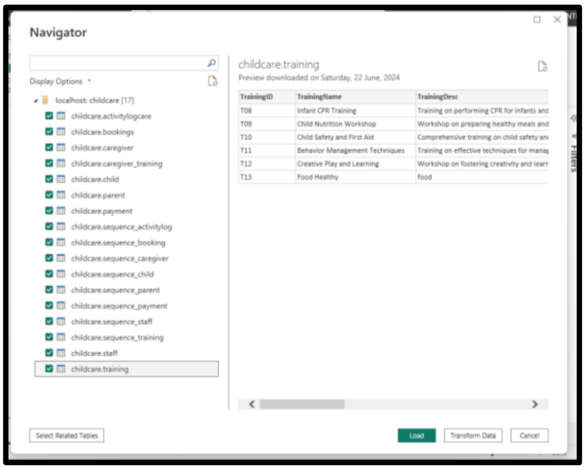


Server
root

Database
childcare

Figure 5.21 : MySQL information

6. Select all the table then click load



localhost: childcare [17]
- ☑ childcare.activitylogcare
- ☑ childcare.bookings
- ☑ childcare.caregiver
- ☑ childcare.caregiver_training
- ☑ childcare.child
- ☑ childcare.parent
- ☑ childcare.payment
- ☑ childcare.sequence_activitylog
- ☑ childcare.sequence_booking
- ☑ childcare.sequence_caregiver
- ☑ childcare.sequence_child
- ☑ childcare.sequence_parent
- ☑ childcare.sequence_payment
- ☑ childcare.sequence_staff
- ☑ childcare.sequence_training
- ☑ childcare.staff
- ☑ childcare.training

childcare.training

Preview downloaded on Saturday, 22 June, 2024

TrainingID	TrainingName	TrainingDesc
T08	Infant CPR Training	Training on performing CPR for infants and
T09	Child Nutrition Workshop	Workshop on preparing healthy meals and
T10	Child Safety and First Aid	Comprehensive training on child safety an
T11	Behavior Management Techniques	Training on effective techniques for manag
T12	Creative Play and Learning	Workshop on fostering creativity and learr
T13	Food Healthy	food

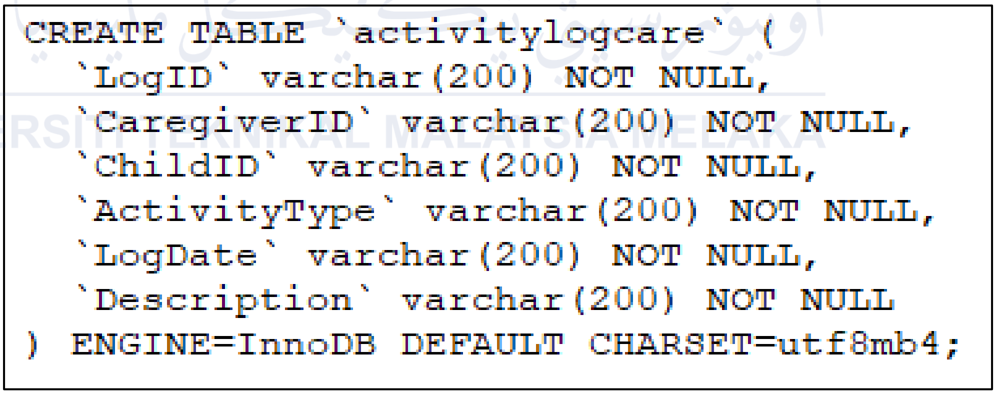
Figure 5.22 : List of MySQL table in the database

5.3 Database Implementation

Database implementation is a critical phase in developing a comprehensive management system, ensuring that the database structure is effectively set up and integrated with the application. This process involves defining the database schema using Data Definition Language (DDL) and Data Control Language (DCL) statements to create and manage database objects such as tables, indexes, and permissions. Additionally, the implementation includes developing stored procedures and triggers to automate and streamline complex processes, enhancing data integrity and operational efficiency. The data loading process populates the database with initial test data to verify system functionality and performance. This phase is essential for ensuring that the database supports the application's requirements and operates seamlessly in real-world scenarios.

5.3.1 Data Definition Language (DDL)

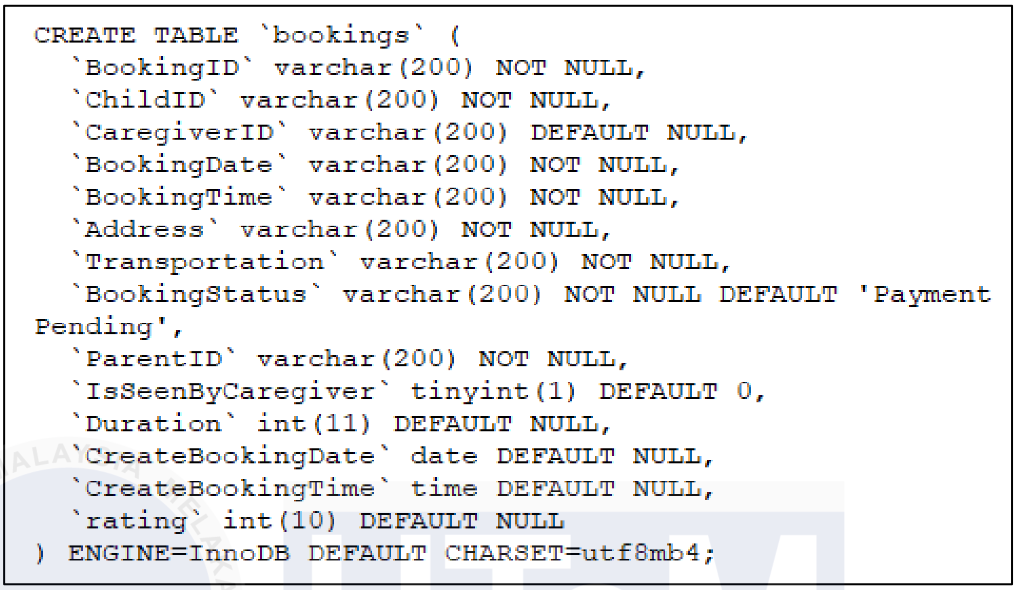
1. Table activitylogcare



```
CREATE TABLE `activitylogcare` (
  `LogID` varchar(200) NOT NULL,
  `CaregiverID` varchar(200) NOT NULL,
  `ChildID` varchar(200) NOT NULL,
  `ActivityType` varchar(200) NOT NULL,
  `LogDate` varchar(200) NOT NULL,
  `Description` varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Figure 5.23 : DDL 'activitylogcare'

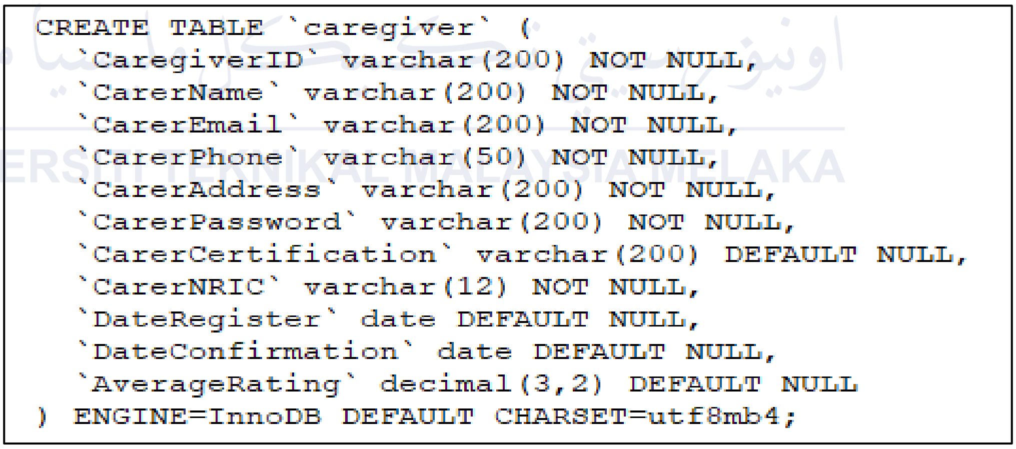
2. Table bookings



```
CREATE TABLE `bookings` (
  `BookingID` varchar(200) NOT NULL,
  `ChildID` varchar(200) NOT NULL,
  `CaregiverID` varchar(200) DEFAULT NULL,
  `BookingDate` varchar(200) NOT NULL,
  `BookingTime` varchar(200) NOT NULL,
  `Address` varchar(200) NOT NULL,
  `Transportation` varchar(200) NOT NULL,
  `BookingStatus` varchar(200) NOT NULL DEFAULT 'Payment
Pending',
  `ParentID` varchar(200) NOT NULL,
  `IsSeenByCaregiver` tinyint(1) DEFAULT 0,
  `Duration` int(11) DEFAULT NULL,
  `CreateBookingDate` date DEFAULT NULL,
  `CreateBookingTime` time DEFAULT NULL,
  `rating` int(10) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Figure 5.24 : DDL ‘bookings’

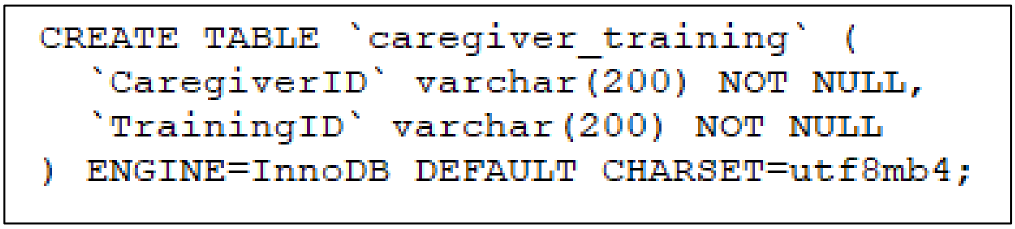
3. Table caregiver



```
CREATE TABLE `caregiver` (
  `CaregiverID` varchar(200) NOT NULL,
  `CarerName` varchar(200) NOT NULL,
  `CarerEmail` varchar(200) NOT NULL,
  `CarerPhone` varchar(50) NOT NULL,
  `CarerAddress` varchar(200) NOT NULL,
  `CarerPassword` varchar(200) NOT NULL,
  `CarerCertification` varchar(200) DEFAULT NULL,
  `CarerNRIC` varchar(12) NOT NULL,
  `DateRegister` date DEFAULT NULL,
  `DateConfirmation` date DEFAULT NULL,
  `AverageRating` decimal(3,2) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Figure 5.25 : DDL ‘caregiver’

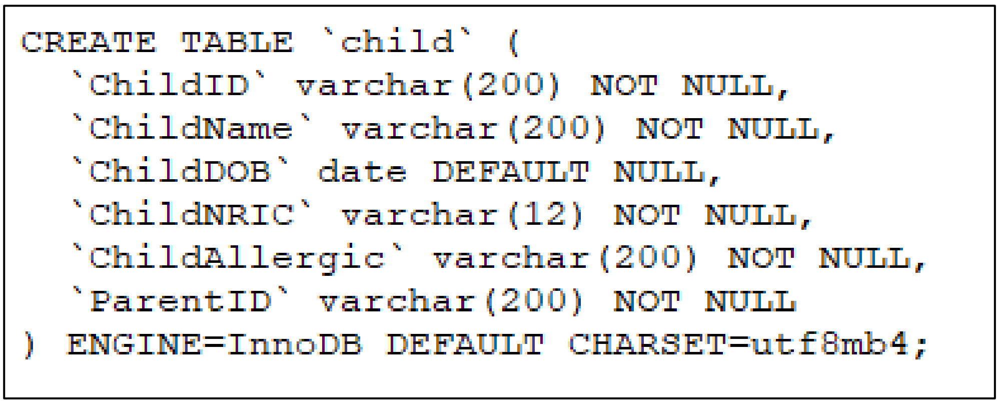
4. Table caregiver_training



```
CREATE TABLE `caregiver_training` (
  `CaregiverID` varchar(200) NOT NULL,
  `TrainingID` varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Figure 5.26 : DDL ‘caregiver_training’

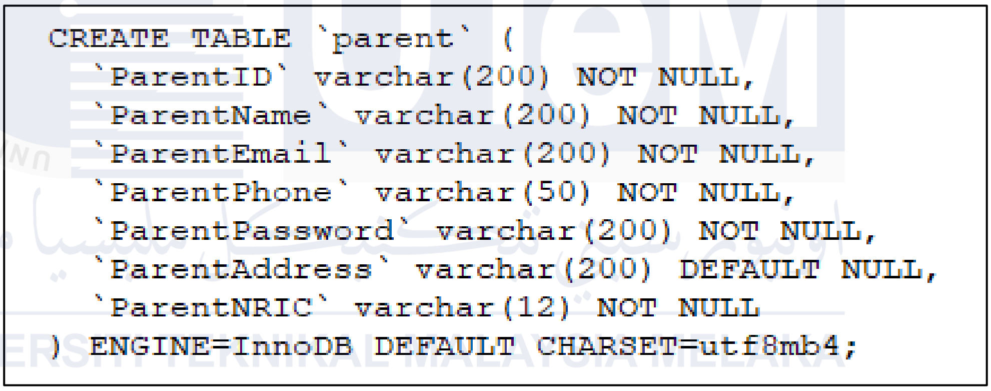
5. Table child



```
CREATE TABLE `child` (
  `ChildID` varchar(200) NOT NULL,
  `ChildName` varchar(200) NOT NULL,
  `ChildDOB` date DEFAULT NULL,
  `ChildNRIC` varchar(12) NOT NULL,
  `ChildAllergic` varchar(200) NOT NULL,
  `ParentID` varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Figure 5.27 : DDL ‘child’

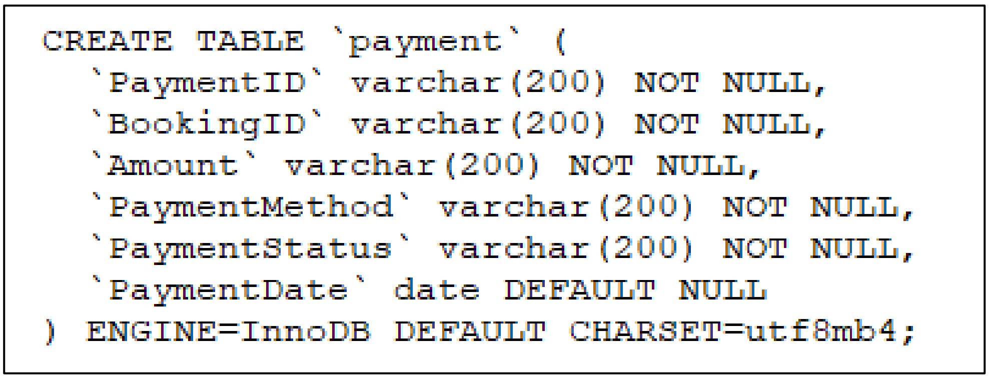
6. Table parent



```
CREATE TABLE `parent` (
  `ParentID` varchar(200) NOT NULL,
  `ParentName` varchar(200) NOT NULL,
  `ParentEmail` varchar(200) NOT NULL,
  `ParentPhone` varchar(50) NOT NULL,
  `ParentPassword` varchar(200) NOT NULL,
  `ParentAddress` varchar(200) DEFAULT NULL,
  `ParentNRIC` varchar(12) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Figure 5.28 : DDL ‘parent’

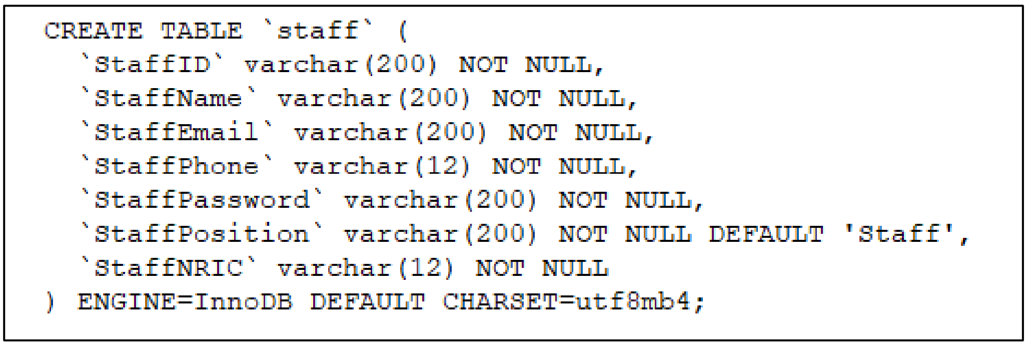
7. Table payment



```
CREATE TABLE `payment` (
  `PaymentID` varchar(200) NOT NULL,
  `BookingID` varchar(200) NOT NULL,
  `Amount` varchar(200) NOT NULL,
  `PaymentMethod` varchar(200) NOT NULL,
  `PaymentStatus` varchar(200) NOT NULL,
  `PaymentDate` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Figure 5.29 : DDL ‘payment’

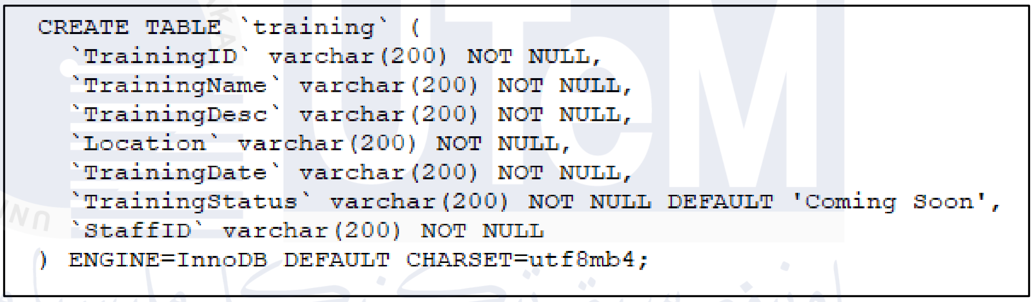
8. Table staff



```
CREATE TABLE `staff` (
  `StaffID` varchar(200) NOT NULL,
  `StaffName` varchar(200) NOT NULL,
  `StaffEmail` varchar(200) NOT NULL,
  `StaffPhone` varchar(12) NOT NULL,
  `StaffPassword` varchar(200) NOT NULL,
  `StaffPosition` varchar(200) NOT NULL DEFAULT 'Staff',
  `StaffNRIC` varchar(12) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Figure 5.30 : DDL ‘staff’

9. Table training



```
CREATE TABLE `training` (
  `TrainingID` varchar(200) NOT NULL,
  `TrainingName` varchar(200) NOT NULL,
  `TrainingDesc` varchar(200) NOT NULL,
  `Location` varchar(200) NOT NULL,
  `TrainingDate` varchar(200) NOT NULL,
  `TrainingStatus` varchar(200) NOT NULL DEFAULT 'Coming Soon',
  `StaffID` varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Figure 5.31 : DDL ‘training’

5.3.2 Implementation of Main Processes

1. Stored Procedure

- **Caregiver Login Procedure**

  The caregiver_login stored procedure verifies a caregiver's credentials. It checks if the provided email and password match an existing caregiver in the database.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `caregiver_login`
(IN `p_username` VARCHAR(200), IN `p_password` VARCHAR(200))
BEGIN
    DECLARE user_count INT;

    -- Check if user is a parent
    SELECT COUNT(*) INTO user_count
    FROM caregiver
    WHERE CarerEmail = p_username AND CarerPassword =
p_password;

    IF user_count = 1 THEN
        -- Login successful
        SELECT 'Login successful' AS message;
    ELSE
        -- Login failed
        SELECT 'Invalid email or password' AS message;
    END IF;
END$$
```

Figure 5.32 : SQL stored procedure caregiver_login

2. Trigger

- **Trigger for 'bookings' table**

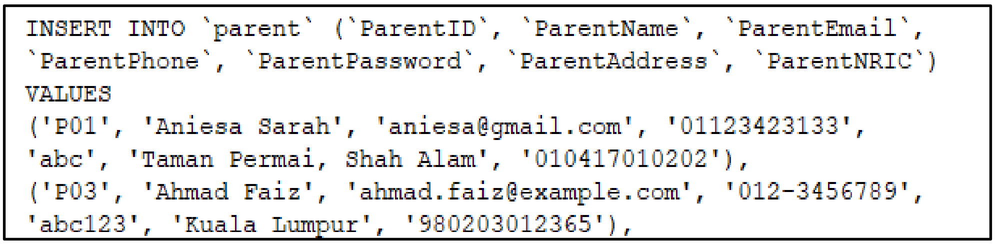
  The bookingIDincrement trigger automatically generates a unique BookingID for each new booking inserted into the bookings table. It uses the sequence_booking table to maintain a sequence of unique IDs.

```
DELIMITER $$
CREATE TRIGGER `bookingIDincrement` BEFORE INSERT ON `bookings`
FOR EACH ROW BEGIN
INSERT into sequence_booking VALUES (NULL);
SET NEW.BookingID=CONCAT('B',LPAD (LAST_INSERT_ID (),2,'0'));
END
$$
DELIMITER ;
```

Figure 5.33 : SQL trigger 'bookingIDincrement'

5.3.3 Data Loading Process

The data loading process for the database involves populating it with initial test data to ensure proper functionality and performance of the system. This is achieved through SQL INSERT statements, which create records in the database tables with predefined values. For example, the parent table was populated with sample data using the following SQL script:



```
INSERT INTO `parent` (`ParentID`, `ParentName`, `ParentEmail`,
`ParentPhone`, `ParentPassword`, `ParentAddress`, `ParentNRIC`)
VALUES
('P01', 'Aniesa Sarah', 'aniesa@gmail.com', '01123423133',
'abc', 'Taman Permai, Shah Alam', '010417010202'),
('P03', 'Ahmad Faiz', 'ahmad.faiz@example.com', '012-3456789',
'abc123', 'Kuala Lumpur', '980203012365'),
```

Figure 5.34 : SQL script INSERT statement table ‘parent’

In this script, each INSERT statement adds a new record into the parent table. The ParentID serves as a unique identifier for each record, while the other fields such as ParentName, ParentEmail, ParentPhone, ParentPassword, ParentAddress, and ParentNRIC are populated with sample data. This test data helps verify that the database schema is correctly defined, and that the application can interact with the database as expected. Additionally, it allows for testing database queries, application logic, and user interface components before deploying the system with real data.

5.4 Conclusion

The implementation phase of the Kidbuddy Management System project is pivotal in transforming design concepts into a functional system, encompassing coding, configuration, and deployment. By establishing a robust software development environment with tools such as XAMPP, Visual Studio Code, Microsoft Azure, and Power BI, the project ensures a solid foundation for development and testing. The database implementation, involving DDL and DCL statements, and the creation of stored procedures and triggers, is crucial for maintaining data integrity and operational efficiency. The data loading process, utilizing SQL INSERT statements, populates the database with initial test data, validating the system’s functionality and performance. This comprehensive approach ensures that the system meets its requirements and is prepared for effective real-world deployment.

CHAPTER 6: TESTING

6.1 Introduction

This chapter provides an overview of the testing phase for the Kidbuddy Management System. Testing ensures that the system functions as intended, meets all specified requirements, and is free of defects. The testing strategy includes white-box testing for internal code verification and black-box testing for functional verification. Additionally, testing for both system and database components is included.

6.2 Test Plan

A test plan is a document that outlines the strategy, scope, resources, and schedule for testing a system or product. It provides a structured approach to testing by detailing how testing will be performed to ensure that the system meets the required standards and functions correctly.

6.2.1 Test Organization

Test Organization refers to the structure and roles involved in the testing process. It outlines the people and their responsibilities, ensuring that all aspects of testing are managed efficiently. Table 6.1 is table test organization for Kidbuddy.

Table 6.1 : Test Organization

No.	Tester	Roles and Responsibilities
1.	Developer	Fixes defects identified during testing and supports any technical issues that arise.
2.	Reviewer/Stakeholder	Reviews test results to ensure they meet project objectives and provides feedback to refine the system.

6.2.2 Test Environment

Test Environment describes the physical and software setup where testing is conducted. It ensures that the conditions under which testing occurs are suitable for identifying defects and validating the system.

Table 6.2 : Test Environtment1

Component	Description
Hardware	-RAM (4GB) -Intel(R) Core(TM) i3-8130U
Software	-Operating System (Microsoft Window 11) -Internet Browser (Google Chrome)
Remarks	-Kidbuddy system perform well

Table 6.3: Test Environtment2

Component	Description
Hardware	-RAM(8GB) - Intel Core i3-10110U, 2.10 GHz (up to 2.59 GHz)
Software	-Operating System (Windows 10 Home (64-bit)) -Internet Browser (Microsoft Edge)
Remarks	-Kidbuddy system perform well

6.2.3 Test Schedule

Test Schedule outlines the timeline and phases of testing, ensuring that all testing activities are completed in a timely manner. It provides a roadmap for when and how testing will be carried out.

Table 6.4: Test Schedule Table

Test	Type	Duration
1. Registration	✓ Unit Test ✓ Integration Test ✓ User Acceptance Test	2 days / 4 times
2. Login System	✓ Unit Test ✓ Integration Test ✓ User Acceptance Test	5 days / 8 times
3. Booking Info	✓ Unit Test ✓ Integration Test ✓ User Acceptance Test	7 days/10 times
4. Payment Info	✓ Unit Test ✓ Integration Test ✓ User Acceptance Test	3 days/5 times
5. Training Info	✓ Unit Test ✓ Integration Test ✓ User Acceptance Test	4 days / 4 times

The registration module will undergo three types of testing: Unit Testing, Integration Testing, and User Acceptance Testing (UAT). The Unit Test will ensure that individual components of the registration process work correctly in isolation. Integration Testing will verify that the registration module interacts properly with other system components, like the database and email notifications. Finally, User Acceptance Testing will involve end-users testing the registration process to ensure it meets their needs and expectations. The entire testing process for the registration module will take 2 days, and each type of test will be conducted 4 times.

The login system is also scheduled for Unit Testing, Integration Testing, and User Acceptance Testing. The Unit Test will check the accuracy of individual functions within the login module, such as input validation and session management. Integration Testing will ensure that the login system works seamlessly with other parts of the application, such as the registration and user dashboard modules. User Acceptance Testing will involve real users verifying that the login process is smooth and meets their requirements. The testing of the login system is planned to span 5 days, with each type of test being repeated 8 times.

The booking information module will be subjected to Unit Testing, Integration Testing, and User Acceptance Testing. The Unit Test will validate those specific elements, such as date selection and booking confirmation, function as expected. Integration Testing will check the interactions between the booking module and other system components, like user profiles and payment processing. User Acceptance Testing will involve users interacting with the booking system to ensure it aligns with their expectations and workflows. Testing for the booking information module will take 7 days, with each type of test performed 10 times.

The payment information module will go through Unit Testing, Integration Testing, and User Acceptance Testing. The Unit Test will focus on individual aspects, such as payment gateway integration and error handling, to confirm they work correctly. Integration Testing will ensure that the payment module interacts effectively with other parts of the system, such as booking and user accounts. During User Acceptance Testing, users will validate that the payment process is intuitive and reliable. The overall testing duration for the payment information module is set to 3 days, with each type of test conducted 5 times.

The training information module will be tested using Unit Testing, Integration Testing, and User Acceptance Testing. The Unit Test will verify the correctness of individual training-related features, such as course enrollment and progress tracking. Integration Testing will ensure that the training module integrates smoothly with other parts of the system, like user profiles and notifications. User Acceptance Testing will involve users assessing the training module to ensure it is user-friendly and meets their needs.

Testing for the training information module is planned to last 4 days, with each type of test being performed 4 times.

6.3 Test Strategy

1. Black Box Testing Strategy

   Black-box testing focuses on evaluating the functionality of the Kidbuddy Management System without considering its internal code. Testers create scenarios based on user requirements and system specifications, ensuring that all features work correctly from an end-user perspective. This approach includes functional testing, usability testing, compatibility testing, and regression testing to verify that the system performs its intended functions accurately and remains user-friendly across different devices and browsers.

2. White Box Testing Strategy

   White-box testing involves examining the internal logic and code structure of the Kidbuddy Management System. This approach requires knowledge of the system's code to ensure that all parts of the code are executed and functioning as expected. It includes unit testing to check individual components, integration testing to verify interactions between components, and code reviews to identify and fix potential issues. The goal is to validate that the internal processes and logic work correctly and efficiently.

6.3.1 Classes of tests

- Security Testing

Security testing focuses on identifying vulnerabilities and weaknesses within the Kidbuddy Management System to protect against unauthorized access and data breaches. This test includes evaluating authentication mechanisms, data encryption, access controls, and potential threats like SQL injection or cross-site scripting (XSS). The goal is to ensure that sensitive information is secure, and that the system adheres to security best practices.

- Usability Testing

  Usability testing evaluates how easy and intuitive the Kidbuddy Management System is for its users. It focuses on the user interface, user experience, and overall design to ensure that the system is user-friendly and meets the needs of its target audience. This test involves observing users as they interact with the system to identify any difficulties or areas for improvement in navigation, accessibility, and overall satisfaction.

- Performance Testing

  Performance testing assesses the responsiveness, speed, and overall performance of the Kidbuddy Management System under normal operating conditions. This includes measuring factors like page load times, system responsiveness, and resource utilization to ensure that the system meets performance benchmarks and provides a smooth user experience.

6.4 Test Design

Test design refers to the process of creating detailed test cases and scenarios based on the requirements and specifications of a software system. The goal is to ensure that the system functions correctly and meets its intended goals. Effective test design involves planning how to verify that the system performs as expected in different conditions.

6.4.1 Test Description

Table 6.5 until Table 6.10 will describe the user description in detail according to the system modules.

6.4.1.1 User Registration Module

The Registration Module is designed to securely capture and process essential user details, including name, email, password, phone number, and NRIC, which are required for creating an account. The module includes validation mechanisms to ensure data accuracy and integrity, particularly for the NRIC and phone number. Once registered, users are seamlessly integrated into the system, enabling them to log in and access their account. Testing ensures the reliability and security of the entire registration process.

Table 6.5: Description of Registration Module

Test Case ID	Description	Action	Expected Output

KA01	Name = blank
	Email = blank
	Password = blank
	Phone = blank
	NRIC = blank
		No input provided	ERROR

KA02	Name = Adlina
	Email = blank
	Password = blank
	Phone = blank
	NRIC = blank
		Email, password, phone and NRIC left empty.	ERROR

KA03	Name = Adlina
	Email = adlinawawa
	Password = blank
	Phone = blank
	NRIC = blank
		Email do not follow format, password, phone and NRIC left empty.	ERROR

KA04	Name = Adlina
	Email = adlinawawa@gmail.com
	Password =blank
	Phone = blank
	NRIC = blank
		Password, phone and NRIC left empty.	ERROR

KA05	Name = Adlina
	Email = adlinawawa@gmail.com
	Password = ***
	Phone = blank
		Phone and NRIC left empty.	ERROR

	NRIC = blank		

KA06	Name = Adlina
	Email = adlinawawa@gmail.com
	Password = ***
	Phone = 0111237632
	NRIC = blank
		NRIC left empty.	ERROR

KA07	Name = Adlina
	Email = adlinawawa@gmail.com
	Password = ***
	Phone = 0111237632
	NRIC = 011126010212
		All necessary input inserted.	SUCCESS

6.4.1.2 Login Module

Table 6.6: Description for Login Module

Test Case ID	Description	Action	Expected Output

KB01	Email = blank
	Password = blank
		No input provided	ERROR

KB02	Email = aniesa
	Password = blank
		Email format is wrong, password left empty.	ERROR

KB03	Email = aniesa@gmail.com
	Password = blank
		Password left empty.	ERROR

KB04	Email = aniesa@gmail.com
	Password = ***
		All necessary input was inserted.	SUCCESS

6.4.1.3 Booking Module

Table 6.7: Description for Booking Module

Test Case ID	Description	Action	Expected Output

KC01	Child Name = Delaila
	Booking Date = blank
	Booking Time = blank
	Duration = 1 hour
	Location = blank
	Transportation = blank
		Booking date, booking time, location and transportation left empty.	ERROR

KC02	Child Name = Delaila
	Booking Date = 22/8/2024
	Booking Time = blank
	Duration = 1 hour
	Location = blank
	Transportation = blank
		Booking time, location and transportation left empty.	ERROR

KC03	Child Name = Delaila
	Booking Date = 22/8/2024
	Booking Time = 10.00
	Duration = 1 hour
	Location = blank
	Transportation = blank
		Location and transportation left empty.	ERROR

KC04	Child Name = Delaila
	Booking Date = 22/8/2024
	Booking Time = 10.00
	Duration = 1 hour
	Location = blank
	Transportation = blank
		Location and transportation left empty.	ERROR

KC05	Child Name = Delaila Booking Date = 22/8/2024 Booking Time = 10.00 Duration = 1 hour	Transportation left empty.	ERROR

	Location = Taman Melati, Selangor
	Transportation = blank		

KC05	Child Name = Delaila
	Booking Date = 22/8/2024
	Booking Time = 10.00
	Duration = 1 hour
	Location = Taman Melati, Selangor
	Transportation = Self Deliver
		All necessary input was inserted	SUCCESS

6.4.2 Test Data

Table 6.8: Test Data for User to Login

Column Name	**TD_01-1**	**TD_01-2**	**TD_01-3**
Test Case ID	**TD_01-1**	**TD_01-2**	**TD_01-3**
Login	The user entered an incorrect email address. **Email:** adlina@gmail **Password:** abc123	The user did not fill in the required fields. **Email:** **Password:**	The user filled in the fields correctly. **Email:** adlina@gmail.com **Password:** abc123
Result test data	Login failed. “Invalid email or password. Please try again.”	Login failed. “Please fill out this field.”	User login successfully

Table 6.9: Test Data for Submit Booking

Column Name	**TD_02-1**	**TD_02-2**	**TD_02-3**
Test Case ID	**TD_02-1**	**TD_02-2**	**TD_02-3**
Booking Info	The user did not fill in all the required fields. Child Name = Delaila Booking Date = 22/8/2024 Booking Time = Duration = 1 hour Location = Transportation =	The user fills in the booking date which is on Sunday Child Name = Delaila Booking Date = 1/9/2024 Booking Time = 10.00 Duration = 1 hour Location = Taman Melati, Selangor Transportation = Self Deliver	The user fills in the field detail correctly. Child Name = Delaila Booking Date = 22/8/2024 Booking Time = 10.00 Duration = 1 hour Location = Taman Melati, Selangor Transportation = Self Deliver
Result Test Data	Booking failed. “Please fill out all field.”	Booking failed. “Booking is not allowed on Sunday. Please select another day”	Booking submits successfully

Table 6.10: Test Data for User Registration

Column Name	**TD_03-1**	**TD_03-2**	**TD_03-3**
Test Case ID	**TD_03-1**	**TD_03-2**	**TD_03-3**

Registration	The user entered an email that already registered.
	Name = Adlina
	Email = adlina@gmail.com
	Password = ***
	Phone = 0111237632
	NRIC = 011126010212
		The user did not fill in the required fields.
		Name = Adlina
		Email = adlinawawa@gmail.com
		Password = ***
		Phone = 0111237632
		NRIC =
			The user filled in the fields correctly.
			Name = Adlina
			Email = adlinawawa@gmail.com
			Password = ***
			Phone = 0111237632
			NRIC = 011126010212

Result test data	Registration failed. “That email already registered”	Login failed. “Please fill out this field.”	User successfully registered.

6.5 Test Results and Analysis

In the test results and analysis phase, both white box and black box testing techniques were utilized to thoroughly evaluate the system's functionality, reliability, and performance. White box testing involved an in-depth examination of the internal logic and code structure. This testing technique included code reviews, branch testing, and loop testing to ensure that all components functioned as intended. The detailed code review helped identify and correct potential issues such as logic errors, security vulnerabilities, and performance bottlenecks. As a result, the white box testing confirmed that the system operates correctly at the code level, with no detected defects that could impact its performance or security.

On the other hand, black box testing focused on validating the system's functionality from the user's perspective without delving into the internal code structure. This technique was implemented through a Google Forms survey, distributed to 20 respondents who provided valuable feedback on the system's usability and overall experience. The black box testing allowed us to gather insights directly from users, helping to identify areas for improvement and ensure that the system meets user needs in real-world scenarios. The results from the black box testing indicated that the system performed well overall, with generally positive feedback from users, although a few areas were identified for further enhancement.

Table 6.11 : Test Results and Analysis of Login Module

Module/Component	**Result**
Test Case ID	**Test Data ID**
TC_01-1	TD_01-1
TC_01-2	TD_01-2
TC_01-3	TD_01-3

- **TC_01-1**: The user entered an incorrect email address (adlina@gmail) and password (abc123). The login attempt failed as expected, with the system returning the message: “Invalid email or password. Please try again.”
- **TC_01-2**: The user did not fill in the required fields for email and password. The login attempt failed with the system prompting the message: “Please fill out this field.”
- **TC_01-3**: The user entered the correct email (adlina@gmail.com) and password (abc123). The login was successful, and the user was granted access to the system.

Table 6.12: Test Results and Analysis of Booking Info Module

Module/Component	**Result**
Test Case ID	**Test Data ID**
TC_02-1	TD_02-1
TC_02-2	TD_02-2
TC_02-3	TD_02-3

- **TC_02-1**: The user did not fill in all required fields (Booking Time, Location). The booking attempt failed with the message: "Please fill out all fields."
- **TC_02-2**: The user attempted to make a booking on a Sunday. The system prevented the booking with the message: "Booking is not allowed on Sunday. Please select another day."
- **TC_02-3**: The user filled in all fields correctly with valid details. The booking was successfully submitted, and the system recorded the booking.

Table 6.13: Test Results and Analysis of User Registration Module

Module/Component	**Result**
Test Case ID	**Test Data ID**
TC_03-1	TD_03-1
TC_03-2	TD_03-2

- **TC_03-1**: The user attempted to register with an email address (adlina@gmail.com) that is already in the system. The registration failed, and the system returned the message: "That email already registered."
- **TC_03-2**: The user did not fill in all required fields (NRIC). The registration failed, with the system prompting the message: "Please fill out this field."
- **TC_03-3**: The user provided all required details correctly. The registration was successful, and the user's information was stored in the system.

6.6 User Acceptance Testing (UAT)

The User Acceptance Testing (UAT) aimed to ensure that the system meets the business requirements and user expectations before deployment. As part of the UAT, black box testing played a significant role. The system's functionality was evaluated through the lens of the end-user, without considering the internal workings of the system. A Google Forms survey was distributed to 30 users, primarily parents and caregivers which is the target audience of the system. These respondents provided feedback on various aspects such as navigation, ease of use, and effectiveness. The feedback was instrumental in validating the system's readiness for deployment, highlighting some areas for improvement.

Based on survey feedback from Figure 12-31 in an attachment, the survey responses for the Kidbuddy Management System indicate a highly positive user experience overall. Most respondents found the system easy to navigate, with the majority rating it a 5, suggesting that the platform is intuitive and user-friendly. The visual design of the system was also well-received, with high marks for appearance, although a few users suggested minor improvements, such as increasing transparency in certain background elements to enhance readability.

Buttons and links within the system were generally deemed clear and easy to understand, reflecting effective labeling and layout. While the text size and font style were mostly considered readable, some respondents recommended slightly larger text for better accessibility, particularly for older users. The color scheme was widely appreciated, though a few users suggested improving color contrast for enhanced visibility.

The logical flow of the system was another highlight, with users finding it straightforward to move from one task to another. Task completion, including booking and profile updates, was reported as easy, though some users felt that simplifying the booking process could make it even more efficient. The steps required for navigation were also rated as clear and well-organized.

In terms of responsiveness, the system performed well, with most users reporting quick reactions to their actions, such as clicking buttons and loading pages. Only a few noted minor issues like occasional slow loading times. Regarding potentially confusing areas, most parents and caregivers did not encounter significant difficulties, although some mentioned that the payment process after booking and certain caregiver functions could be made clearer.

Users felt confident in understanding the system's features and generally did not require additional guidance, though a few suggested that a more detailed user manual could benefit new users. Finding help or support within the system was relatively easy for most, but there were mentions of potential enhancements, like adding a chatbot for real-time assistance.

Overall, respondents were confident in using the system independently and found it effective in helping them manage childcare tasks. The additional comments were largely positive, with users praising the system's design, functionality, and ease of use, while offering constructive feedback on areas for further improvement.

6.7 Conclusion

In conclusion, the testing phase for the Kidbuddy Management System has been thoroughly conducted to ensure that all components of the system function as intended. By utilizing both black-box and white-box testing strategies, we were able to verify the system's functionality from both an end-user perspective and an internal code structure perspective. The combination of unit tests, integration tests, and user acceptance tests across different modules, including Registration, Login, Booking, Payment, and Training, has provided a comprehensive validation of the system.

The system has met the specified requirements and passed all critical test cases, confirming its readiness for deployment. The testing process also identified and resolved potential issues, ensuring that the system is robust, secure, and user-friendly. By adhering to a structured test plan and schedule, the testing team has successfully validated that the Kidbuddy Management System is fully functional, reliable, and ready for use by its intended audience.

CHAPTER 7: CONCLUSION

7.1 Introduction

This chapter provides a comprehensive conclusion to the development and analysis of the Kidbuddy management system. It synthesizes the findings from the project, evaluates the system's strengths and weaknesses, proposes areas for future improvement, and outlines the key contributions made by this project. By reflecting on the overall development process and assessing the system's impact, this chapter aims to offer insights into the system's effectiveness and potential for further enhancement. The objective is to provide a holistic understanding of the system's current state while suggesting pathways for its evolution.

7.2 Observation on Weakness and Strengths

1. Strengths

The Kidbuddy management system demonstrates significant strengths, particularly in its comprehensive functionality, user-friendly interface, automation of key processes, and scalability.

First and foremost, the system successfully integrates a wide range of essential features, including booking management, payment processing, activity tracking, and training management. This comprehensive approach directly addresses many of the challenges commonly faced in the childcare sector, such as inefficiencies in communication, scheduling conflicts, and the need for streamlined operations. By providing a one-stop solution, the system enhances the overall experience for parents, caregivers, and administrators alike.

The user interface has been meticulously designed to prioritize simplicity and intuitiveness. By focusing on ease of navigation, the system ensures that users, regardless of their technical proficiency, can efficiently interact with its various components. This user-centric design not only improves the adoption rate but also enhances the satisfaction of users, who find the system accessible and straightforward.

Automation plays a pivotal role in reducing the manual effort required by users. The automation of booking, payment, and notification processes minimizes the likelihood of human error, thereby improving efficiency. In the childcare sector, where timely communication and precise scheduling are critical, this automation significantly contributes to smoother operations and better service delivery.

Scalability is another key strength of the system. It has been designed with future growth in mind, making it adaptable to an expanding user base and potential geographical expansion. This ensures that the system can evolve alongside the needs of its users, maintaining its relevance and utility over time.

2. Weakness

Despite its strengths, the system also exhibits certain weaknesses that warrant attention. One notable weakness is the complexity of the initial setup. For users without a technical background, deploying the system may pose challenges, even though the system itself is user-friendly once operational. This complexity could be a barrier to entry for some users, necessitating the availability of technical support during the setup phase.

Another limitation is the system's current lack of extensive customization options. While the system covers most essential features, it offers limited flexibility for users who require specific functionalities or tailored interfaces. This could be a drawback for larger organizations or those with unique operational needs, as they may find the system less adaptable to their specific requirements.

A significant operational weakness is the system's dependency on internet connectivity. As a web-based platform, the system relies heavily on a stable internet connection. In regions with poor or unreliable connectivity, users may experience disruptions in service, leading to frustration and reduced efficiency.

This dependency on internet connectivity highlights the need for an offline mode to mitigate potential connectivity issues.

Additionally, one of the system's primary weaknesses lies in the booking process. There is a possibility that users may not secure a caregiver after making a booking, as it depends on whether the caregiver decides to accept the job. This uncertainty can be frustrating for users, particularly in situations where immediate childcare is required. The system's reliance on caregiver availability introduces a level of unpredictability that could undermine user confidence in the booking process.

7.3 Propositions for Improvement

To address the identified weaknesses and further enhance the system's capabilities, several areas for improvement have been proposed.

One key area is the enhancement of customization options. To better cater to the diverse needs of users, future iterations of the KidBuddy management system could include more flexible configuration settings. This could involve customizable dashboards, reporting features, and the ability to add or remove modules based on user preferences. By offering greater customization, the system would be able to accommodate a wider range of user requirements, making it more appealing to larger organizations and those with unique operational needs.

Another significant enhancement would be the development of offline functionality. Recognizing the challenges posed by unreliable internet connectivity, the introduction of an offline mode could greatly improve the system's usability in areas with poor internet access. This mode would allow users to access critical features such as booking management and activity tracking without an active internet connection, with data syncing automatically when connectivity is restored. Such a feature would ensure that the system remains functional and reliable, even in challenging environments.

The development of a dedicated mobile application is also proposed to enhance user experience. While the system is currently accessible via web browsers, a mobile app could offer a more streamlined and convenient interface, particularly for caregivers and parents who rely on mobile devices for quick access to services. The app could include features such as push notifications for real-time updates, an optimized interface for mobile use, and offline capabilities, further improving the system's accessibility and usability.

Integration with third-party services could also significantly enhance the system's functionality. By integrating with external services such as calendar apps, payment gateways, and messaging platforms, the system could offer users a more seamless experience. For example, users could sync appointments with their personal calendars, utilize a wider range of payment options, and communicate more effectively through integrated messaging tools. This integration would add convenience and efficiency, making the system more appealing to a broader audience.

7.4 Project Contribution

This project has made several valuable contributions, both to the academic community and the childcare industry.

From an academic perspective, the project demonstrates the practical application of database and software development methodologies in addressing real-world challenges. It serves as a case study and reference for future projects focused on childcare management systems, offering insights into the development process, challenges encountered, and solutions implemented. The project can also be used as a teaching tool in relevant courses, providing students with a tangible example of how theoretical concepts can be applied in practice.

The context of the childcare industry, the system offers a modern solution to many of the operational challenges faced by childcare providers. By providing tools that improve operational efficiency, communication, and service quality, the KidBuddy management system has the potential to elevate industry standards and set a new benchmark for childcare management systems. Its comprehensive feature set, user-friendly interface, and scalability make it a valuable tool for childcare providers seeking to enhance their operations and service delivery.

The user manual, which is detailed in Appendix XX, serves as a comprehensive guide for new users and administrators. It provides clear instructions on system setup, usage, and troubleshooting, ensuring that users can quickly and effectively adopt the system. This manual is an essential resource for ensuring a smooth transition to the new system and maximizing its benefits.

7.5 Conclusion

The development of the KidBuddy management system represents a significant advancement in addressing the operational challenges faced by parents, caregivers, and childcare administrators. Through the automation of key processes and the enhancement of communication, the system not only improves operational efficiency but also contributes to a better overall user experience.

While the system has demonstrated considerable strengths in terms of comprehensiveness, usability, and scalability, it is not without its limitations. The complexity of the initial setup, limited customization options, dependency on internet connectivity, and the uncertainty in securing a caregiver after booking are notable areas that require attention. These weaknesses, however, present opportunities for further refinement and enhancement.

The proposed improvements, including enhanced customization options, offline functionality, mobile app development, and integration with third-party services, offer a clear roadmap for future iterations of the system. By addressing these areas, the system has the potential to evolve into an even more robust and versatile tool, capable of meeting the diverse needs of its users.

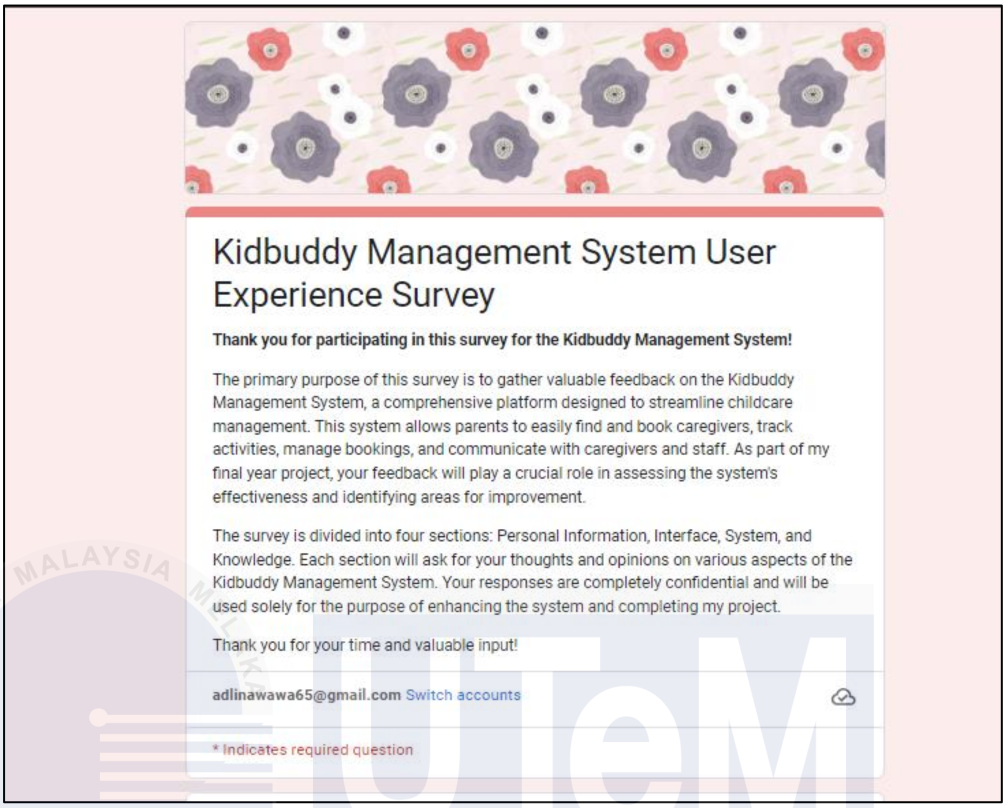
In conclusion, the KidBuddy management system has made significant contributions to the childcare industry, offering a modern and efficient solution to longstanding challenges. With ongoing enhancements and improvements, the system is well-positioned to become a leading solution in the childcare management sector, providing tangible benefits to all stakeholders involved.

REFERENCES

Chin, T., Bing, C., Dhamotharan, M., & Che Mustafa, M. (2021). Issues and challenges in early childhood care and education centre registration process: What the operators say?. Southeast Asia Early Childhood Journal, 10, 53-62. https://doi.org/10.37134/saecj.vol10.sp.5.2021

Herman, E.R., Breedlove, M.L. & Lang, S.N. Family Child Care Support and Implementation: Current Challenges and Strategies from the Perspectives of Providers. Child Youth Care Forum 50, 1037–1062 (2021). https://doi.org/10.1007/s10566-021-09613-5

Xu, T., Akerson, A. & Fox, B. Incorporating technology to foster effective communication and documentation in child care programs. *Educ Inf Technol* **26**, 5685–5695 (2021). https://doi.org/10.1007/s10639-021-10561-9



Kidbuddy Management System User Experience Survey

Thank you for participating in this survey for the Kidbuddy Management System!

The primary purpose of this survey is to gather valuable feedback on the Kidbuddy Management System, a comprehensive platform designed to streamline childcare management. This system allows parents to easily find and book caregivers, track activities, manage bookings, and communicate with caregivers and staff. As part of my final year project, your feedback will play a crucial role in assessing the system's effectiveness and identifying areas for improvement.

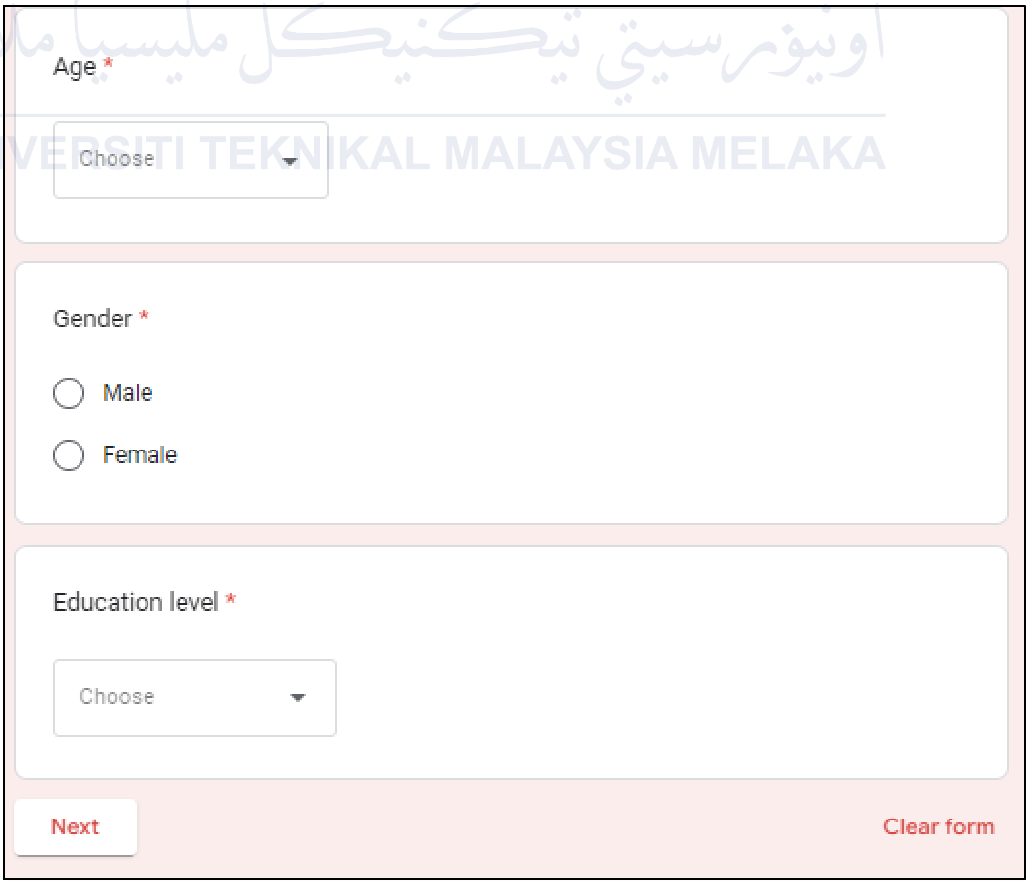
The survey is divided into four sections: Personal Information, Interface, System, and Knowledge. Each section will ask for your thoughts and opinions on various aspects of the Kidbuddy Management System. Your responses are completely confidential and will be used solely for the purpose of enhancing the system and completing my project.

Thank you for your time and valuable input!

adlinawawa65@gmail.com Switch accounts

* Indicates required question

Figure 2 : Permission letter to do the survey



Age *

Choose

Gender *

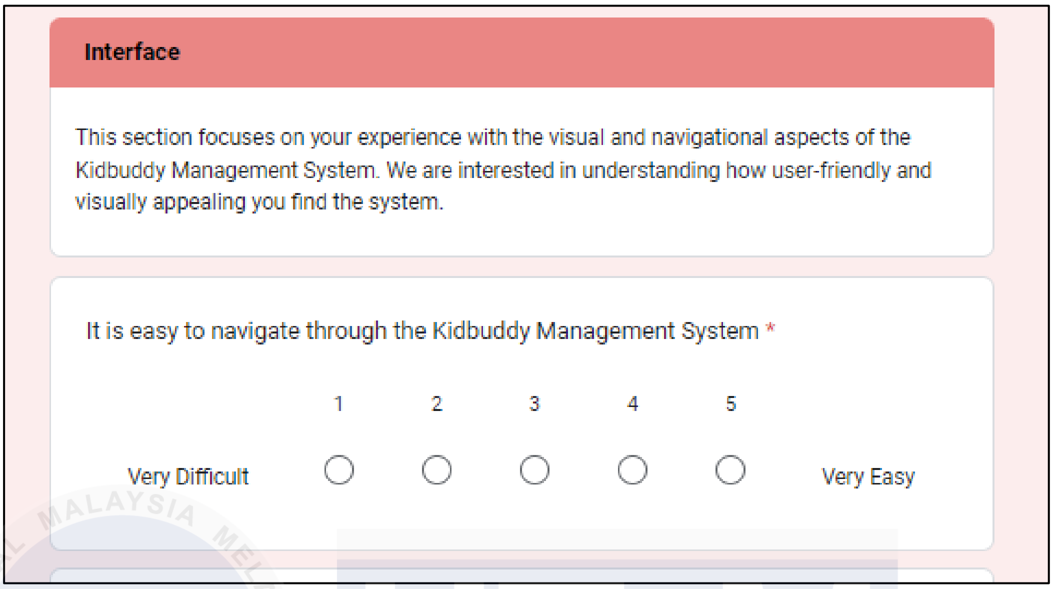
○ Male

○ Female

Education level *

Choose

Figure 3 : Survey form for testing purpose (b)



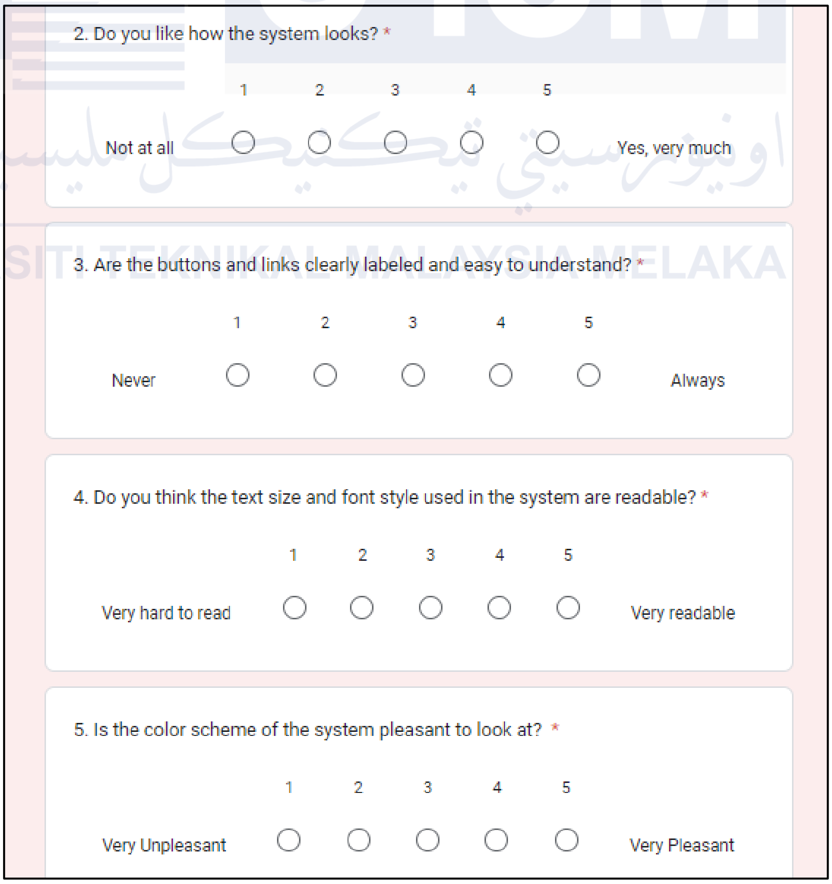
Interface

This section focuses on your experience with the visual and navigational aspects of the Kidbuddy Management System. We are interested in understanding how user-friendly and visually appealing you find the system.

It is easy to navigate through the Kidbuddy Management System *

	1	2	3	4	5	
Very Difficult	○	○	○	○	○	Very Easy

Figure 4 : Survey form for testing purpose (c)



2. Do you like how the system looks? *

	1	2	3	4	5	
Not at all	○	○	○	○	○	Yes, very much

3. Are the buttons and links clearly labeled and easy to understand? *

	1	2	3	4	5	
Never	○	○	○	○	○	Always

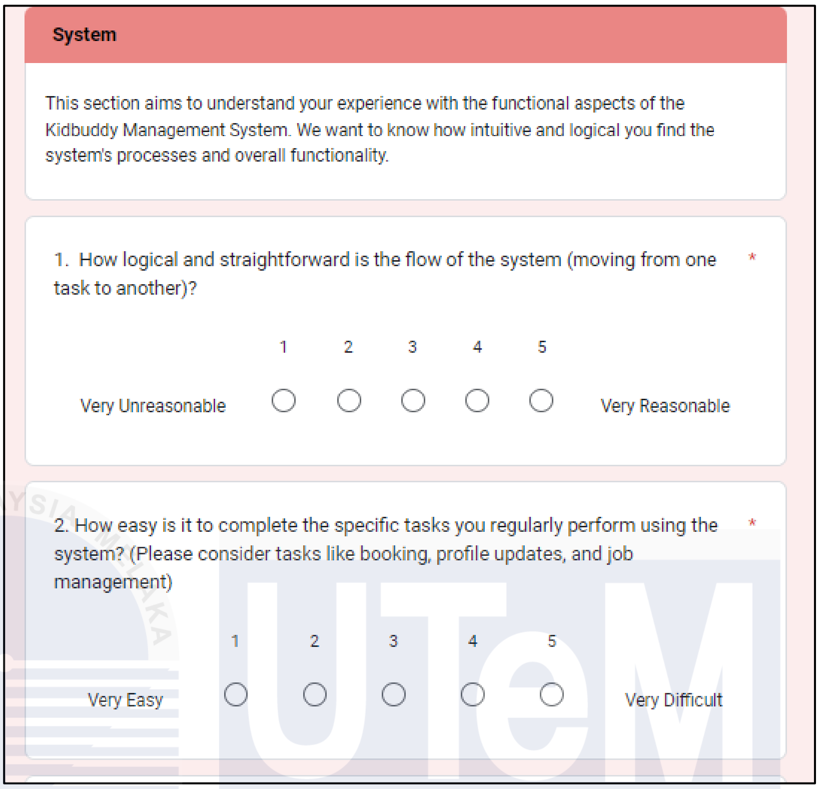
4. Do you think the text size and font style used in the system are readable? *

	1	2	3	4	5	
Very hard to read	○	○	○	○	○	Very readable

5. Is the color scheme of the system pleasant to look at? *

	1	2	3	4	5	
Very Unpleasant	○	○	○	○	○	Very Pleasant

Figure 5 : Survey form for testing purpose (d)



System

This section aims to understand your experience with the functional aspects of the Kidbuddy Management System. We want to know how intuitive and logical you find the system's processes and overall functionality.

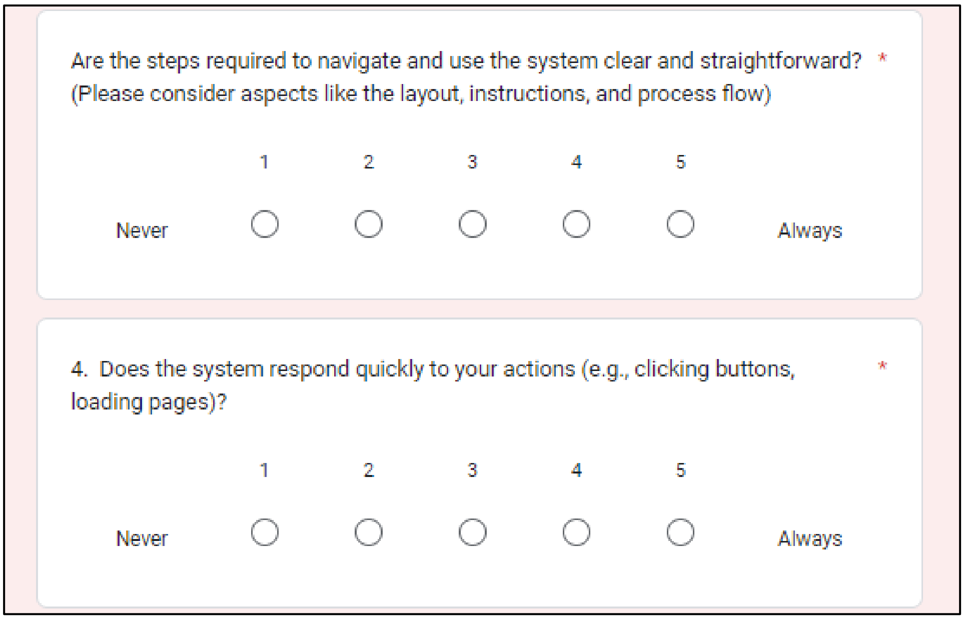
1. How logical and straightforward is the flow of the system (moving from one task to another)? *

	1	2	3	4	5	
Very Unreasonable	○	○	○	○	○	Very Reasonable

2. How easy is it to complete the specific tasks you regularly perform using the system? (Please consider tasks like booking, profile updates, and job management) *

	1	2	3	4	5	
Very Easy	○	○	○	○	○	Very Difficult

Figure 6: Survey form for testing purpose (e)



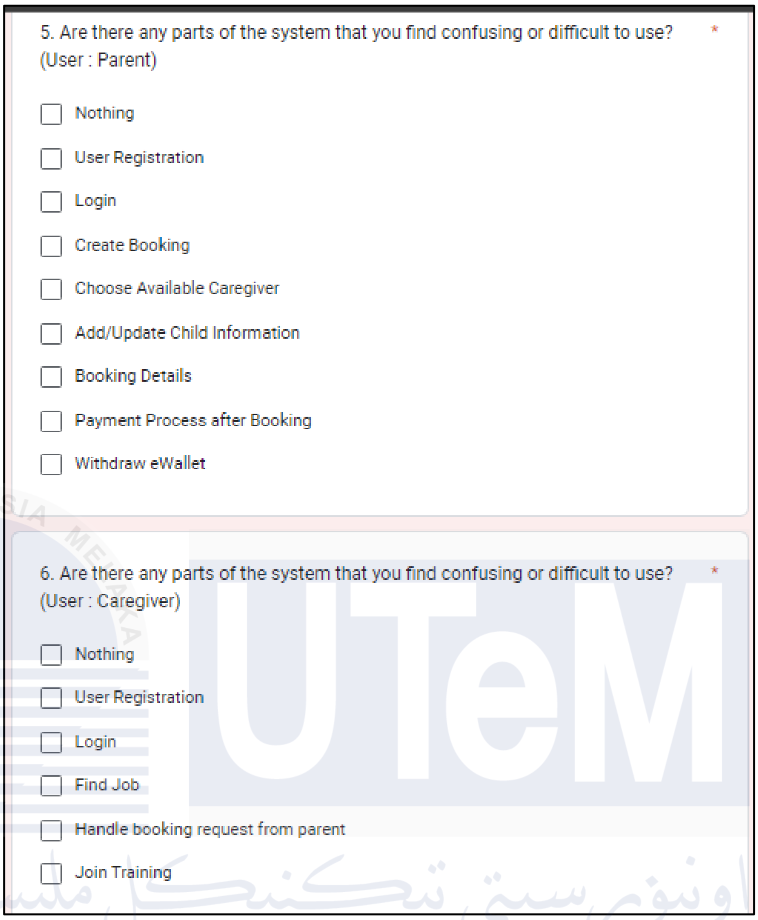
Are the steps required to navigate and use the system clear and straightforward? *
(Please consider aspects like the layout, instructions, and process flow)

	1	2	3	4	5	
Never	○	○	○	○	○	Always

4. Does the system respond quickly to your actions (e.g., clicking buttons, loading pages)? *

	1	2	3	4	5	
Never	○	○	○	○	○	Always

Figure 7: Survey form for testing purpose (f)



5. Are there any parts of the system that you find confusing or difficult to use? (User : Parent) *

□ Nothing

□ User Registration

□ Login

□ Create Booking

□ Choose Available Caregiver

□ Add/Update Child Information

□ Booking Details

□ Payment Process after Booking

□ Withdraw eWallet

6. Are there any parts of the system that you find confusing or difficult to use? (User : Caregiver) *

□ Nothing

□ User Registration

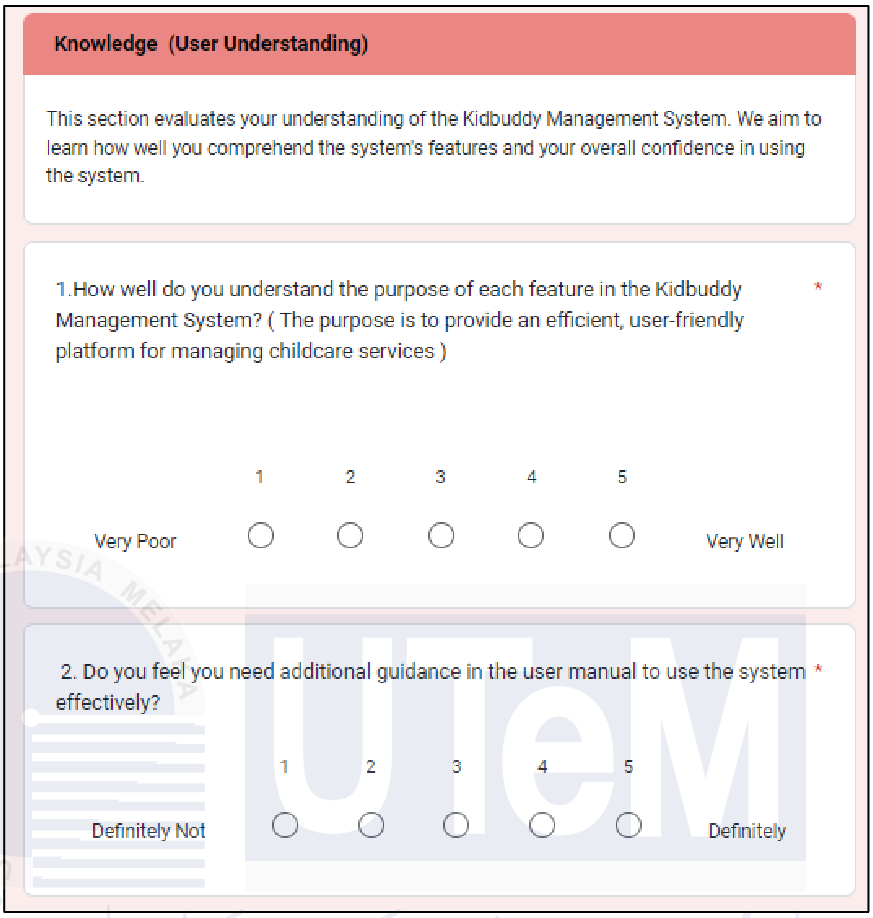
□ Login

□ Find Job

□ Handle booking request from parent

□ Join Training

Figure 8: Survey form for testing purpose (g)



Knowledge (User Understanding)

This section evaluates your understanding of the Kidbuddy Management System. We aim to learn how well you comprehend the system's features and your overall confidence in using the system.

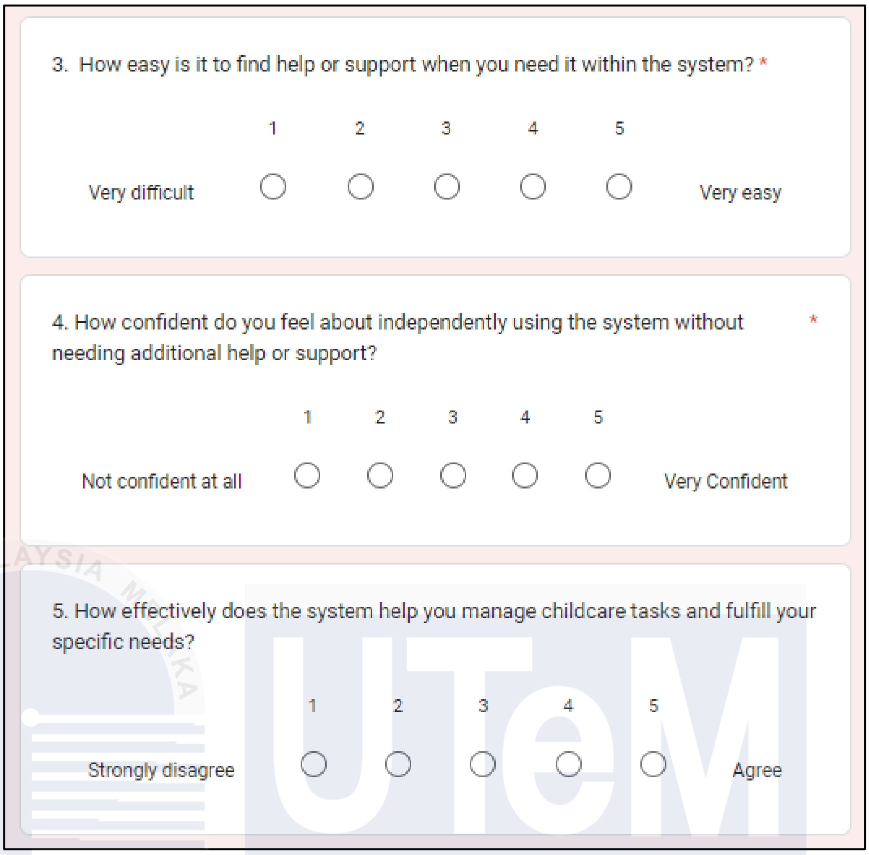
1.How well do you understand the purpose of each feature in the Kidbuddy Management System? (The purpose is to provide an efficient, user-friendly platform for managing childcare services) *

	1	2	3	4	5	
Very Poor	○	○	○	○	○	Very Well

2. Do you feel you need additional guidance in the user manual to use the system effectively? *

	1	2	3	4	5	
Definitely Not	○	○	○	○	○	Definitely

Figure 9: Survey form for testing purpose (h)



3. How easy is it to find help or support when you need it within the system? *

	1	2	3	4	5	
Very difficult	○	○	○	○	○	Very easy

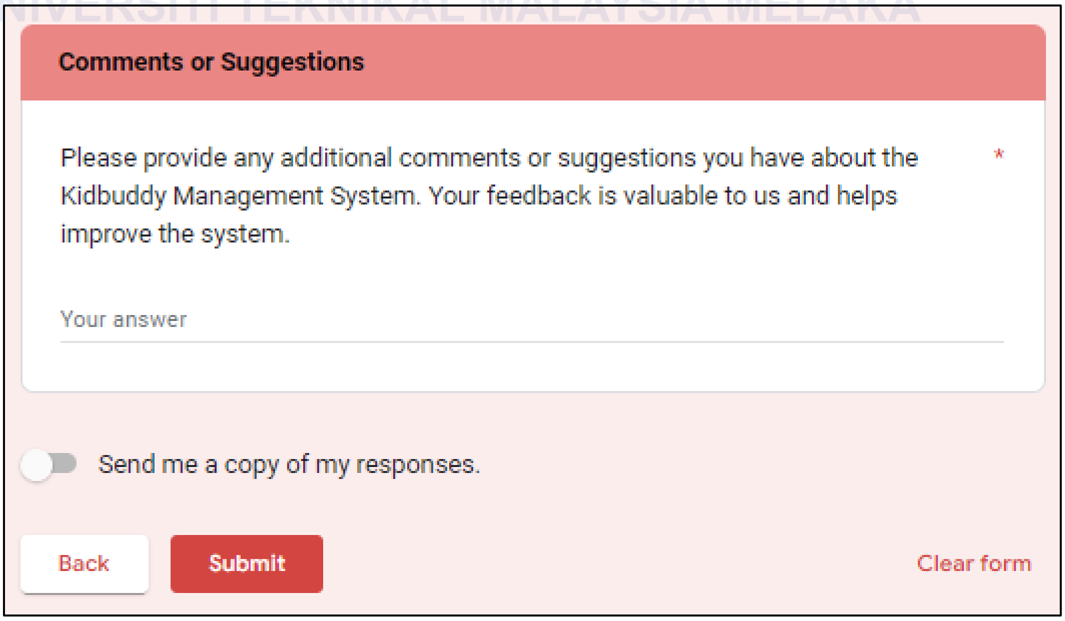
4. How confident do you feel about independently using the system without needing additional help or support? *

	1	2	3	4	5	
Not confident at all	○	○	○	○	○	Very Confident

5. How effectively does the system help you manage childcare tasks and fulfill your specific needs?

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Agree

Figure 10: Survey form for testing purpose (i)



Comments or Suggestions

Please provide any additional comments or suggestions you have about the Kidbuddy Management System. Your feedback is valuable to us and helps improve the system. *

Your answer

Send me a copy of my responses.

Figure 11: Survey form for testing purpose (j)

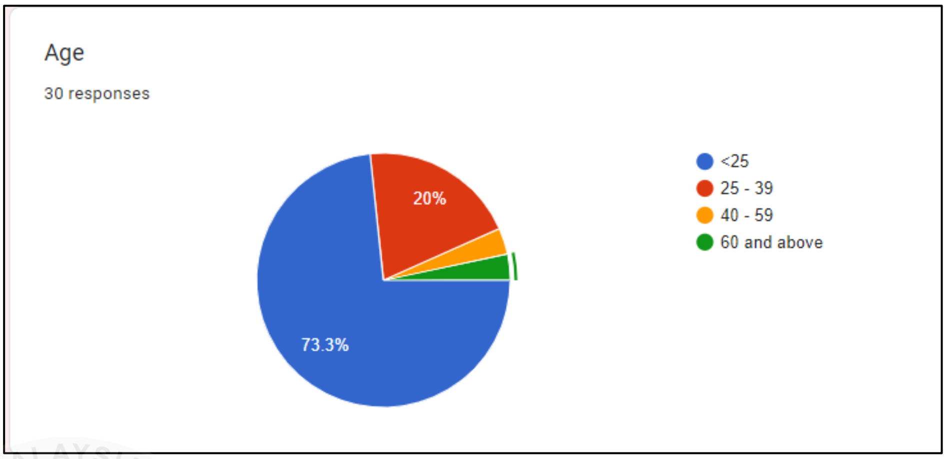


Figure 12: Age Distribution of Survey Respondents: Majority (73.3%) are Under 25 Years Old

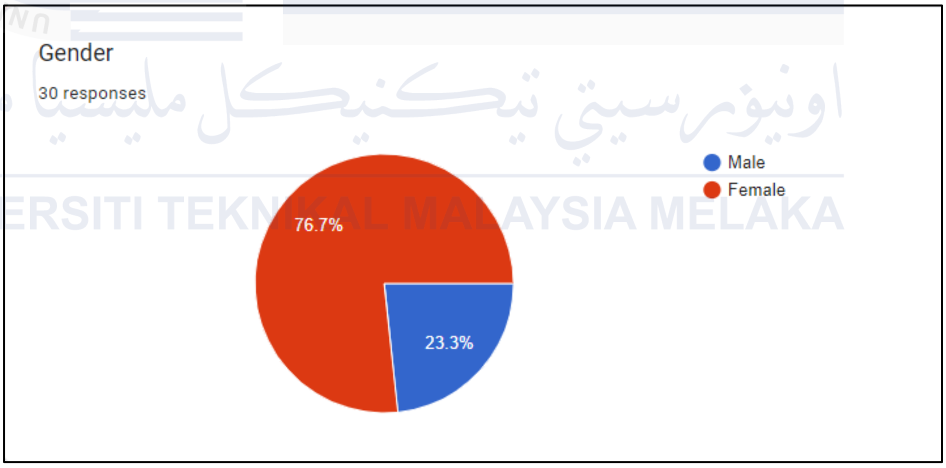


Figure 13: Gender Distribution of Survey Respondents: Majority (76.7%) are Female

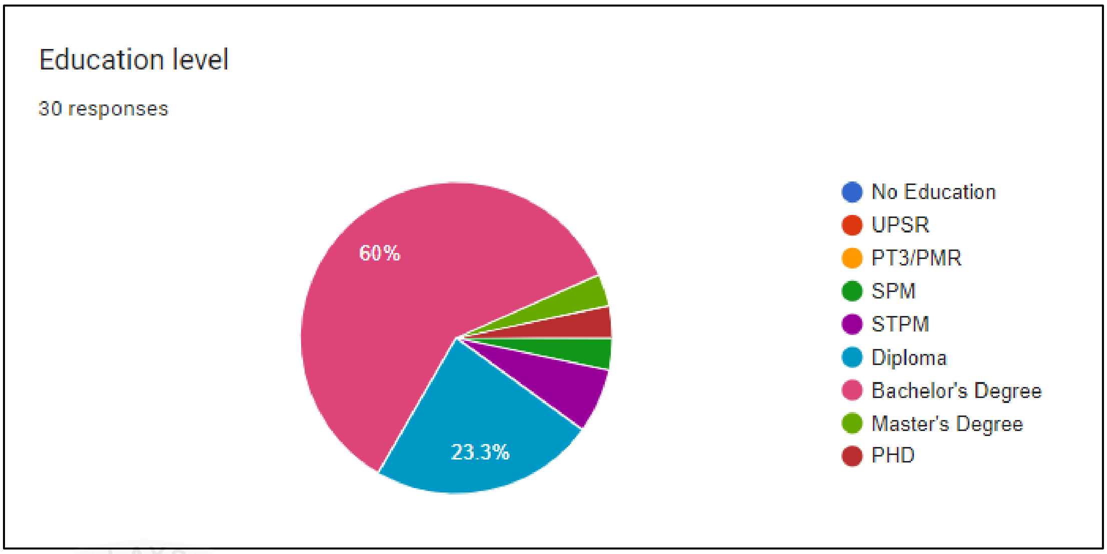


Figure 14: Education Level Distribution of Survey Respondents: Majority (60%) are Bachelor's Degree

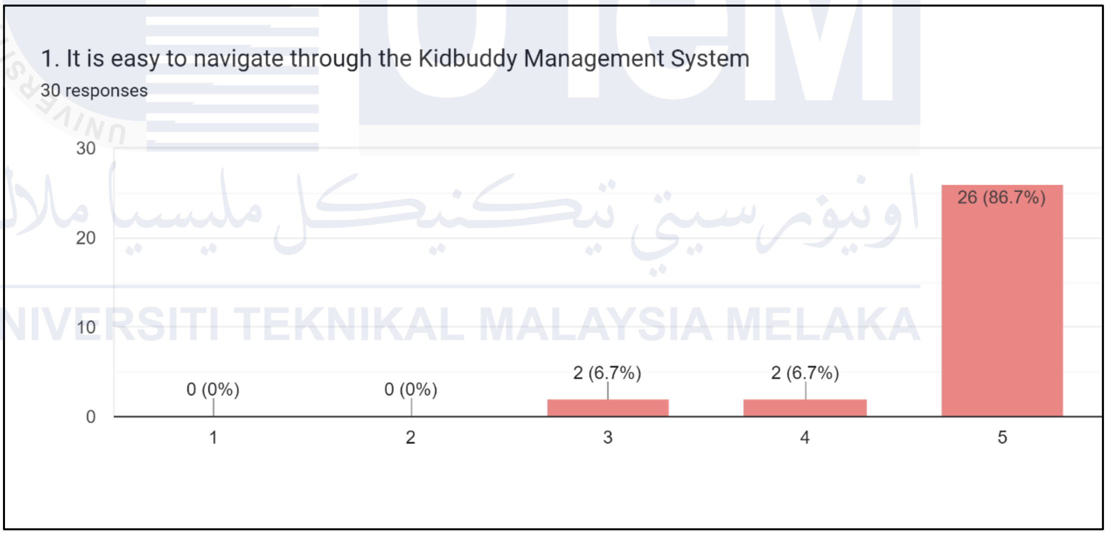
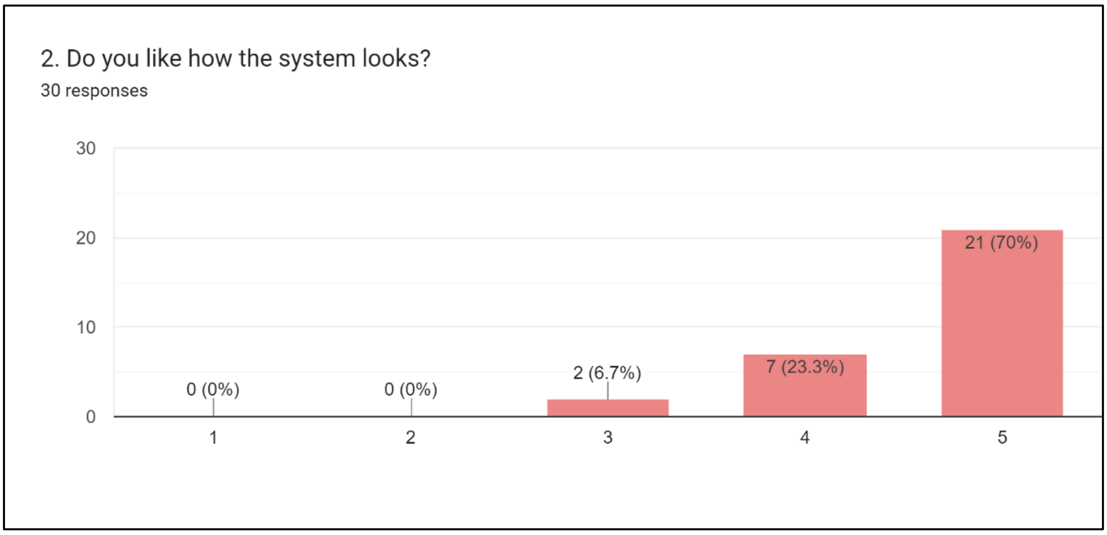
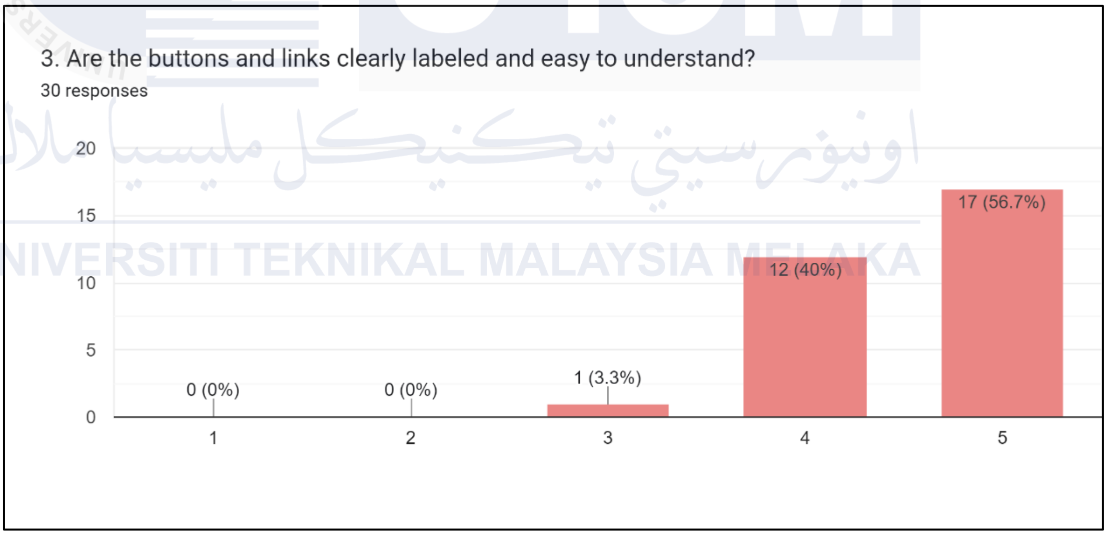


Figure 15: Question 1 Interface Distribution of Survey Respondents: Majority rate 5


2. Do you like how the system looks?

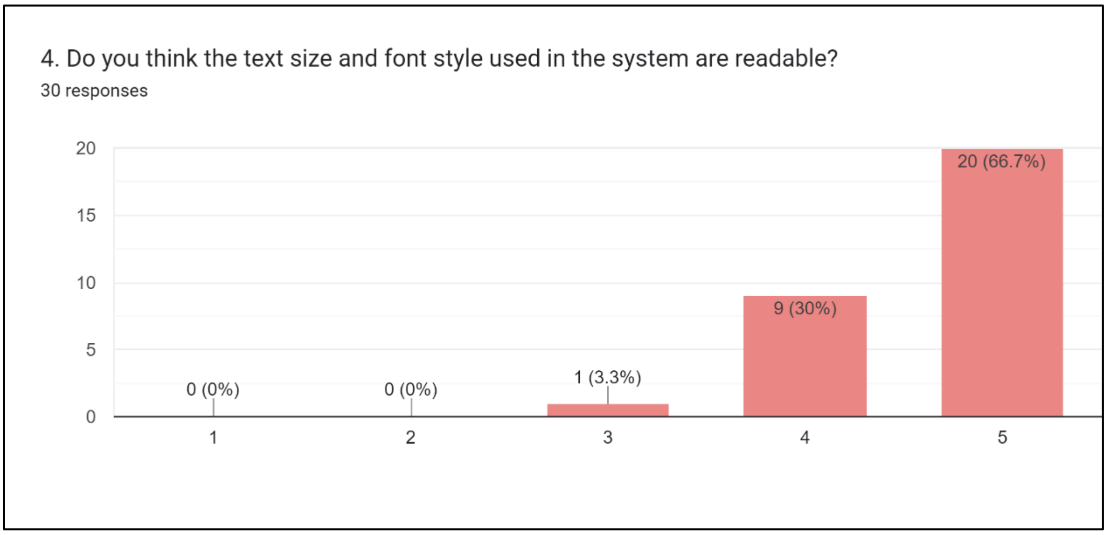
30 responses

Figure 16: Question 2 Interface Distribution of Survey Respondents: Majority rate 5


3. Are the buttons and links clearly labeled and easy to understand?

30 responses

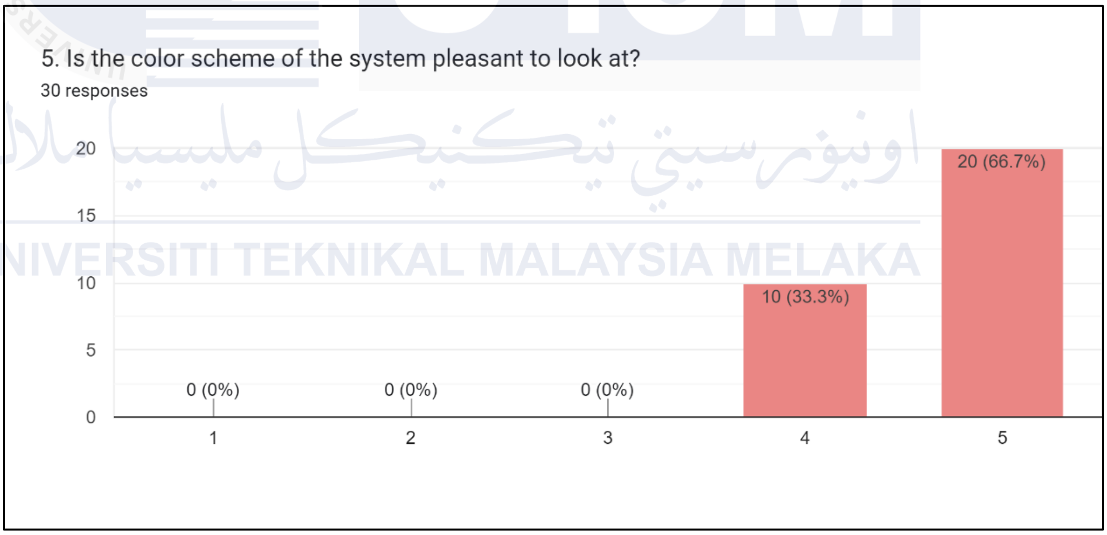
Figure 17: Question 3 Interface Distribution of Survey Respondents: Majority rate 5



4. Do you think the text size and font style used in the system are readable?

30 responses

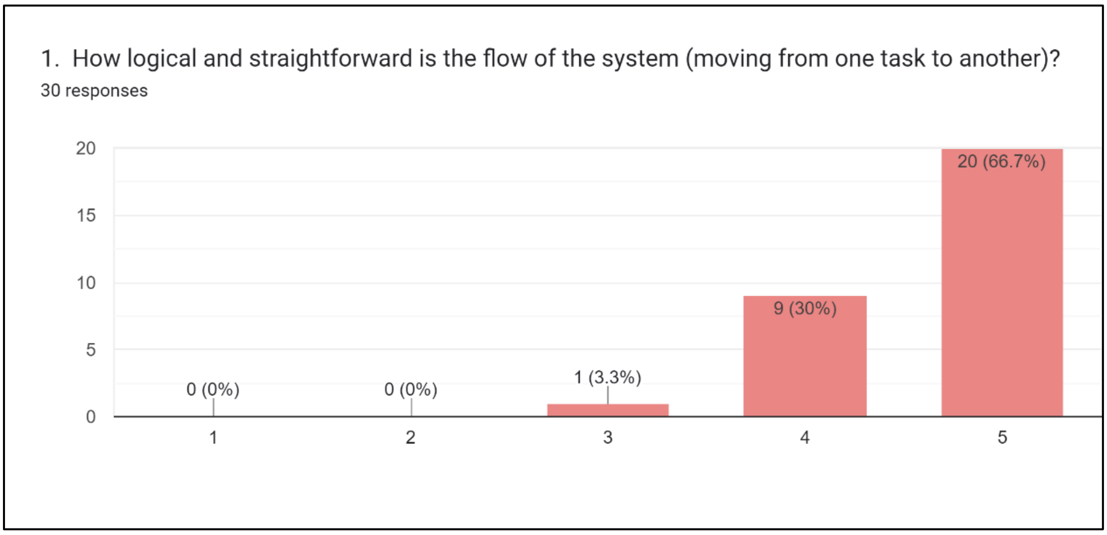
Figure 18: Question 4 Interface Distribution of Survey Respondents: Majority rate 5



5. Is the color scheme of the system pleasant to look at?

30 responses

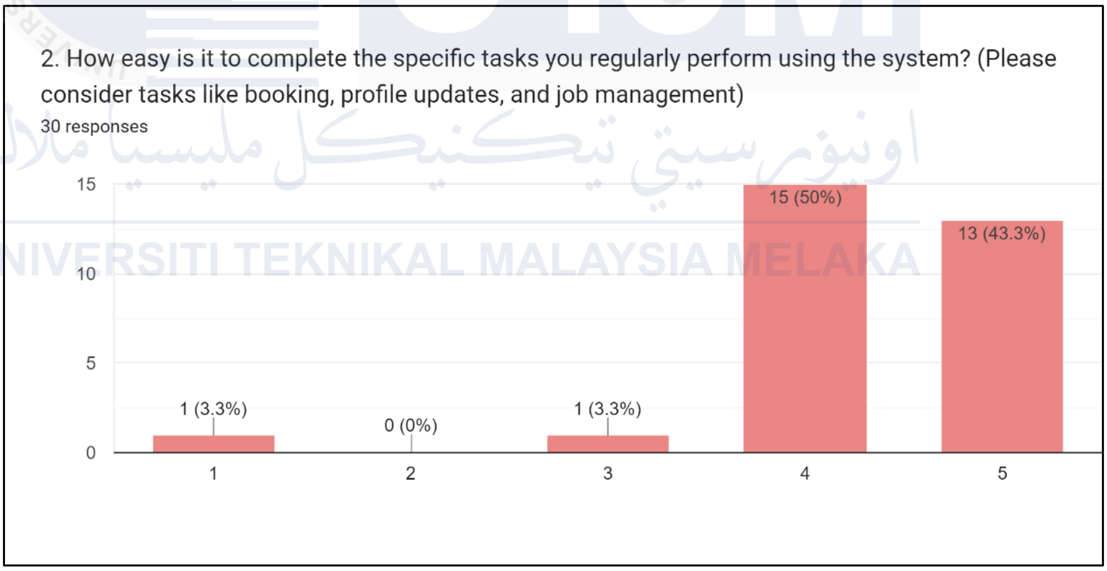
Figure 19: Question 5 Interface Distribution of Survey Respondents: Majority rate 5



1. How logical and straightforward is the flow of the system (moving from one task to another)?

30 responses

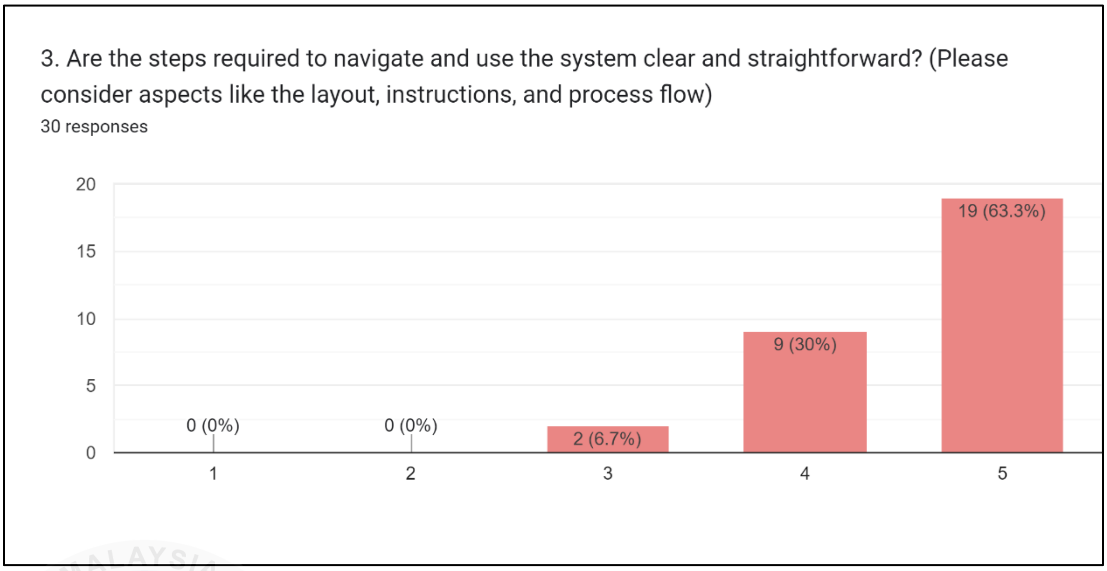
Figure 20: Question 1 System Distribution of Survey Respondents: Majority rate 5



2. How easy is it to complete the specific tasks you regularly perform using the system? (Please consider tasks like booking, profile updates, and job management)

30 responses

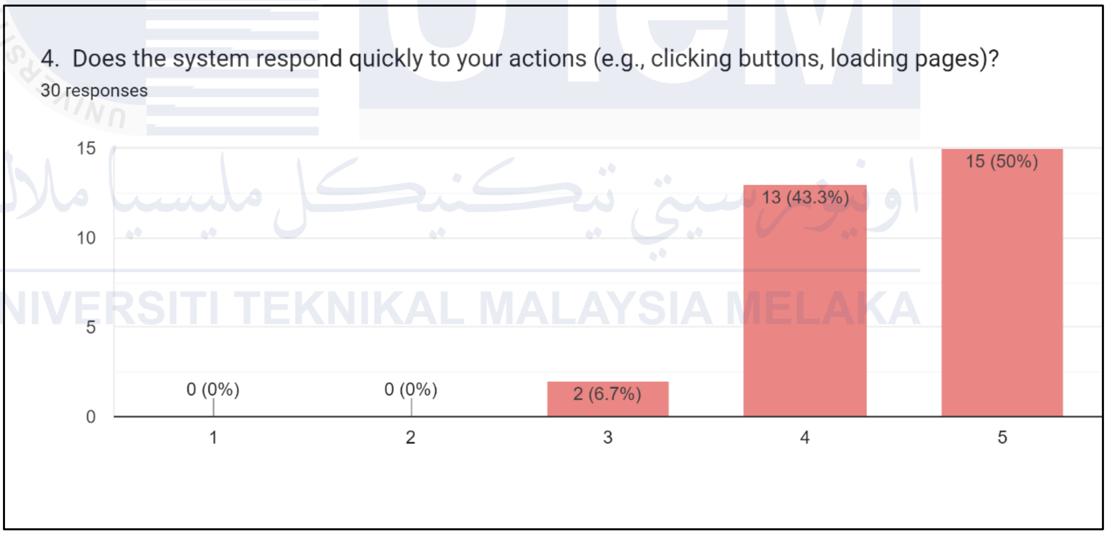
Figure 21: Question 2 System Distribution of Survey Respondents: Majority rate 5



3. Are the steps required to navigate and use the system clear and straightforward? (Please consider aspects like the layout, instructions, and process flow)

30 responses

Figure 22: Question 3 System Distribution of Survey Respondents: Majority rate 5



4. Does the system respond quickly to your actions (e.g., clicking buttons, loading pages)?

30 responses

Figure 23: Question 4 System Distribution of Survey Respondents: Majority rate 5

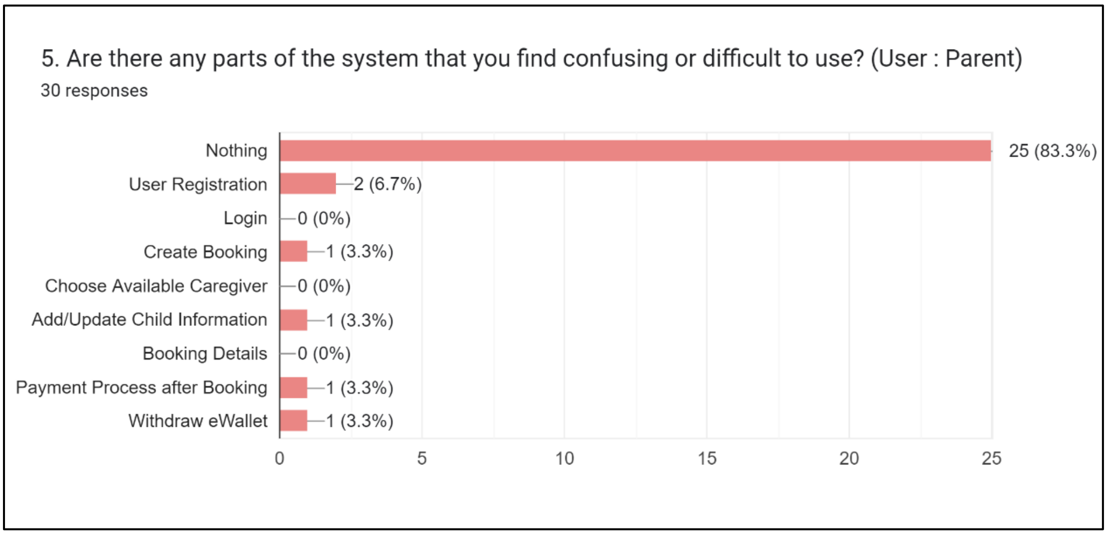


Figure 24: Question 5 System Distribution of Survey Respondents: Majority rate nothing

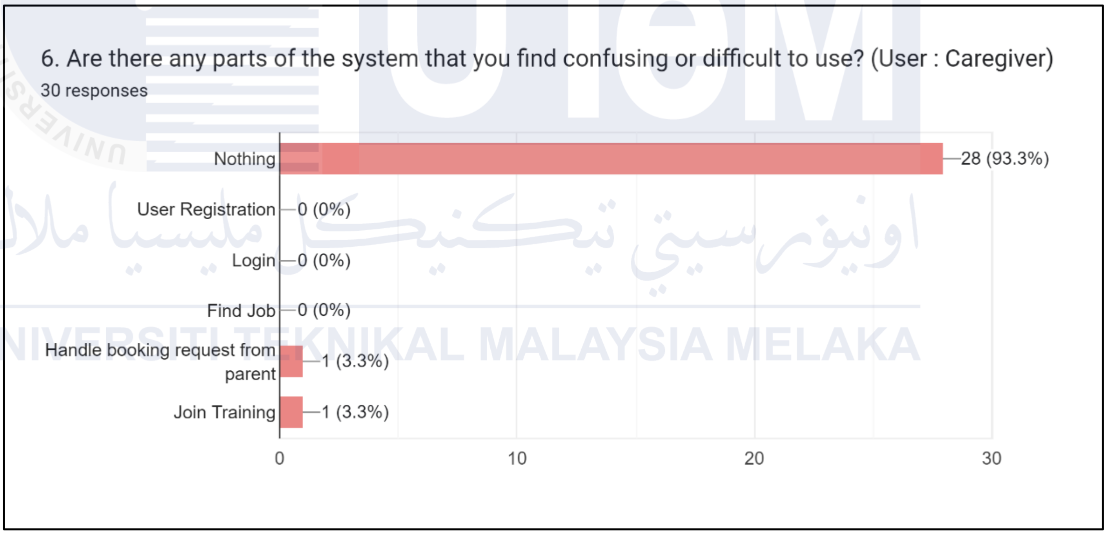
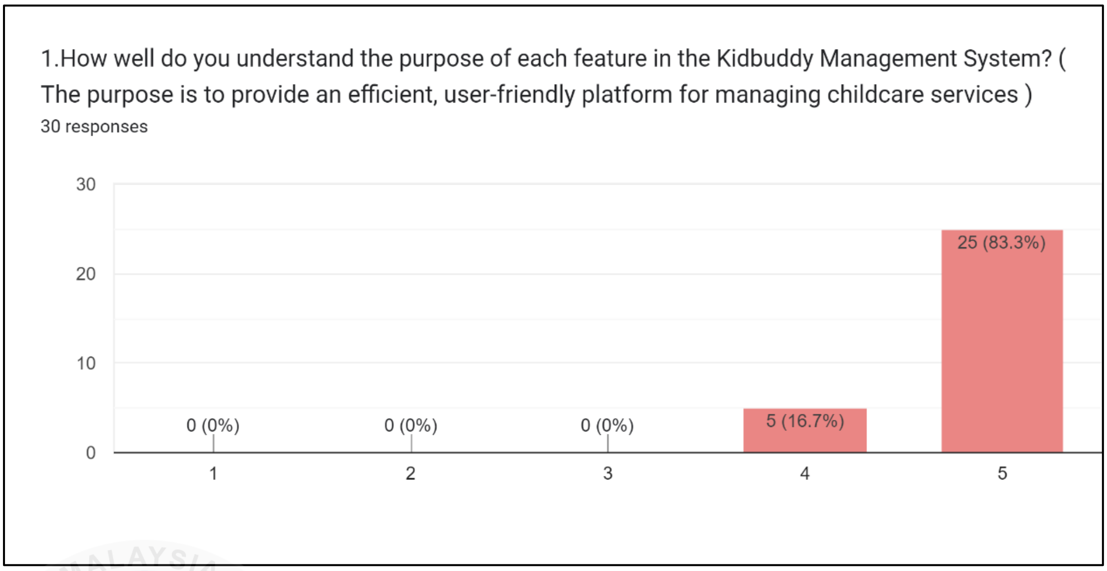
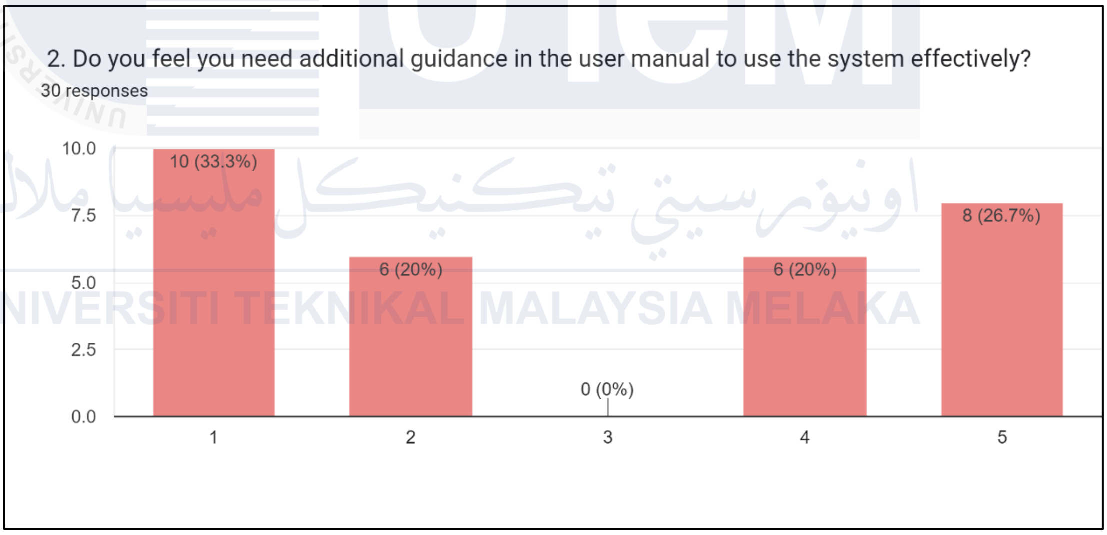


Figure 25: Question 6 System Distribution of Survey Respondents: Majority rate nothing


1.How well do you understand the purpose of each feature in the Kidbuddy Management System? (The purpose is to provide an efficient, user-friendly platform for managing childcare services)

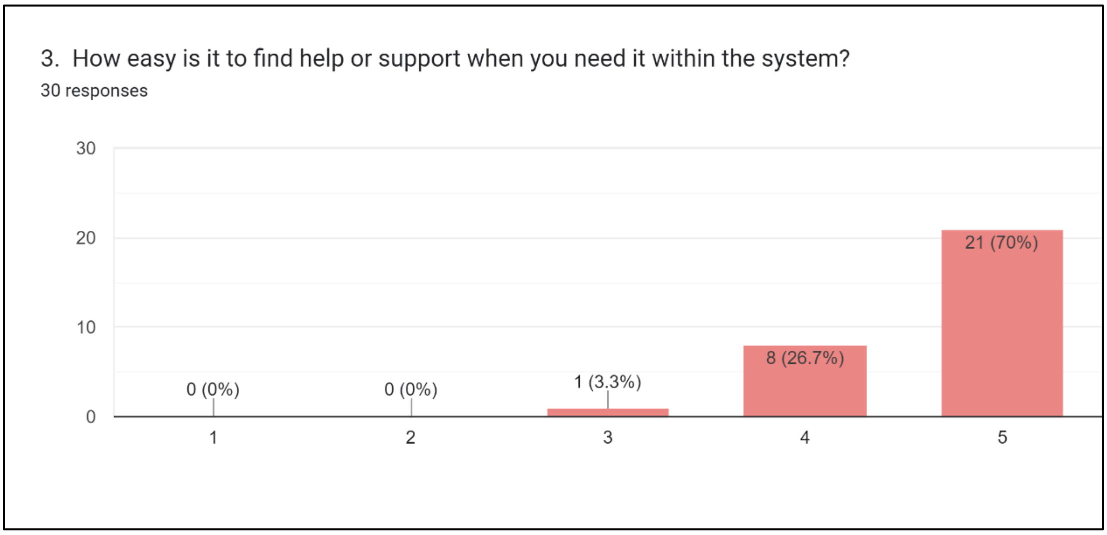
30 responses

Figure 26: Question 1 Knowledge Distribution of Survey Respondents: Majority rate 5


2. Do you feel you need additional guidance in the user manual to use the system effectively?

30 responses

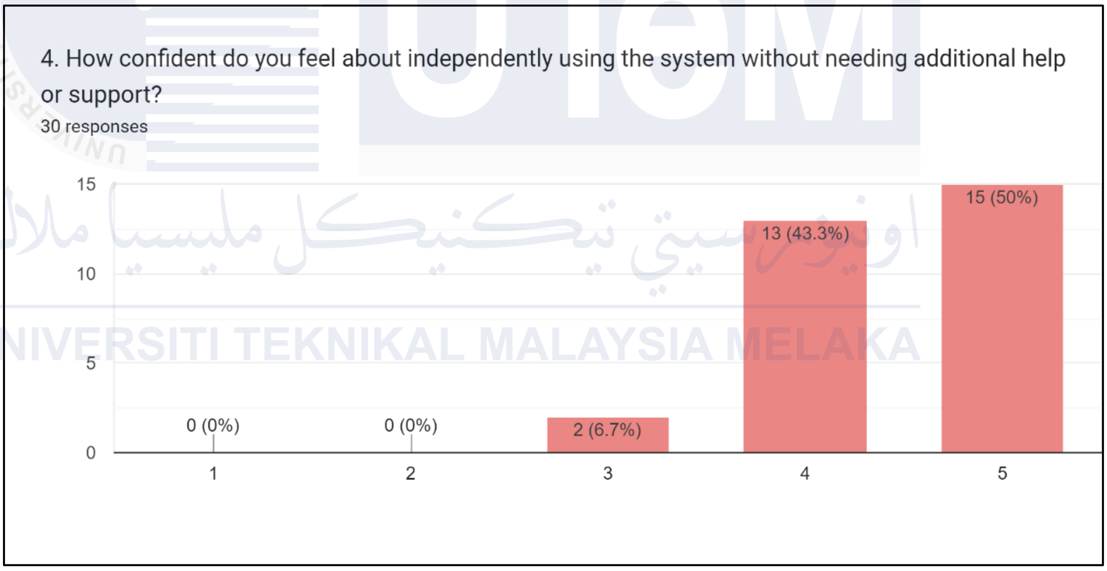
Figure 27: Question 2 Knowledge Distribution of Survey Respondents: Majority rate 1



3. How easy is it to find help or support when you need it within the system?

30 responses

Figure 28: Question 3 Knowledge Distribution of Survey Respondents: Majority rate 5



4. How confident do you feel about independently using the system without needing additional help or support?

30 responses

Figure 29: Question 4 Knowledge Distribution of Survey Respondents: Majority rate 5

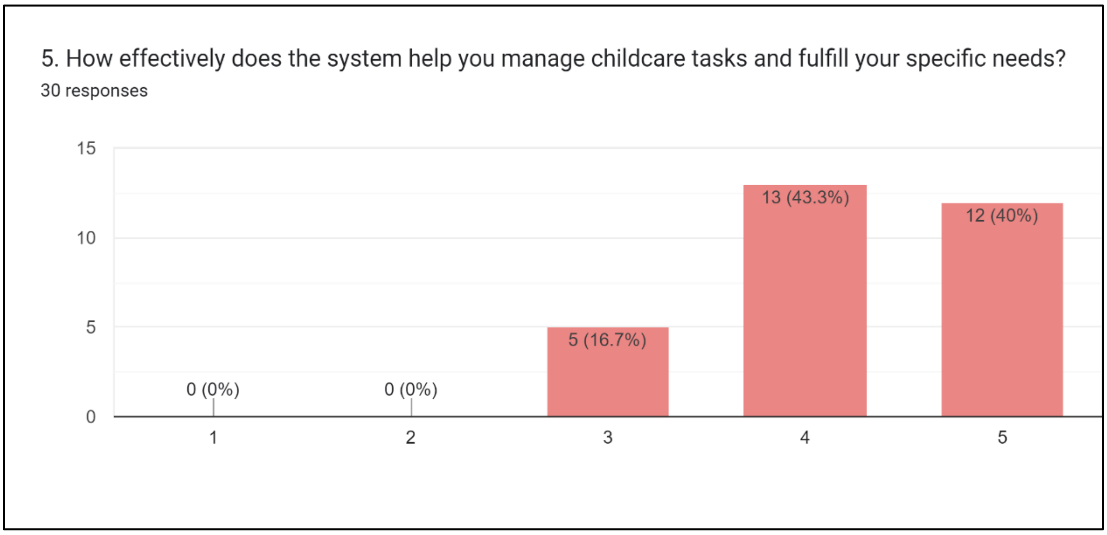
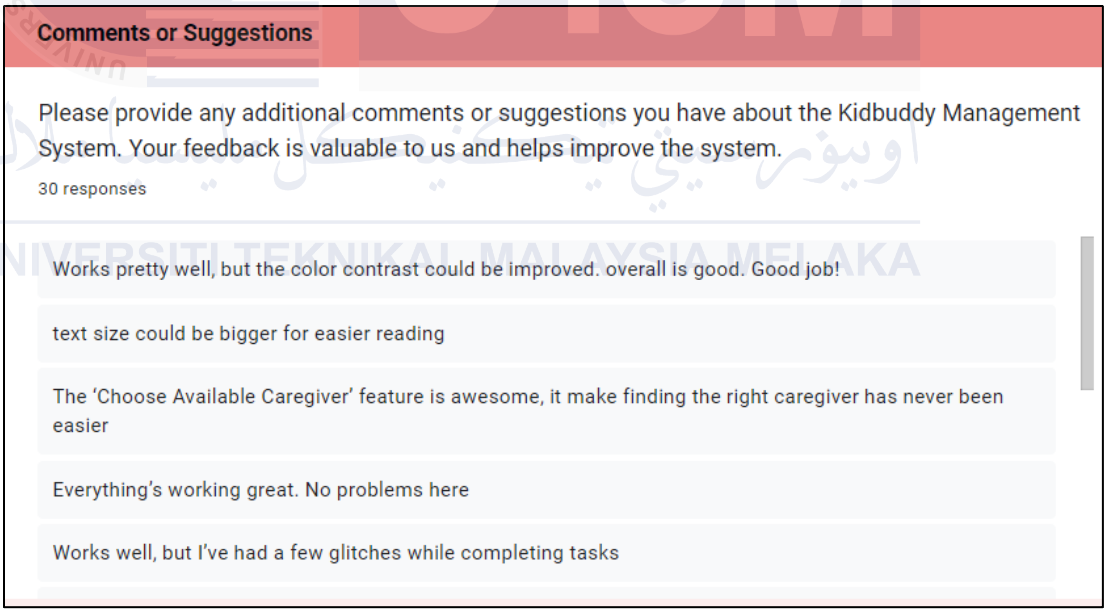


Figure 30: Question 5 Knowledge Distribution of Survey Respondents: Majority rate 4


Comments or Suggestions

Please provide any additional comments or suggestions you have about the Kidbuddy Management System. Your feedback is valuable to us and helps improve the system.

30 responses

Works pretty well, but the color contrast could be improved. overall is good. Good job!

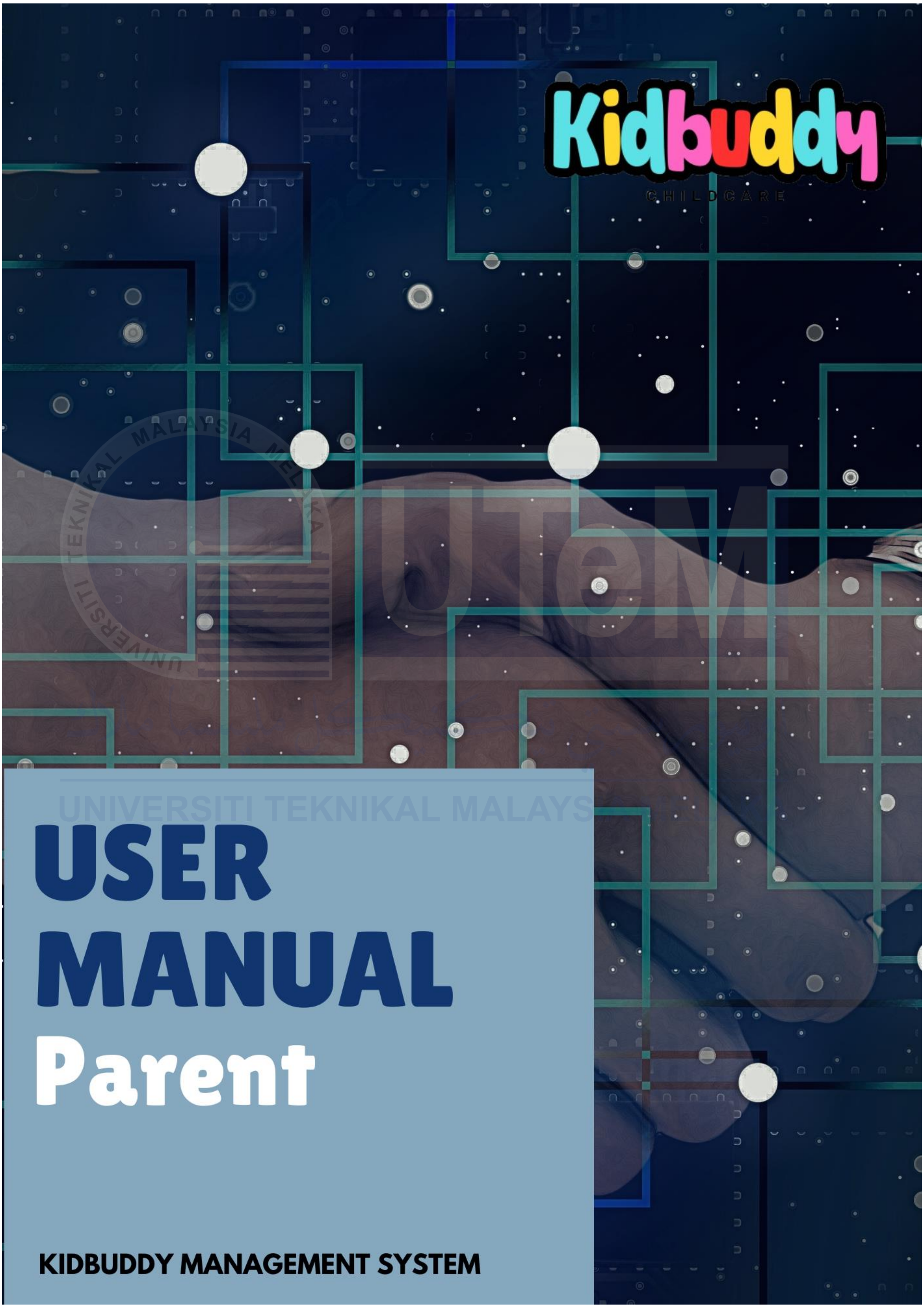
text size could be bigger for easier reading

The ‘Choose Available Caregiver’ feature is awesome, it make finding the right caregiver has never been easier

Everything’s working great. No problems here

Works well, but I’ve had a few glitches while completing tasks

Figure 31: Comment or Suggestion Feedback



USER MANUAL

Parent

KIDBUDDY MANAGEMENT SYSTEM

Table of Contents

1.0 Introduction

Welcome to the Kidbuddy Management System User Manual. This manual is designed to help you navigate and utilize the features of the Kidbuddy Management System effectively. Whether you are booking appointments, managing profiles, or tracking job history, this guide will provide you with step-by-step instructions to make the most of the system.

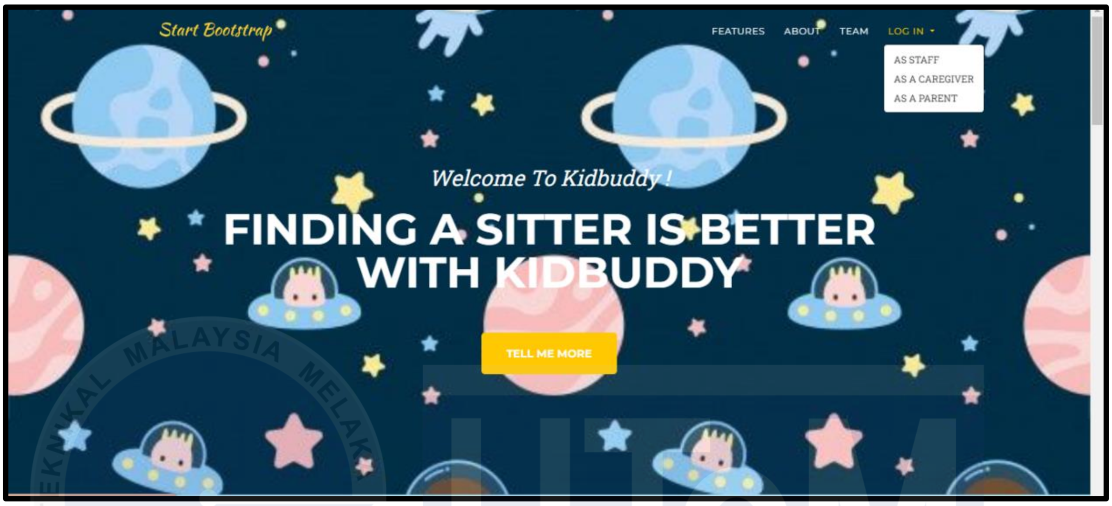
The Kidbuddy Management System is tailored to streamline the management of childcare services, making it easier for parents and caregivers to connect and manage their schedules efficiently. This manual is intended for both new and experienced users, ensuring that everyone can find the information they need.

1.1 How to access Kidbuddy Management System

- Open browser and can be accessed by this address: https://atikah.net/kidbuddy

2.0 Parent Module

2.1 How to register and login to the system

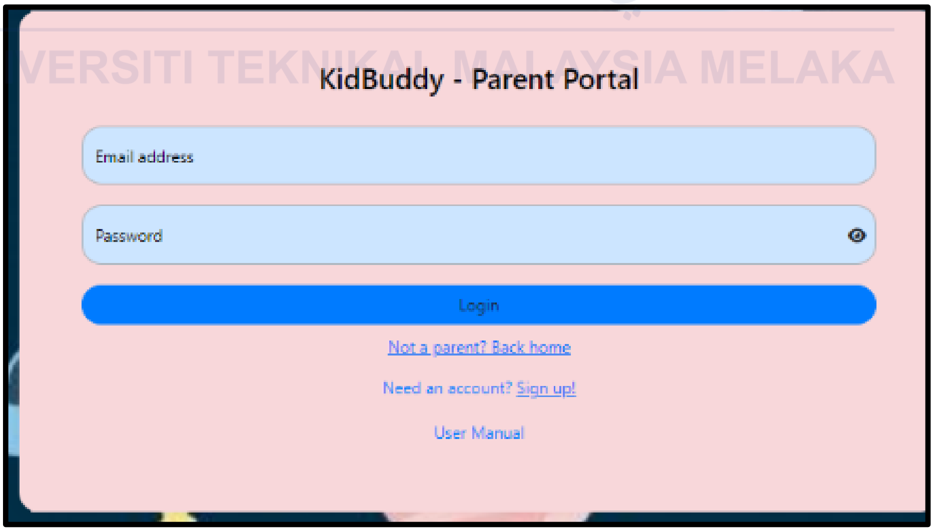


Welcome To Kidbuddy!

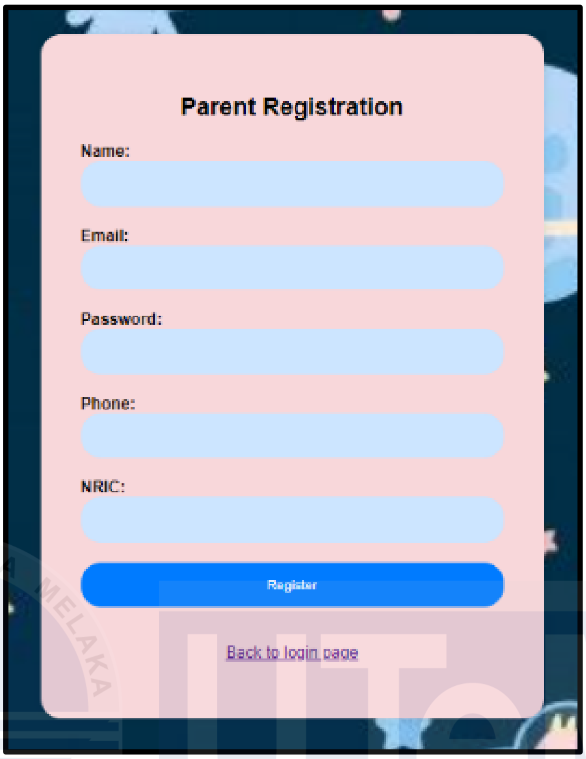
FINDING A SITTER IS BETTER WITH KIDBUDDY

<u>To signup/login, follow the steps below:</u>

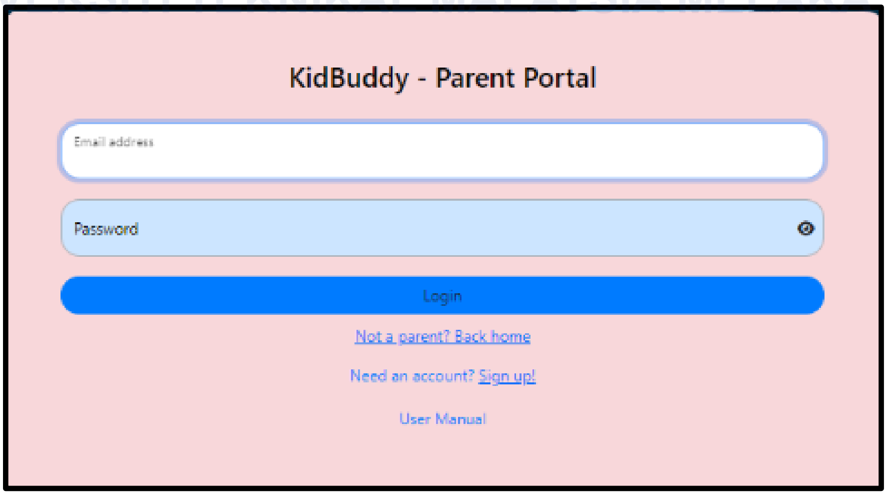
1. Go to homepage of kidbuddy then click **Log in**.
2. Next, click **parent**.


KidBuddy - Parent Portal

1. User are required to sign up first before login.
2. If the user is not a parent, click **'not a parent? Back home'** then it will direct to homepage kidbuddy.

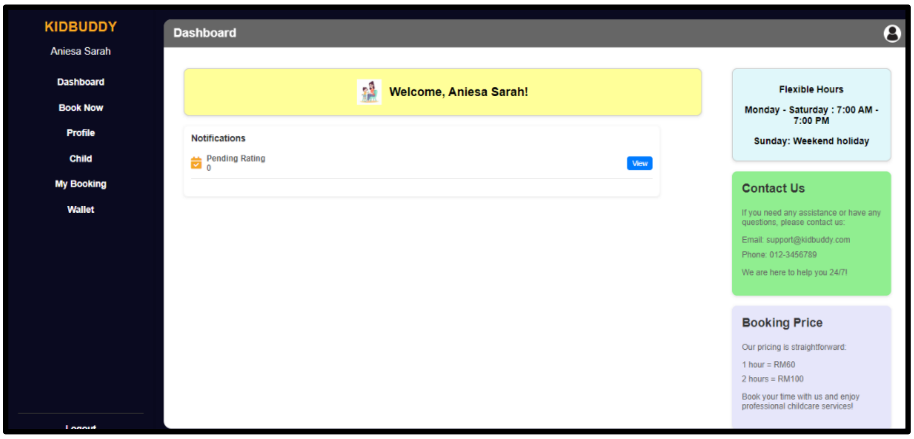


1. User are required to key in their name, email, password, phone and NRIC in the sign-up form
2. If already registered, click **'back to login page'** to proceed with login



1. User are required to key in their email and password that already registered.
2. If the password is wrong, error will show, and user need to try again.
3. Click **Login**.

2.2 Dashboard Parent



Welcome, Aniesa Sarah!

Notifications

Pending Rating
0

Flexible Hours

Monday - Saturday : 7:00 AM - 7:00 PM

Sunday: Weekend holiday

Contact Us

If you need any assistance or have any questions, please contact us:

Email: support@kidbuddy.com

Phone: 012-3456789

We are here to help you 24/7!

Booking Price

Our pricing is straightforward:

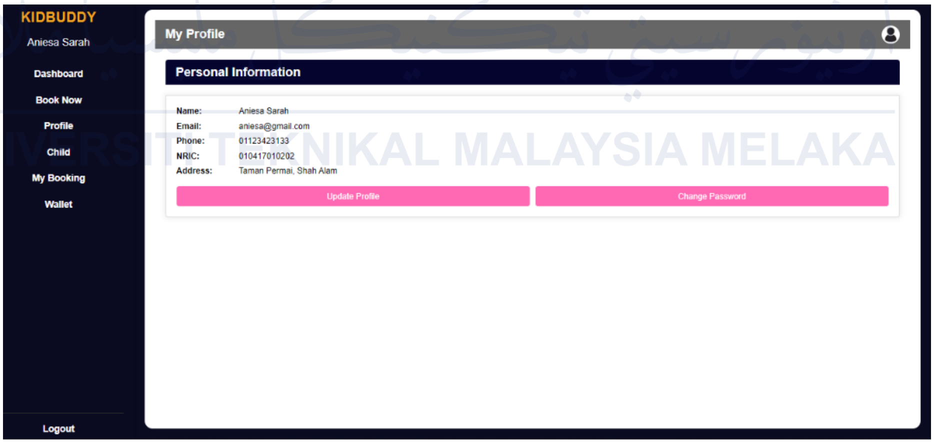
1 hour = RM60

2 hours = RM100

Book your time with us and enjoy professional childcare services!

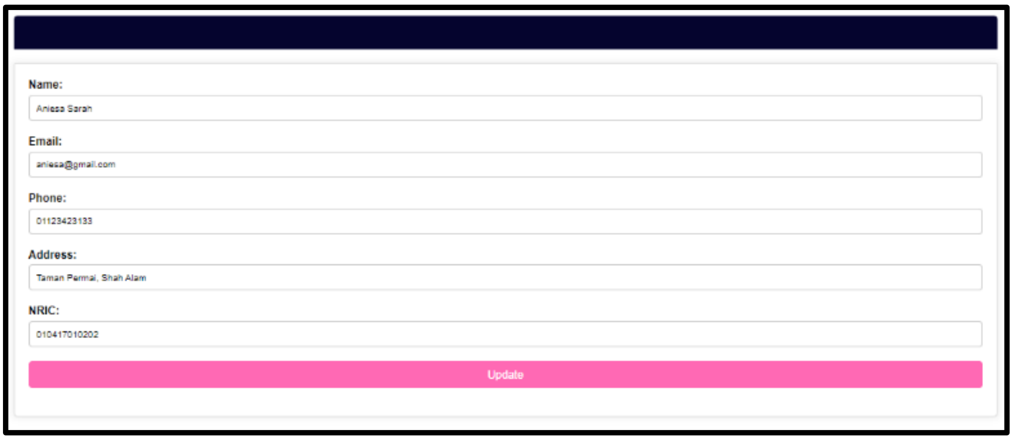
1. Once the user successfully login, this dashboard will appear.
2. On this dashboard, user can find contact information, price information, caregiver working hours information and total of pending rating that user need to view.
3. On the sidebar left side, there are list of features that you can click.

3.3 How to view and update profile



Name: Aniesa Sarah
Email: aniesa@gmail.com
Phone: 01123423133
NRIC: 010417010202
Address: Taman Permai, Shah Alam

1. Click **Profile** then you can view your profile information.
2. Click **Update Profile** to update your profile and **Change Password** to change your password.



Name:

Aniesa Sarah

Email:

aniesa@gmail.com

Phone:

01123423133

Address:

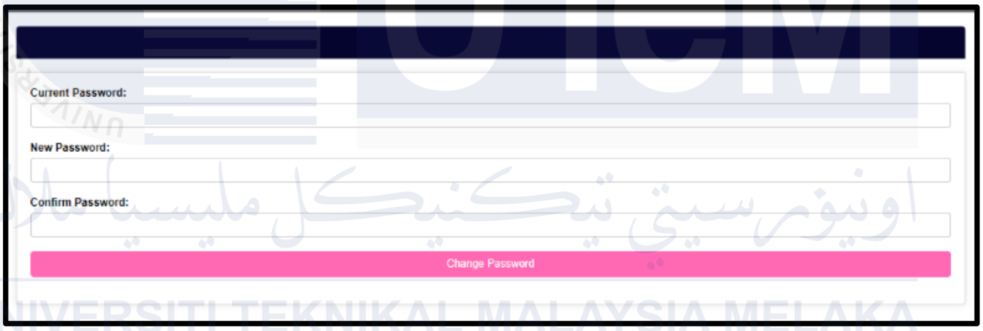
Taman Permai, Shah Alam

NRIC:

010417010202

To update your profile

1. Once completed, click **Update** then your information will successfully be updated



Current Password:

New Password:

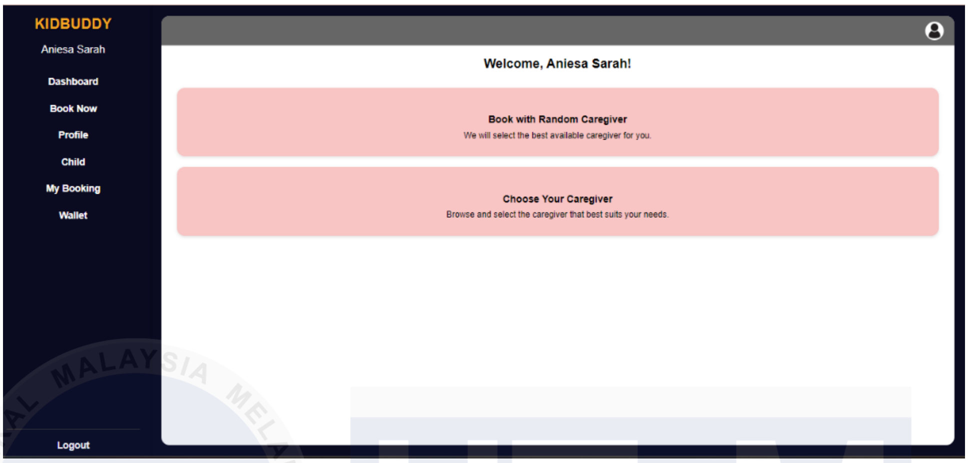
Confirm Password:

To change your password

2. Key in your current password and your new password then click **Change Password** button.

2.4 How to make a booking

2.4.1 Book with random caregiver

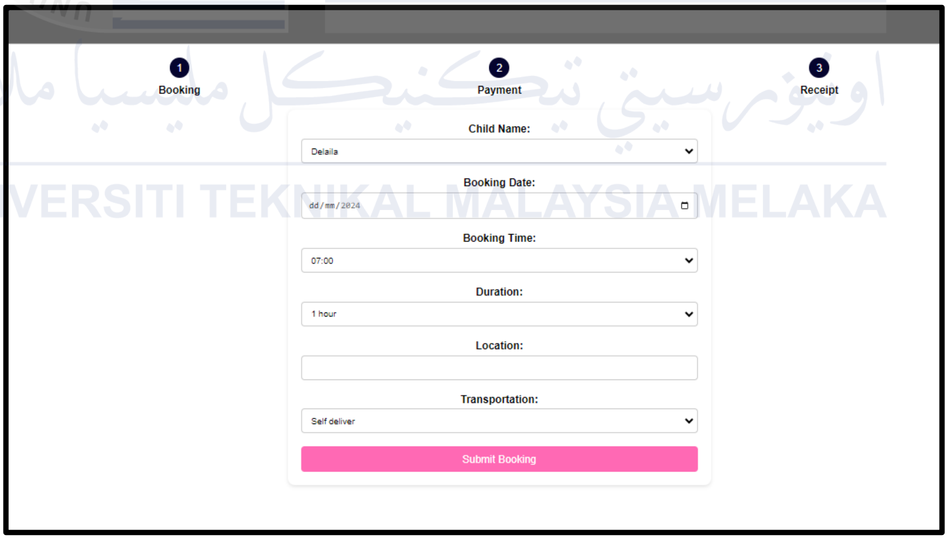


Welcome, Aniesa Sarah!

Book with Random Caregiver
We will select the best available caregiver for you.

Choose Your Caregiver
Browse and select the caregiver that best suits your needs.

1. Click **Book Now** on the sidebar
2. Click **Book with Random Caregiver**



Child Name: Delaila

Booking Date: dd/mm/2024

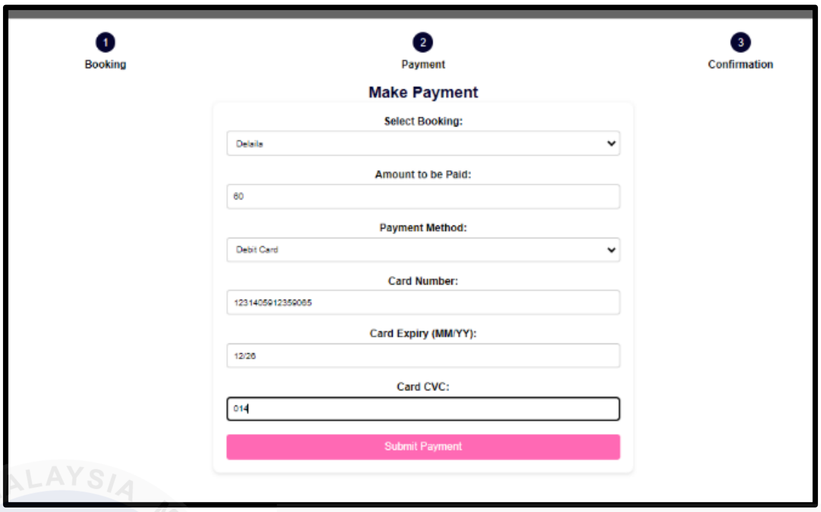
Booking Time: 07:00

Duration: 1 hour

Location:

Transportation: Self deliver

1. Key in your booking details then click **Submit Booking** button.
2. Your child’s name will be auto fill but please make sure you insert your child information first.
3. For booking time, please make sure you do not choose on Sunday.
4. Transportation :
   - 2.4 Self deliver means you can send your child to the caregiver’s place
   - 2.5 Caregiver pickup means caregiver come to your house or they will pickup your child.



Make Payment

Select Booking:

Delaila

Amount to be Paid:

60

Payment Method:

Debit Card

Card Number:

1231405912359065

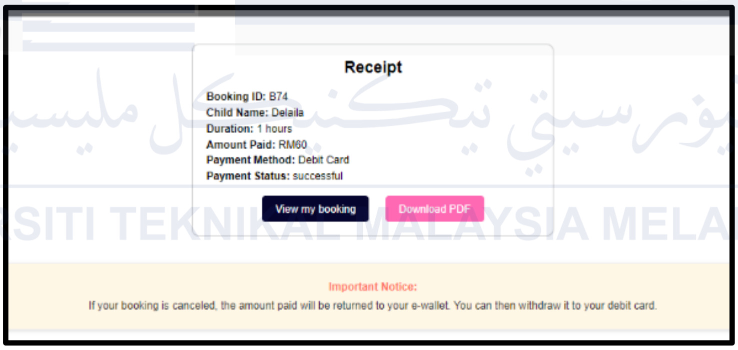
Card Expiry (MM/YY):

12/26

Card CVC:

014

1. After Submit Booking, it will redirect to payment. Key in your payment details then click **Submit Payment.**



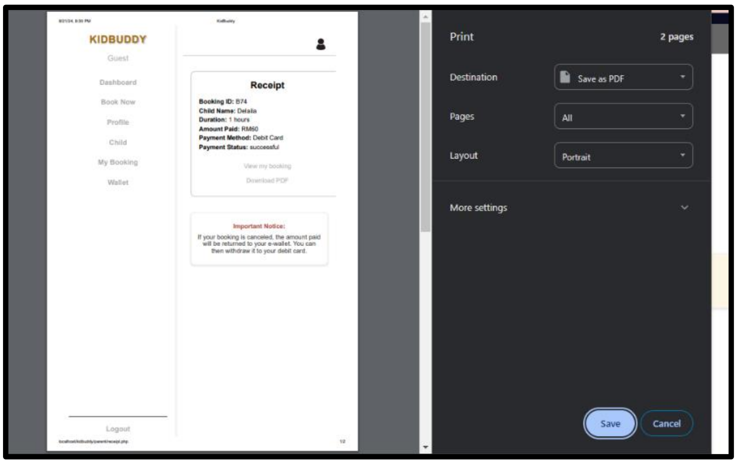
Receipt

Booking ID: B74
Child Name: Delaila
Duration: 1 hours
Amount Paid: RM60
Payment Method: Debit Card
Payment Status: successful

Important Notice:

If your booking is canceled, the amount paid will be returned to your e-wallet. You can then withdraw it to your debit card.

3. Then, it will automatically generate this receipt. Click **View My Booking** to view your booking otherwise click **Download PDF** if you want to save/download the receipt.



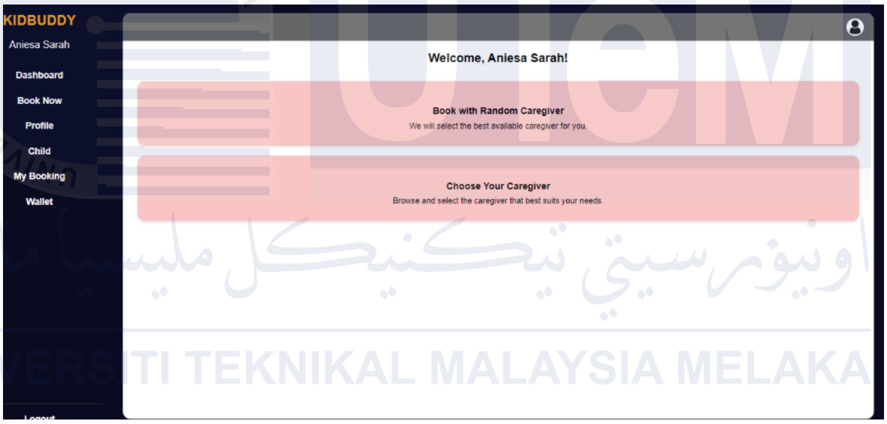
Receipt

Booking ID: B74
Child Name: Delaila
Duration: 1 hours
Amount Paid: RM60
Payment Method: Debit Card
Payment Status: successful

Important Notice:
If your booking is canceled, the amount paid will be returned to your e-wallet. You can then withdraw it to your debit card.

Print 2 pages
Destination: Save as PDF
Pages: All
Layout: Portrait
More settings

2.4.2 Choose your caregiver

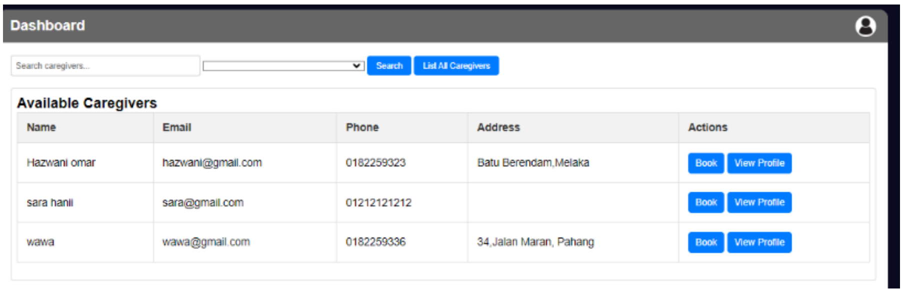


Welcome, Aniesa Sarah!

Book with Random Caregiver
We will select the best available caregiver for you.

Choose Your Caregiver
Browse and select the caregiver that best suits your needs.

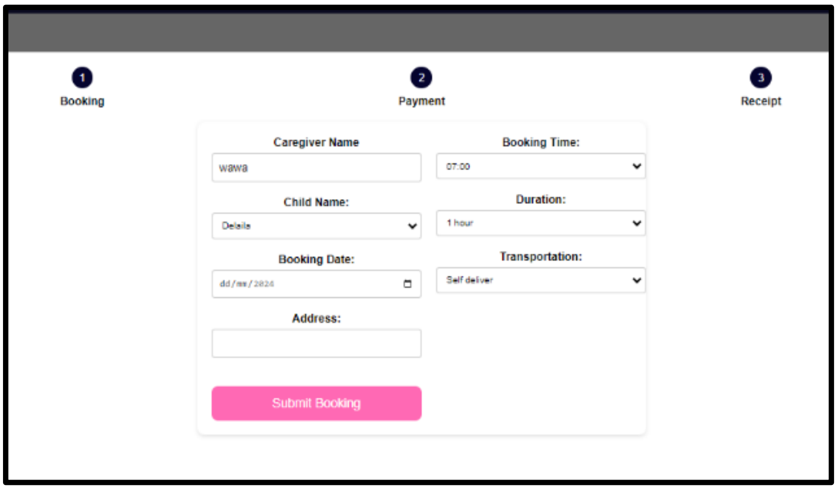
1. Click **Book Now** on the sidebar.
2. Click **Choose Your Caregiver.**



Available Caregivers

Name	Email	Phone	Address	Actions
Hazwani omar	hazwani@gmail.com	0182259323	Batu Berendam,Melaka	Book View Profile
sara hanii	sara@gmail.com	01212121212		Book View Profile
wawa	wawa@gmail.com	0182259336	34,Jalan Maran, Pahang	Book View Profile

1. Click **View Profile** button to view full information of the caregiver.
2. Seach your caregiver by their name and location.
3. Click **Book** button of the caregiver you want to choose



Caregiver Name: wawa

Booking Time: 07:00

Child Name: Delaila

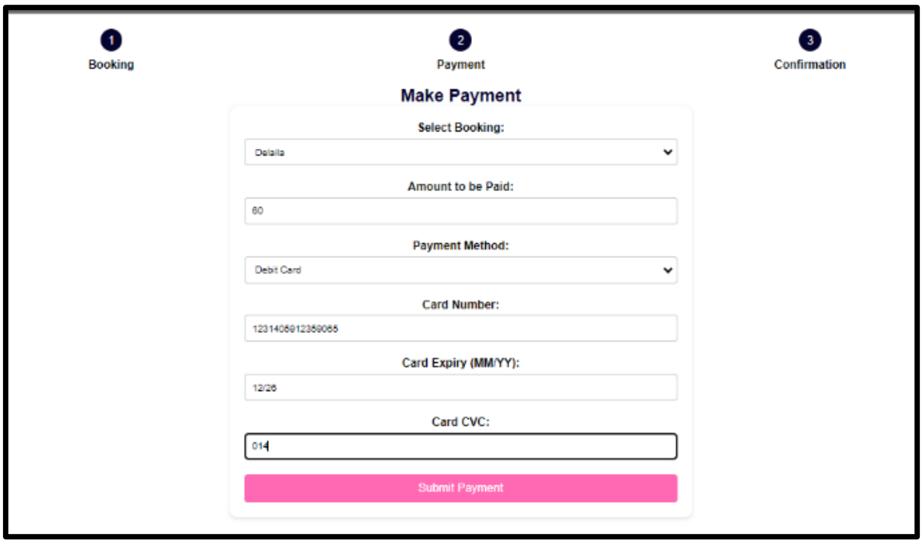
Duration: 1 hour

Booking Date: dd/mm/2024

Transportation: Self deliver

Address:

1. Key in your booking details then click **Submit Booking** button.
2. Your child's will be auto fill but please make sure you insert your child information first.
3. Caregiver name will be auto fill based on the caregiver you choose.
4. For booking time, please make sure you do not choose on Sunday.
5. Transportation:
   - Self-deliver means you can send your child to the caregiver's place
   - Caregiver pickup means caregiver come to your house or they will pick up your child.



Make Payment

Select Booking: Delaila

Amount to be Paid: 60

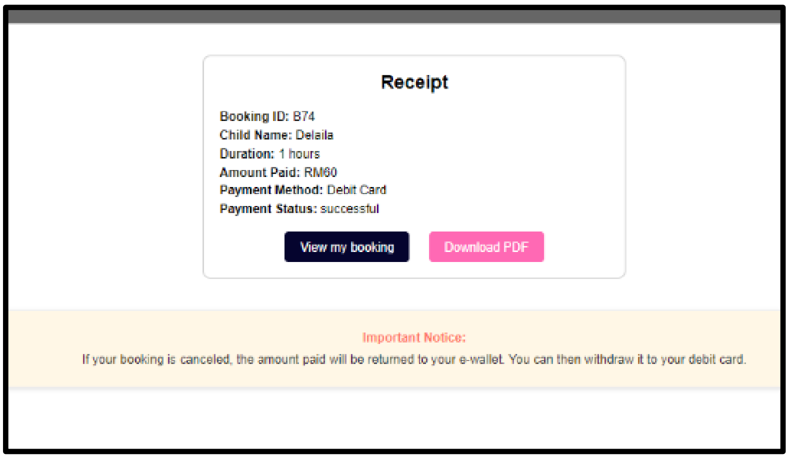
Payment Method: Debit Card

Card Number: 1231405912359065

Card Expiry (MM/YY): 12/26

Card CVC: 014

1. After Submit Booking, it will redirect to payment. Key in your payment details then click **Submit Payment**



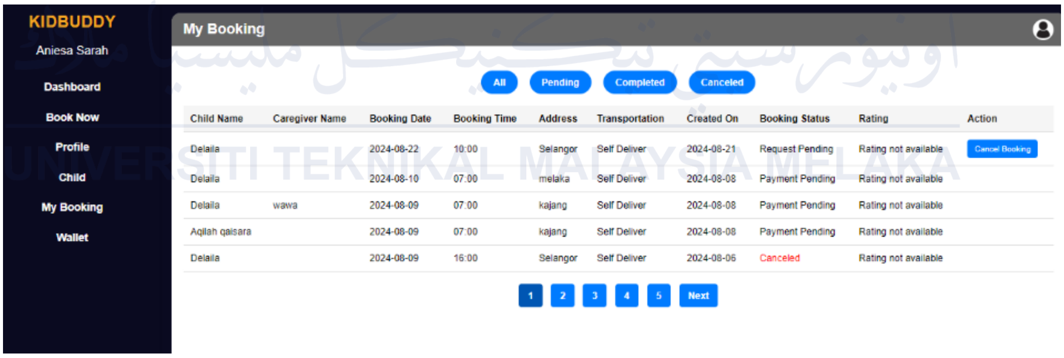
Receipt

Booking ID: B74
Child Name: Delaila
Duration: 1 hours
Amount Paid: RM60
Payment Method: Debit Card
Payment Status: successful

Important Notice:
If your booking is canceled, the amount paid will be returned to your e-wallet. You can then withdraw it to your debit card.

1. Then, it will automatically generate this receipt. Click **View My Booking** to view your booking otherwise click **Download PDF** if you want to save/download the receipt.

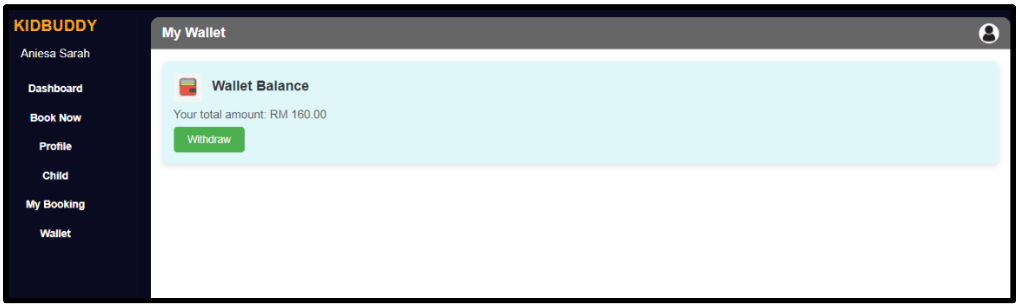
2.5 View Booking



Child Name	Caregiver Name	Booking Date	Booking Time	Address	Transportation	Created On	Booking Status	Rating	Action
Delaila		2024-08-22	10:00	Selangor	Self Deliver	2024-08-21	Request Pending	Rating not available	Cancel Booking
Delaila		2024-08-10	07:00	melaka	Self Deliver	2024-08-08	Payment Pending	Rating not available	
Delaila	wawa	2024-08-09	07:00	kajang	Self Deliver	2024-08-08	Payment Pending	Rating not available	
Aqilah qaisara		2024-08-09	07:00	kajang	Self Deliver	2024-08-08	Payment Pending	Rating not available	
Delaila		2024-08-09	16:00	Selangor	Self Deliver	2024-08-06	Canceled	Rating not available	

1. Click **My Booking** if you want to view all your booking information that have been made and their status.
2. **All** button will list all the booking that have been made.
3. **Pending** button will list all the booking that still pending do not accept by any caregiver.
4. **Completed** button will list all the booking that have been completed.
5. **Canceled** button will list all the booking that cancel means there are no caregiver available for your booking.
6. **Cancel booking** is available when you want to cancel your booking but you only can cancel if there are no caregiver accept your booking and must be canceled 2 hours before your booking time.

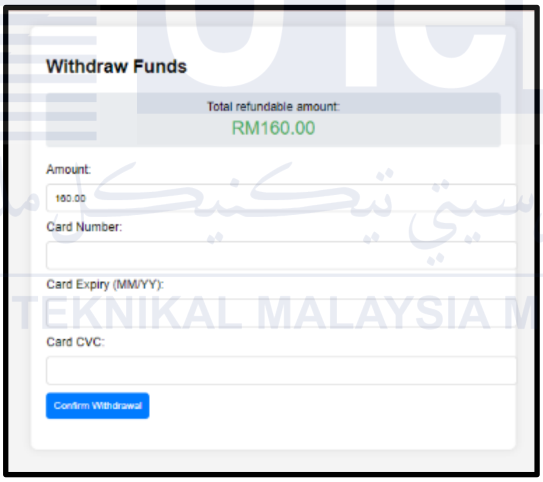
2.6 Withdraw refunded payment



Wallet Balance

Your total amount: RM 160.00

1. Click **Wallet** on the sidebar to view your wallet balance which is all your payment booking that have been canceled will refunded here.
2. Click **Withdraw** button to withdraw your money.



Withdraw Funds

Total refundable amount:
RM160.00

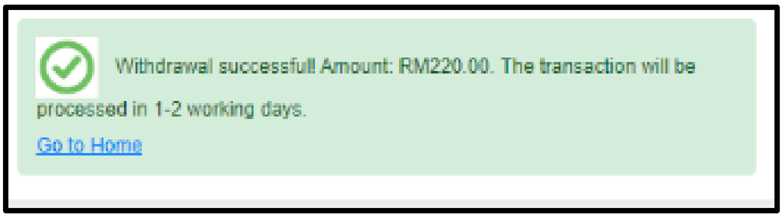
Amount:
160.00

Card Number:

Card Expiry (MM/YY):

Card CVC:

1. The amount in the form will automatically filled with amount that available in your wallet account.
2. Fill in your card information then click **confirm withdrawal** button.



Withdrawal successful! Amount: RM220.00. The transaction will be processed in 1-2 working days.

- Next, once successful, you will get this notification and click **Go To Home** to back to kidbuddy system.