PROJECT TASK MANAGEMENT SYSTEM

MUHAMMAD AMMAR BIN MUSHID

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS LAPORAN

JUDUL: PROJECT TASK MANAGEMENT SYSTEM

SESI PENGAJIAN: 2022/2023

Saya: MUHAMMAD AMMAR BIN MUSHID

mengaku membenarkan tesis Projek Sarjana Muda ini disimpan di Perpustakaan Universiti Teknikal Malaysia Melaka dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan unituk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. * Sila tandakan (✓)

______	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
______	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi / badan di mana penyelidikan dijalankan)
✓	TIDAK TERHAD	

[illegible]

(TANDATANGAN PELAJAR)

Muhammad Ammar Bin Mushid
No 8, Jalan Impian Murni
2, Taman Impian Murni, 43000, Kajang, Selangor

Tarikh: 26 January 2024

[illegible]

(TANDATANGAN PENYELIA)

Ts. Hidayah Binti Rahmalan

Tarikh: 26 January 2024

CATATAN: * Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa.

PROJECT TASK MANAGEMENT SYSTEM

MUHAMMAD AMMAR BIN MUSHID

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

PROJECT TASK MANAGEMENT SYSTEM

is written by me and is my own effort and that no part has been plagiarized

without citations.

STUDENT : MUHAMMAD AMMAR BIN MUSHID Date : 26 January 2024

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 : TS. HIDAYAH BINTI RAHMALAN Date : 26 January 2024

DEDICATION

As well as everything that I do, I dedicate this final year project to my parents, whose unwavering love, support, and encouragement have been the driving force behind my academic journey. Your belief in my abilities and your sacrifices have been instrumental in shaping me into the person I am today. I extend my heartfelt gratitude to my supervisor and lectures at University Technical Malaysia Melaka (UTeM) for their guidance, expertise, and inspiration. Your teachings and insights have broadened my knowledge and nurtured my intellectual growth. I am deeply grateful to my friends and classmates for their camaraderie and collaboration. Your enthusiasm and shared experiences have made this journey memorable and enjoyable. Together, we have overcome challenges and celebrated achievements, forging lifelong connections. I would like to express my appreciation to the industry professionals and experts who have generously shared their knowledge and insights, enriching my understanding of the requirement during analysis phase. Your expertise has been invaluable in shaping the trajectory of my project. Finally, I dedicate this project to myself, as a testament to my dedication, perseverance, and hard work. This project represents the culmination of years of study, late nights, and countless hours of research and coding. It is a reflection of my passion for computer science in database management and my commitment to continuous learning. May this project serve as a stepping stone towards a successful career in the field of computer science, and may it contribute to the advancement of project task management.

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to the following individuals and organizations who have contributed to the successful completion of my final year project:

First and foremost, I am immensely thankful to my supervisor, Ts Hidayah binti Rahmalan, for their invaluable guidance, expertise, and unwavering support throughout this project. Their deep knowledge, constructive feedback, and constant encouragement have been instrumental in shaping the direction and quality of my work. I am truly grateful for their mentorship and the time they have dedicated to helping me grow as a researcher.

I would like to extend my appreciation to the faculty members of the Software Engineering Department at University Technical Malaysia Melaka for imparting their knowledge and providing a conducive learning environment. Their dedication to excellence in education has played a significant role in shaping my academic journey.

Furthermore, I would like to express my appreciation to the open-source community and developers worldwide for their contributions to the field of computer science. Their collective efforts have provided a wealth of tools, libraries, and resources that have been indispensable in the development and implementation of this project.

Lastly, I would like to thank my family and friends for their unconditional love, encouragement, and patience throughout my academic journey. Their unwavering support and belief in my abilities have been a constant source of motivation.

To everyone mentioned above and to those who have contributed in ways I may have unintentionally omitted, please accept my sincere gratitude for your invaluable contributions to the successful completion of this final year project

ABSTRACT

The "Project Task Management System" is a comprehensive and user-friendly software application developed to enhance project task management in various industries and domains. The system utilizes database management techniques to streamline project workflows, improve task efficiency, and facilitate collaboration among project team members. The project aims to address the challenges associated with task management, including task assignment, tracking, and progress monitoring. By leveraging intelligent algorithms and data-driven decision making, the system prioritizes tasks based on factors such as deadlines, dependencies, and resource availability, ensuring optimal allocation of project resources and timely completion of project milestones. The system provides a user-friendly interface that allows project managers and team members to create tasks, assign responsibilities, set deadlines, and define task dependencies. Real-time updates and notifications enable stakeholders to monitor task progress, identify potential bottlenecks, and take proactive measures to ensure smooth project execution. Additionally, the system incorporates features such as resource management, generating reports and analytics, and facilitating communication and collaboration among team members. These functionalities contribute to efficient resource allocation, data-driven decision making, and seamless information sharing, fostering effective teamwork and enabling informed project management. Through the implementation of an appropriate database schema and the integration of intelligent algorithms, the system offers a robust foundation for managing project tasks. It empowers project managers to make informed decisions, optimize resource utilization, and improve project outcomes. The results of this project demonstrate the effectiveness and practicality of the Project Task Management System. It serves as a valuable tool for organizations and individuals involved in project management, offering a streamlined and efficient approach to task management and fostering successful project completion.

ABSTRAK

“Project Task Management System” merupakan aplikasi perisian yang komprehensif dan mesra pengguna yang dibangunkan untuk meningkatkan pengurusan tugas projek dalam pelbagai industri dan bidang. Sistem ini menggunakan teknik pengurusan pangkalan data untuk menyelaraskan alur kerja projek, meningkatkan kecekapan tugas, dan memudahkan kerjasama di kalangan ahli pasukan projek. Projek ini bertujuan untuk menangani cabaran yang berkaitan dengan pengurusan tugas, termasuk penugasan tugas, penjejakan, dan pemantauan kemajuan. Dengan menggunakan algoritma berintelek dan pengambilan keputusan berasaskan data, sistem ini memberi keutamaan kepada tugas berdasarkan faktor seperti tarikh akhir, ketergantungan, dan ketersediaan sumber, memastikan penggunaan sumber projek yang optimum dan penyelesaian tepat pada masanya. Sistem ini menyediakan antara muka mesra pengguna yang membolehkan pengurus projek dan ahli pasukan mencipta tugas, menetapkan tanggungjawab, menetapkan tarikh akhir, dan menentukan ketergantungan tugas. Pembaruan dan pemberitahuan secara masa nyata membolehkan pemegang berkepentingan memantau kemajuan tugas, mengenal pasti halangan yang mungkin timbul, dan mengambil langkah proaktif untuk memastikan pelaksanaan projek yang lancar. Selain itu, sistem ini mempunyai ciri-ciri seperti pengurusan sumber, penyediaan laporan dan analitik, serta memudahkan komunikasi dan kerjasama di antara ahli pasukan. Fungsi-fungsi ini menyumbang kepada penggunaan sumber yang berkesan, pengambilan keputusan berasaskan data, dan perkongsian maklumat yang lancar, mendorong kerja berpasukan yang berkesan dan membolehkan pengurusan projek yang terarah. Melalui pelaksanaan skema pangkalan data yang sesuai dan integrasi algoritma berintelek, sistem ini menyediakan asas yang kukuh untuk pengurusan tugas projek. Ia membolehkan pengurus projek membuat keputusan yang berinformasi, mengoptimumkan penggunaan sumber, dan meningkatkan hasil projek. Keputusan projek ini menunjukkan keberkesanan dan kebolehgunaan Sistem Pengurusan Tugas Projek Berintelek. Ia berfungsi sebagai alat yang berharga untuk organisasi dan individu yang terlibat dalam pengurusan projek, menawarkan pendekatan pengurusan tugas yang lancar dan cekap serta memperkasa penyelesaian projek yang berjaya.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

FYP	**-**	**Final Year Project**
PTMS	**-**	**Project Task Management System**
DFD	**-**	**Data Flow Diagram**
DBLC	**-**	**Database Life Cycle**
ERD	**-**	**Entity Relationship Diagram**
GUI	**-**	**Graphical User Interface**
DDL	**-**	**Data Definition Language**
DML	**-**	**Data Manipulate Language**
PHP	**-**	**Hypertext Preprocessor**
OS	**-**	**Operating System**
CRUD	**-**	**Create, Read, Update, Delete**
SDLC	**-**	**Software Development Life Cycle**

CHAPTER 1: INTRODUCTION

1.1 Project Background

SE Infinity Sdn Bhd, a software and electrical company, specializes in software application development and system integration. With a portfolio of over a hundred self-developed systems, the company is involved in numerous projects and tenders. However, they face challenges in effectively managing and tracking the progress of these projects and tenders. Additionally, the human resource department struggles to identify the individuals responsible for each project or tender. Furthermore, when SE Infinity acquires new projects or tenders, they encounter difficulties in assigning the right staff members, as they lack visibility on the skill sets and capabilities of their employees.

These challenges highlight the need for a robust project task management system within SE Infinity. Such a system would streamline project and tender tracking, facilitate efficient communication and collaboration among team members, and enable the human resource department to easily identify suitable staff members for each project.

By implementing a comprehensive project task management system, SE Infinity can address these issues and enhance their overall project management process. The system would provide a centralized platform for tracking the progress of projects and tenders, allowing project managers to monitor milestones, allocate resources effectively, and identify potential bottlenecks. Additionally, the system would include features such as task assignment, deadline setting, and task dependency management, enabling smooth coordination and ensuring project timelines are met.

Moreover, the system would incorporate a user-friendly interface, allowing staff members to view their assigned tasks, update task progress, and collaborate with team members. This would improve communication channels and enhance productivity within the organization.

To address the challenge of assigning new projects or tenders, the project task management system would include a comprehensive staff profile database. This database would store information on the skills, expertise, and past project experiences of each employee. By leveraging this database, project managers and the human resource department can easily identify suitable staff members for new projects based on their qualifications and availability.

Overall, the implementation of a project task management system within SE Infinity would streamline their project and tender management processes, improve project tracking, enhance communication and collaboration, and facilitate efficient resource allocation. This system would empower SE Infinity to optimize their project management capabilities and successfully deliver projects and tenders with increased efficiency and effectiveness.

1.2 Problem Statement

i. **Lack of Project and Tender Tracking.**

SE Infinity struggles to efficiently follow up on the progress of their ongoing projects and tenders. They did not have information that allows them to monitor milestones, track task completion, and identify potential bottlenecks.

ii. **Difficulty in Identifying Project/Tender Ownership.**

The company's human resource department finds it challenging to determine the client or contact person who has authentication for each project or tender. This lack of clarity leads to confusion and delays in project execution.

iii. **Inefficient Assignment of New Projects/Tenders**

SE Infinity encounters difficulties in assigning new projects or tenders to the most suitable staff members. The company lacks visibility in the skills and capabilities of their employees, making it challenging to match project requirements with the right personnel.

1.3 Objective

i. Able to Manage Projects and Tenders Tracking

Create a comprehensive system that allows SE Infinity to track the progress of their projects and tenders in real-time. This system should provide visibility into project milestones, task completion, and potential bottlenecks to facilitate effective project management.

ii. Establish Clear Ownership for Projects and Tenders

Implement a mechanism within the system that enables the clear identification of the client or contact person who has authentication for each project or tender. This feature will eliminate confusion and delays by ensuring accountability and clear lines of communication.

iii. Enhance Assignment of New Projects and Tenders

Develop a system that enables efficient assignment of new projects and tenders to suitable staff members. This system should leverage employee profiles, including skills, expertise, and past project experience, to match project requirements with the most qualified individuals.

1.4 Scope

1.4.1 Target User

The scope of the system can be divided into target user and modules.

i. Project Manager

Project managers will have access to project-related functionalities. They can create and manage projects, assign tasks to team members, monitor task progress, and track project milestones. They can also generate reports and analyze project performance.

ii. Staff

Staff will have access to assigned tasks, task details, and project-related communication. They can update task progress, collaborate with other team members, and submit task deliverables through the system.

1.4.2 Modules

i. Staff Management

This module is responsible for managing user accounts within the Project Task Management System. It includes functions such as user registration, authentication, and profile management.

ii. Project Management

The Project Management module enables the creation, tracking, and monitoring of projects. Users can create new projects, define project details, and set milestones and deadlines. The module allows project managers to assign tasks to team members, track task progress, and manage project timelines. It provides a centralized view of project status and facilitates effective project management.

iii. Task Management

The Task Management module handles the creation, assignment, and tracking of individual tasks within projects. Users can create tasks, set priorities, and define deadlines. Tasks can be assigned to specific team members, and progress can be tracked in real-time. This module enables efficient task delegation, monitors task dependencies, and provides notifications to keep everyone informed about task updates.

iv. Reporting and Analytics

The Reporting and Analytics module generates reports and provides data analysis capabilities for project evaluation. It includes features to generate project status reports, track performance metrics, and visualize data insights.

1.5 Project Significant

The development and implementation of the Project Task Management System at SE Infinity will bring significant benefits to the organization. The system will streamline project management processes, improve resource allocation, establish clear project ownership, and enable timely monitoring of project progress. It will enhance collaboration and communication among team members, facilitate efficient assignment of new projects, and support data-driven decision-making. Overall, the system's implementation will lead to improved project efficiency, productivity, and customer satisfaction, contributing to SE Infinity's growth and success in the software and electrical industry.

1.6 Expected Outcome

The implementation of the Project Task Management System is expected to result in improved project efficiency, enhanced resource utilization, clear project ownership and accountability, timely project monitoring and decision-making, effective communication and collaboration, better project planning and forecasting, increased customer satisfaction, and scalability for future growth. The system will streamline project management processes, optimize resource allocation, foster effective teamwork, provide real-time project insights, and contribute to successful project delivery.

1.7 Conclusion

In conclusion, the Project Task Management System proposed for SE Infinity addresses the challenges faced by the organization in effectively managing their projects and tenders. By providing a centralized platform for project tracking, clear ownership assignment, and efficient resource allocation, the system streamlines project management processes and enhances collaboration among team members. With real-time monitoring, timely updates, and data-driven insights, the system enables informed decision-making, improves project efficiency, and drives overall productivity.

CHAPTER 2: PROJECT METHODOLOGY AND PLANNING

2.1 Introduction

This chapter provides an overview of the chosen development methodology for this project. Various System Development Life Cycle (SDLC) methodologies, such as the waterfall model, agile model, and spiral model, can be employed to design, develop, and test high-quality software. For this project, the Database Life Cycle (DBLC) methodology has been selected as it encompasses the key stages of database initial study, design, implementation & loading, training and evaluation, and operations and maintenance & evaluation. The DBLC is a continuous process as it includes activities such as monitoring, modification, and maintenance that extend beyond the initial implementation phase, effectively covering the lifetime of the database.

2.2 Project Methodology

The project methodology section outlines the approach and framework used in this project to achieve its objectives. It encompasses the systematic steps and activities involved in the development and implementation of the Project Task Management System. Various methodologies, such as waterfall, agile, and spiral models, exist for software development, each with its own advantages and considerations. For this project, the Database Life Cycle (DBLC) methodology has been chosen as it provides a comprehensive framework that covers the entire lifecycle of the database, including initial study, design, implementation, training, evaluation, and ongoing operations and maintenance. The selected methodology ensures a structured and efficient approach to project execution, enabling effective planning, development, and delivery of the Project Task Management System.

2.2.1 Database Initial Study

The database initial study is a crucial step in addressing the challenges faced by SE Infinity Sdn Bhd in managing their projects and tenders efficiently. This study aims to thoroughly examine the existing database system and gather relevant information to guide the development of the Project Task Management System. Key aspects of the database initial study include assessing the current database structure, identifying data dependencies and relationships, evaluating data quality and integrity, and understanding the performance requirements. Additionally, security and privacy considerations, regulatory compliance, and the available technology infrastructure will be analyzed. The database initial study will provide valuable insights and recommendations for designing and implementing an improved database system that caters to the specific needs of SE Infinity Sdn Bhd, ensuring smoother project management, streamlined task allocation, and enhanced resource utilization.

2.2.2 Database Design

The second phase of the project focuses on designing the database model to meet the operational and objective requirements of the system. This phase is crucial for ensuring that the final product aligns with the system's needs. It comprises four sub-phases: conceptual design, DBMS software selection, logical design, and physical design. In the conceptual design phase, an entity-relationship diagram (ERD) is created, and normalization is applied to ensure data consistency and integrity. For the DBMS software, MySQL is chosen as the preferred option. In the logical database design phase, a shared understanding of data definitions and business logic is established. A relational data model is developed to specify data structures and queries based on the identified data requirements. Finally, in the physical design phase, the system is translated into tables, columns, indexes, sequences, and constraints. Attributes from the logical database are grouped to represent core business rules and detailed data relationships at a granular level.

2.2.3 Implementation and Loading

During this phase, the DBMS software, namely MySQL, is installed on the computer and fine-tuned to optimize its performance based on the hardware, software, and usage conditions. The databases are then created using the Data Definition Language (DDL) in alignment with the Entity-Relationship Diagram (ERD). Data Manipulation Language (DML) is employed to insert data into the database tables. To develop the user interface, MAMP APACHE 2.4.10 (Win32) and PHP 5.5.15 with the hostname "Post" (localhost: 80) are utilized. Additionally, database performance, adherence to security standards, data integrity, and backup recovery procedures are critical considerations throughout the implementation phase.

2.2.4 Testing and Evaluation

In this phase, there are three sub phases which are test the database, fine-tune the database, evaluate the database and its application programs is used to test on this system to ensure integrity, security, performance, backup and recovery of the database.

i. Test the database.

During the database testing phase, the focus is on ensuring data integrity and security, as well as assessing system performance. Data integrity is upheld by the DBMS through the implementation of primary keys, foreign keys, unique constraints, and other mechanisms. Data security is evaluated through tests involving password security, data encryption, and the assignment of privileges and access rights. The testing phase includes both unit testing and system testing. Unit testing involves evaluating individual functions such as login, admin, and candidate functionalities, using sample data to verify database connections and validate correct data input. System testing, on the other hand, verifies the overall business process of the system, ensuring that it runs smoothly and without any errors.

ii. Fine-Tune the database.

In this phase, the database is fine-tuned to optimize its performance based on the specific challenges faced by SE Infinity. This includes adjusting the setup variables of the DBMS software, MySQL, according to the hardware, software, and usage conditions. The goal is to enhance the efficiency and responsiveness of the database system. Additionally, attention is given to ensuring data integrity and security. This involves enforcing data integrity constraints, such as primary keys, foreign keys, and unique constraints, to maintain the accuracy and consistency of the data. Security measures, such as password policies, data encryption, and access controls, are implemented to protect sensitive information. Through these fine-tuning efforts, the database system is optimized to meet the unique requirements of SE Infinity, ensuring reliable performance, data integrity, and secure operations.

iii. Evaluate the Database and Its Application Programs.

The evaluation of the database and its application programs is a comprehensive process that involves assessing performance, data integrity, functionality, usability, and security. It includes analyzing system performance to ensure efficient operations, verifying data integrity and accuracy, testing functionality and usability to meet requirements, and evaluating security measures. The evaluation provides valuable insights to optimize the system, delivering a reliable, efficient, and secure solution for SE Infinity Sdn Bhd.

2.2.5 Operation

Following the evaluation process, the system undergoes transformation into a fully functional information system. At this stage, users of the system commence its operation, engaging in activities such as data loading, data management, and accessing reports to facilitate the desired flow of information.

2.2.6 Maintenance and Evolution

In the database maintenance phase, various tasks are performed to ensure the smooth operation and upkeep of the database. These tasks include general maintenance activities like index maintenance, table optimization, user management (adding/removing users, changing passwords), and regular backups to prevent data loss. Additionally, as new system requirements or organizational changes arise, requests for modifications and enhancements are addressed through adaptive maintenance. This may involve the creation of new fields or tables to improve performance and accommodate evolving needs. The database maintenance phase plays a vital role in sustaining the database's performance, security, and adaptability to meet the changing demands of the system and organization.

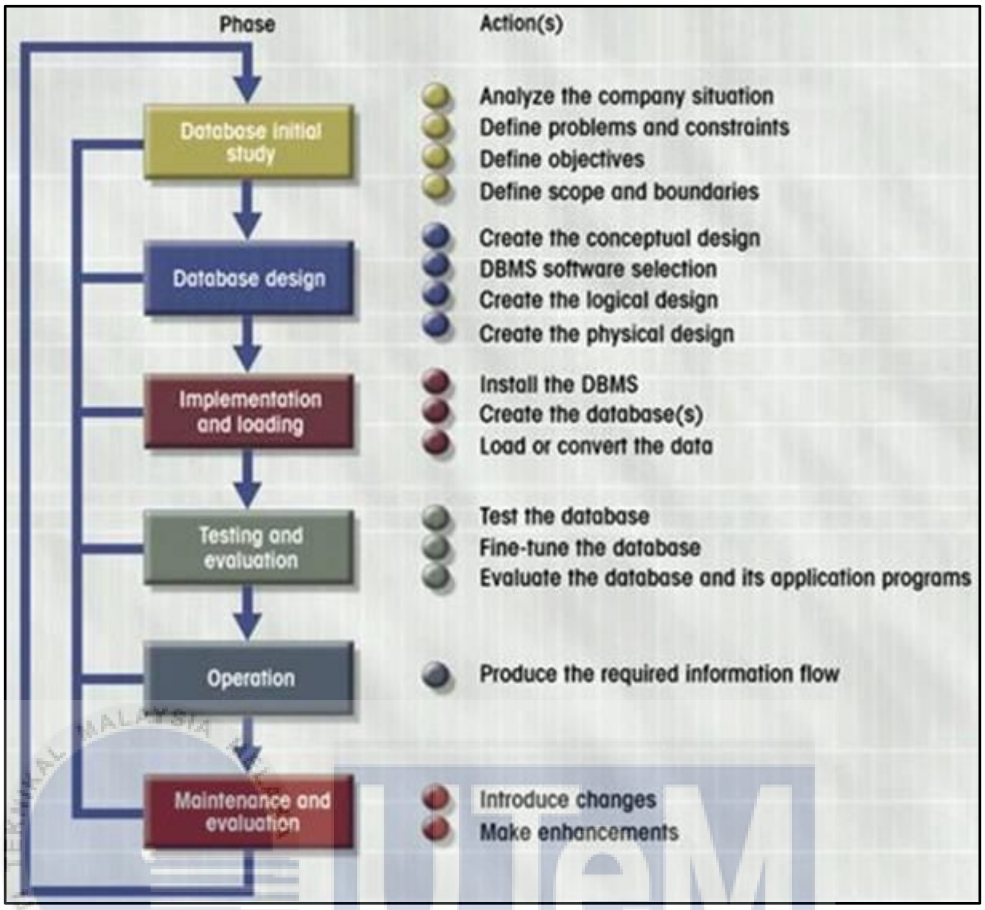

Phase
Action(s)
Database initial study
Analyze the company situation
Define problems and constraints
Define objectives
Define scope and boundaries
Database design
Create the conceptual design
DBMS software selection
Create the logical design
Create the physical design
Implementation and loading
Install the DBMS
Create the database(s)
Load or convert the data
Testing and evaluation
Test the database
Fine-tune the database
Evaluate the database and its application programs
Operation
Produce the required information flow
Maintenance and evaluation
Introduce changes
Make enhancements

Figure 2.1 Database Life Cycle

2.3 Project Schedule and Milestones

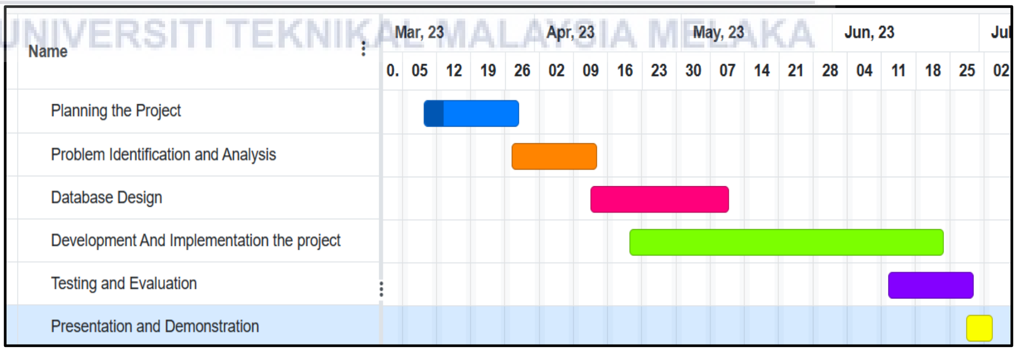


Figure 2.2 Gantt Chart of Project Task Management System

In Table 2.1, it shows the milestones of development of Project Task Management System. In each milestone, there are outcomes that are expected to be produced to ensure all processes are within the expected timeline and should be on time with relevant.

Table 2.1 Project Milestones

Milestone	**Expected Outcome**	**Date**
Problems identification and analysis	1. Problem statement, objective 2. Flow chart of the current system and the proposed system 3. Requirement and module of proposed system.	13/03/2023
Conceptual design of the proposed system	1. DFD and ERD of the proposed system 2. Business rules 3. Data Definition Language 4. Data Manipulation Language 5. Normalization and query	27/03/2023
Implementation of the proposed system	1. Database environment set up. 2. Graphical User Interface (GUI) of the proposed system 3. Source code of the proposed system each function 4. Create Function and trigger.	10/04/2023
Testing of the proposed system functionality	1. Test plan 2. Test case 3. Test result and analysis 4. Solution solving of error message from testing process	12/12/2023
Completed documentation.	1. PSM 1 final report 2. Project Demonstration	03/01/2024
Project Demonstration	Final Presentation	25/01/2024

2.4 Conclusion

In this chapter, the chosen methodology for system development is elucidated, providing insights into the adopted approach, the processes involved in each phase of the Database Life Cycle (DBLC), and the rationale behind selecting this methodology. The DBLC, consisting of six stages including database initial study, database design, implementation & loading, training and evaluation, and operations and maintenance, was chosen to guide the development of this project. Additionally, the maintenance phase will be ongoing to ensure the system's smooth operation. A more comprehensive technical analysis of the overall system will be discussed in the subsequent chapter.

CHAPTER 3: ANALYSIS

3.1 Introduction

In this chapter, the focus will be on the analysis phase of the project, which involves breaking down the entire system into sub-modules and gathering factual data to understand the processes involved. The objective is to identify any existing problems and provide feasible suggestions for enhancing the system. This includes studying the current recruitment process of the insurance company, gathering operational data, comprehending the flow of required data, identifying bottlenecks experienced by users, and proposing solutions to alleviate their workloads. The chapter will delve into understanding the current recruitment process, pinpointing weaknesses, and outlining improvements for the new system to meet user requirements.

3.2 Problem Analysis

The problem analysis reveals several key challenges faced by SE Infinity in effectively managing their projects and tenders. One major issue is the absence of a centralized system that can provide real-time monitoring of project milestones, track task completion, and identify potential bottlenecks. This lack of visibility hampers the company's ability to stay on top of project progress and make informed decisions in a timely manner. Without a clear overview of project status and potential obstacles, SE Infinity faces difficulties in effectively coordinating and managing their ongoing projects.

Another significant problem arises from the company's struggle to determine the appropriate staff members to assign to new projects or tenders. The absence of a comprehensive system that captures and organizes information about employee capabilities and expertise makes it challenging to identify the best-suited individuals for specific tasks. As a result, project assignments may be based on incomplete or outdated information, leading to suboptimal resource allocation and potential gaps in skill sets required for successful project execution.

3.3 The Purposed Improvement and Solution

These issues ultimately impede SE Infinity's efficiency, productivity, and overall project success. A solution that addresses these challenges is crucial to streamline project management processes, enhance communication and collaboration among team members, and ensure optimal resource utilization. By implementing a project task management system, SE Infinity can gain a centralized platform that provides real-time project visibility, facilitates efficient task allocation, and promotes effective coordination among team members. Such a system would not only improve project outcomes but also contribute to the company's overall growth and success.

3.4 Requirement analysis of the to-be system

Requirement analysis will describe the system requirement for each user and the functional requirement.

3.4.1 Functional Requirement (Process Model)

i. User Registration and Authentication
The system should provide a user registration process where staff members can create accounts and authenticate themselves to access the system.

ii. Project Creation and Monitoring
Users should be able to create new projects within the system, define project details (such as start date, end date, milestones), and monitor the progress of each project through intuitive dashboards or reports.

iii. Task Management
The system should allow users to create and assign tasks to specific team members, set deadlines, track task completion, and provide notifications and reminders for pending tasks.

iv. Reporting and Analytics
The system should generate comprehensive reports and provide analytical insights on project progress, task completion rates, resource utilization, and other relevant metrics. This will facilitate informed decision-making and enable.

3.4.1.1 Data Flow Diagram (DFD)

i. Context Diagram

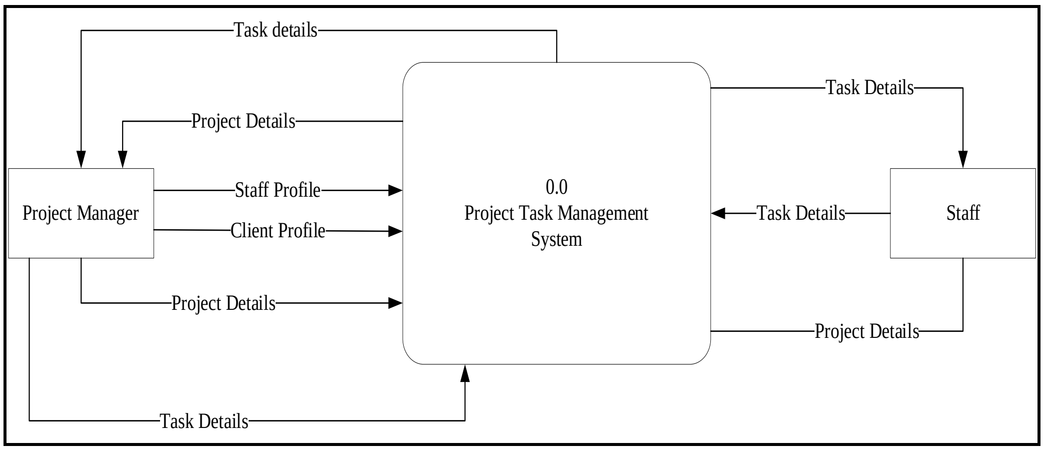


Figure 3.1 Context Diagram

ii. Data Flow Diagram (DFD) Level 1

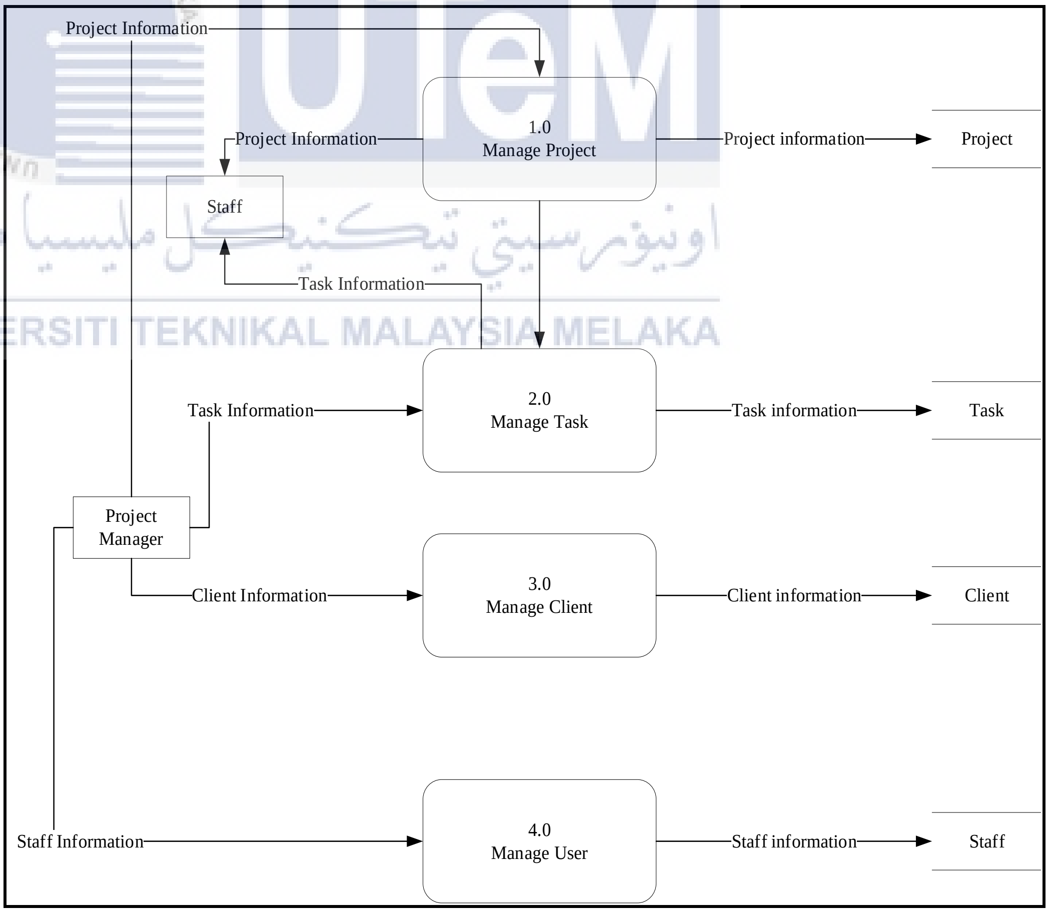


Figure 3.2 DFD Level 1

Figure 3.4 shows a more detailed breakout of pieces of the Context Level Diagram. It highlights the main functions carried out by the system, such as breakdown the high-level process of the Context Diagram into its sub processes.

iii. Data Flow Diagram (DFD) Level 2

DFD Level 2 offers a more detailed look at the processes that make up an information system than a level 1 DFD. It can be used to plan or record the specific makeup of a system.

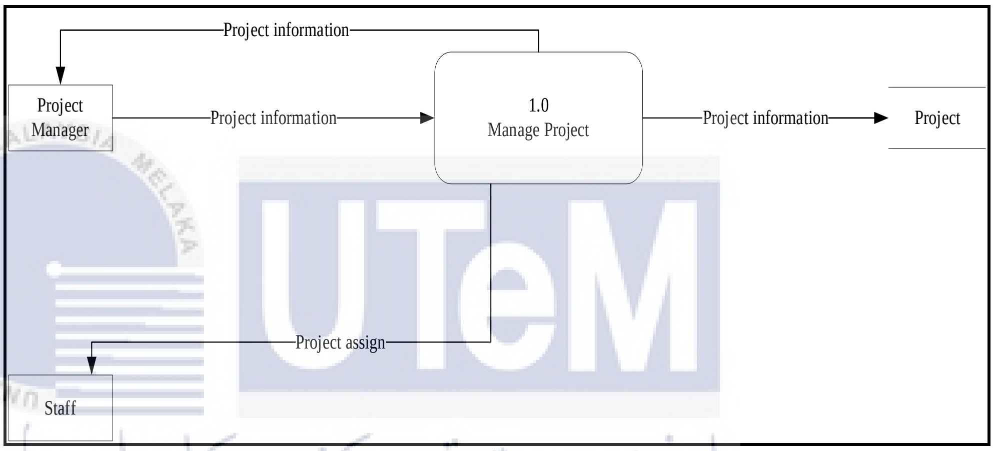


Figure 3.3 DFD Level 2 (Manage Project)

Figure 3-3 show manage project. Project managers are able to manage projects which can do a Create, Read, Update, Delete (CRUD) and staff can view the project that assign from task project

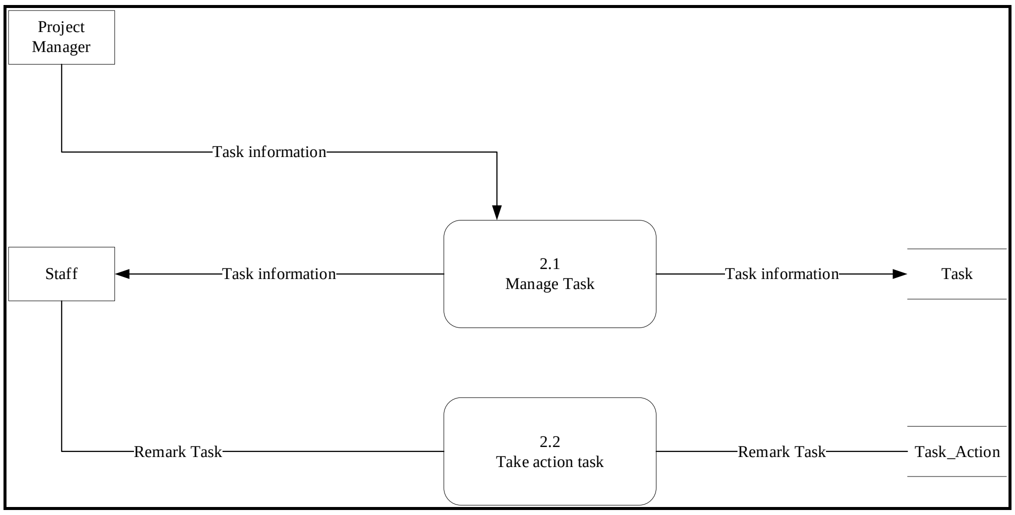


Figure 3.4 DFD Level 2 (Manage Task)

Figure 3-4 show manage task. Project manager will do CRUD task under project and assign to staff. Staff able to view the task that assign from project manager and staff can take action remark to the task.

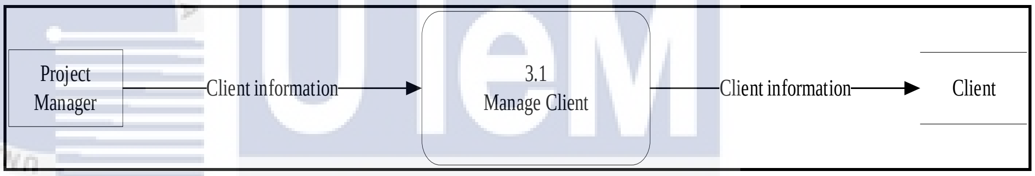


Figure 3.5 DFD Level 2 (Manage Client)

Figure 3-5 show managing client. The client profile will be added by the project manager in order to create a new project.

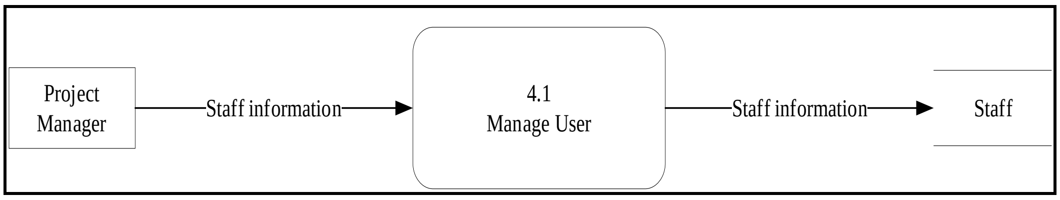


Figure 3.6 DFD Level 2 (Manage User)

3.4.2 Non-functional Requirement

Non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. Table 3.1 shows the non-functional requirements and its description for Project Task Management System.

Table 3.1 Non-Functional Requirement

No.	Non-Functional	Description
1	Performance	The system should be highly responsive and capable of handling multiple concurrent users and large data volumes without significant delays. It should provide quick response times for accessing and updating project information.
2	Scalability	The system should be scalable to accommodate future growth and increased project demands. It should be able to handle an expanding user base, growing project portfolios, and additional functionalities without compromising performance.
3	Usability	The system should have a user-friendly interface that is intuitive and easy to navigate. It should require minimal training for users to understand and operate the system effectively. Clear instructions, tooltips, and error messages should be provided to guide users throughout their interactions.
4	Reliability	The system should always be reliable and available to ensure uninterrupted access and data availability. It should have appropriate backup mechanisms and disaster recovery plans in place to mitigate the risk of data loss or system downtime.
5	Security	The system should have robust security measures to protect sensitive project information. It should employ encryption techniques for data transmission and storage, implement strong authentication and access control mechanisms, and regularly update security patches to address potential vulnerabilities.

3.4.3 Other Requirements

There are listed the requirements and specifications of software components which have been used in the system and such a software requirement and hardware requirements shown in Table 3.2 and Table 3.3 respectively.

Table 3.2 Software Requirement

Software	**Description**
Microsoft Word 2020	Microsoft Word is a word processor to write report.
Microsoft Visio 2019	Microsoft Visio 2019 is a software diagramming program for Microsoft Windows that uses vector graphics to create diagrams. These diagrams include flow chart, context diagram, data flow diagram (DFD) and entity relationship diagram (ERD)
MySQL (My Structured Query Language)	MySQL is an open-source relational database management system. Its scope includes data query and update, scheme creation and modification, and data access control.
PHP	PHP is a general-purpose programming language originally designed for web development.
Sublime Text 5	Sublime Text 5 is a source code editor to design the interface and implementation coding.

Software	Description
XAMP	XAMPP stands for Cross-Platform (X), Apache (A), MySQL (M), PHP (P), and Perl (P). It's designed to simplify the process of setting up a local web development environment. Once installed, you can start and stop the Apache web server and MySQL database server with a simple click. XAMPP includes PHP and Perl interpreters, making it a convenient package for developing and testing dynamic web applications on your local machine before deploying them to a live server.
Heidi SQL	HeidiSQL is a user-friendly database management tool that simplifies the interaction with MySQL, PostgreSQL, and Microsoft SQL databases. Connecting to a database is straightforward, and once connected, users can seamlessly view and manage databases and tables. The intuitive interface allows for the execution of SQL queries, browsing and editing data, and modifying table structures.

Table 3.3 Hardware Requirement

No	Hardware	Description
1	Processor	Intel Core i5
2	Memory	8 GB DDR3 L
3	Hard Disk	500 GB
4	Drive	DVD-ROM Drive

3.5 Conclusion

The current and proposed systems are visually represented through flow charts, depicting the process flow. The Context Diagram provides an overview of the entire system to be developed, while the Data-Flow Diagram illustrates the flow of data within the system or process. These diagrams also highlight the inputs, outputs, entities, and processes involved. Chapter III concludes by summarizing the analysis of the existing system, identifying areas for improvement or solutions, and presenting the conceptual framework for the new system.

CHAPTER 4: DESIGN

4.1 Introduction

During the design phase, the focus is on establishing the architecture of the system. The primary goal is to create a design that effectively fulfills the application requirements agreed upon. The requirements identified during the earlier analysis phase are refined and expanded to encompass all the specified functions of the application. This phase involves translating the conceptual design into a detailed technical blueprint that guides the development process. By carefully crafting the design, the system's structure, components, and interactions are defined to ensure a comprehensive and robust solution aligned with the project's objectives.

4.2 System Architecture Design

The application is structured into three tiers: the presentation tier, logical tier, and data tier. At the topmost level is the presentation tier, responsible for the user interface that communicates tasks and results to the user. The logical tier handles application processes, executing commands, making logical decisions, performing evaluations, and calculations. It also facilitates the movement and processing of data between the surrounding layers. The data tier stores and retrieves data from a database or file system. The information is then passed back to the logical tier for further processing before being presented back to the user. This tiered architecture ensures clear separation of responsibilities and promotes efficient data flow and processing within the application.

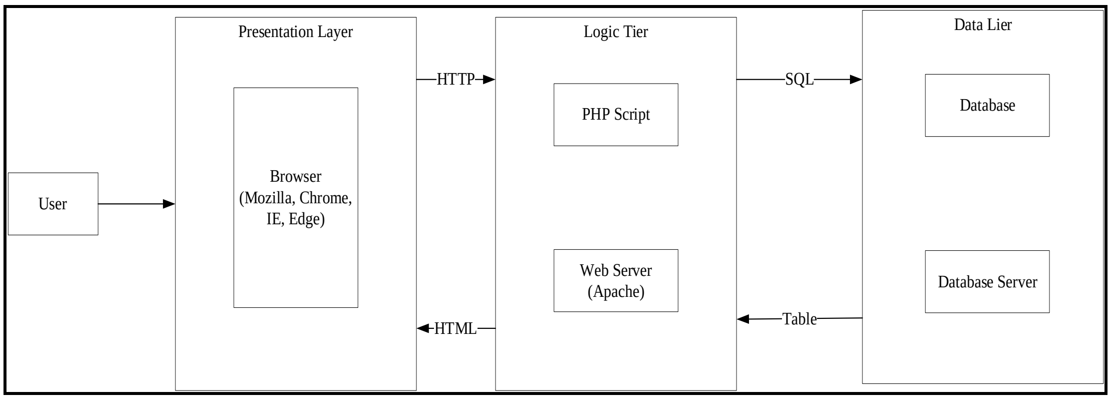


Figure 4.1 3 Tier Architecture

4.3 Database Design

Database design involves the meticulous creation of a comprehensive data model for a database. This data model encompasses the logical and physical design choices and storage parameters necessary to generate a design using a data definition language, which can then be utilized to build a database. A fully attributed data model provides detailed attributes for each entity, capturing the essential characteristics of the data.

The term "database design" can encompass various aspects of designing an entire database system. Primarily, it refers to the logical design of the fundamental data structures employed for data storage. In the context of the relational model, these structures consist of tables and views. On the other hand, in an object database, entities and relationships directly map to object classes and named relationships. By undertaking a well-structured database design process, organizations can establish an efficient and effective foundation for storing and managing their data.

4.3.1 Conceptual Design

Figure 4.2 shows the Entity Relationship Diagram (ERD) of Project Task Management System.

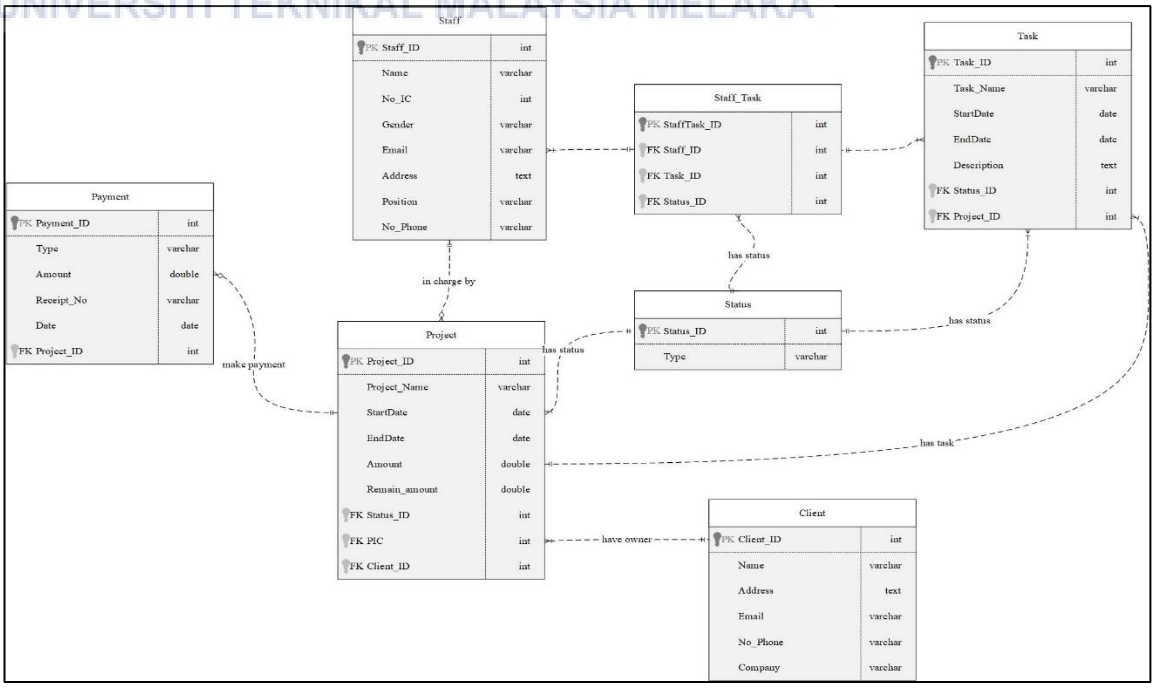


Figure 4.2 ERD of Project Task Management System

Business rules for ERD

1. Each staff member must be assigned at least one task.
2. A staff member can be assigned to multiple tasks simultaneously.
3. Every task must belong to a specific project.
4. A single task cannot belong to multiple projects.
5. Projects can have multiple tasks associated with them.
6. Only staff assigned to the project can be in charge of tasks within that project.
7. Each project may have one or more payments associated with it.
8. Each payment must be linked to a specific project it relates to.
9. Every project must have a single, identified client associated with it.
10. A client can have multiple projects associated with them.

4.3.2 Logical Design

The logical design phase is a crucial step in database design, following the initial conceptual design. During this phase, the relationships between local logical data are established, ensuring they are properly normalized and validated to align with user transactions. Additionally, constraints for each local logical data are thoroughly reviewed to ensure data integrity and consistency. By carefully defining and validating these relationships and constraints, the logical design phase lays the foundation for an efficient and robust database structure.

4.3.2.1 Data Dictionary for Entity Relationship Diagram

i. Staff Table

Table 4.1 Table Staff

Column	Type	Null	PK/FK	Reference Table
Staff_ID	int (10)	No	PK	
Name	varchar (255)	No		

Column	Type	Null	PK/FK	Reference Table
No_IC	varchar (30)	No		
Gender	varchar (50)	No		
Email	varchar (100)	No		
Address	text	No		
Position	varchar (100)	No		
No_Phone	varchar (100)	No		

ii. Project Table

Table 4.2 Table Project

Column	Type	Null	PK/FK	Reference Table
Project_ID	int (10)	No	PK	
Project_Name	varchar (100)	No		
StartDate	date	No		
EndDate	date	No		
Remain_amount	double	Yes		
Amount	double	No		
Status_ID	int (10)	No	FK	Status
PIC	int (10)	No	FK	Staff
Client_ID	int (10)	No	FK	Client

iii. Task Table

Table 4.3 Table Task

Column	Type	Null	PK/FK	Reference Table
Task_ID	int (10)	No	PK	
Task_Name	varchar (100)	No		
StartDate	date	No		
EndDate	date	No		
Desc	text	Yes		
Status_ID	int (10)	No	FK	Status
Project_ID	int (10)	Yes	FK	Project

iv. Client Table

Table 4.4 Table Client

Column	Type	Null	PK/FK	Reference Table
Client_ID	int (10)	No	PK	
Name	varchar (100)	No		
Address	text	No		
Email	varchar (100)	No		
No_Phone	varchar (100)	No		
Company	varchar (100)	Yes		

v. Payment Table

Table 4.5 Table Payment

Column	Type	Null	PK/FK	Reference Table
Payment_ID	int (10)	No	PK	
Type	varchar (100)	No		
Amount	double	No		
Date	date	No		
Receipt_No	varchar (100)	No		
Project_ID	int (10)	No	FK	Project

vi. Status Table

Table 4.6 Table Status

Column	Type	Null	PK/FK	Reference Table
Status_ID	int (10)	No	PK	
Type	varchar (100)	No		

vii. StaffTask Table

Table 4.7 Table StaffTable

Column	Type	Null	PK/FK	Reference Table
StaffTask_ID	Int(10)	No	PK	
Staff_ID	Int (10)	No	FK	Staff
Task_ID	Int (10)	No	FK	Task
Status_ID	Int (10)	No	FK	Status

4.3.2.2 Normalization

The conceptual design uses normalization displayed. Each table shows attributes, primary key and foreign key.

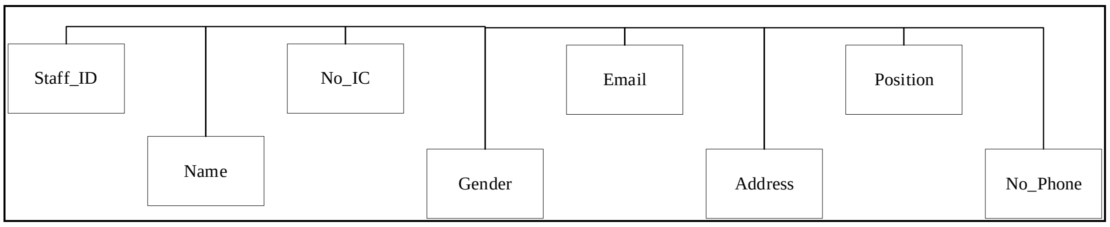


Figure 4.3 Staff Table 3NF

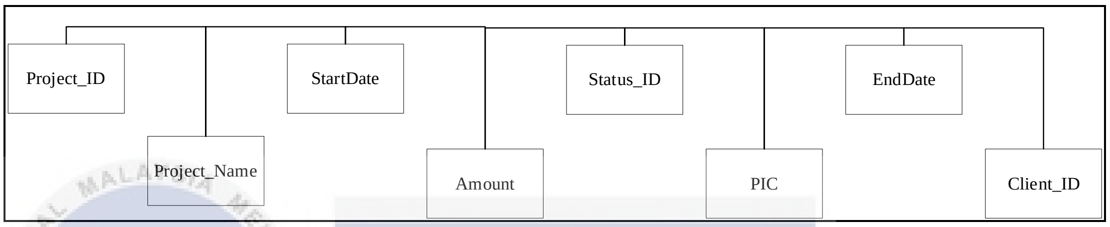


Figure 4.4 Project Table 3NF

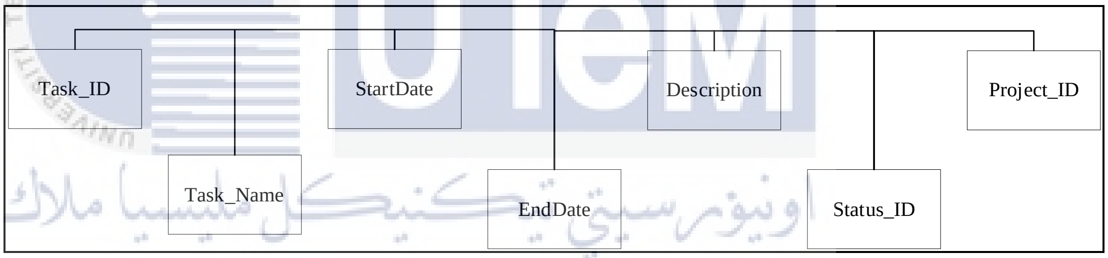


Figure 4.5 Task Table 3NF

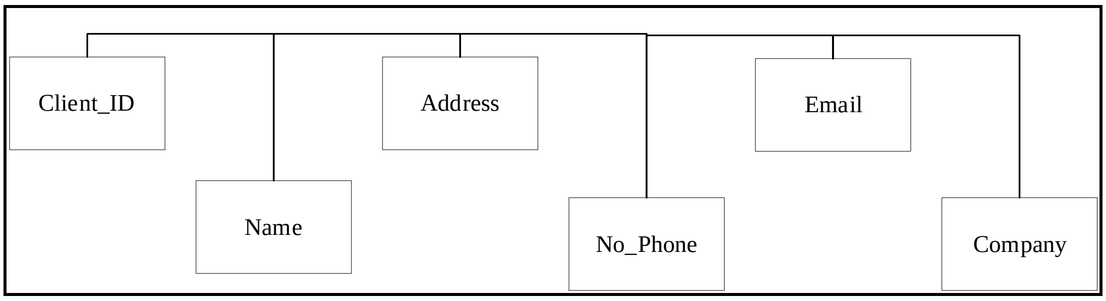


Figure 4.6 Client Table 3NF

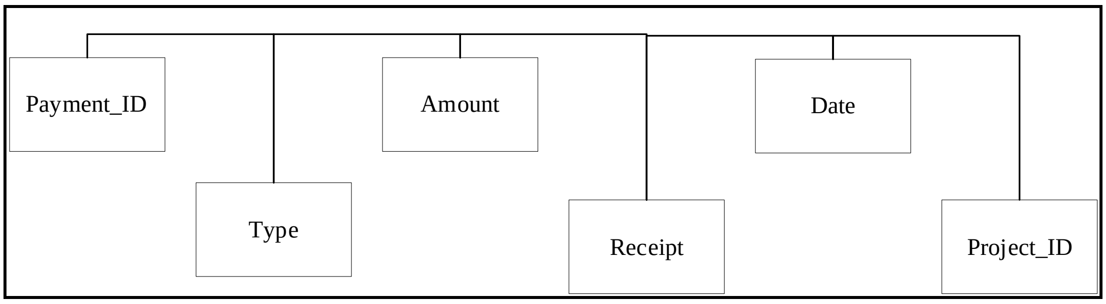


Figure 4.7 Payment Table 3NF

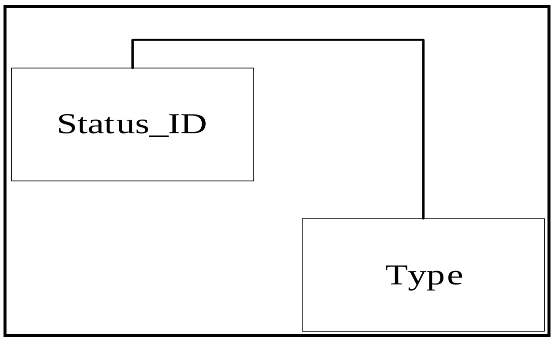


Figure 4.8 Status Table 3NF

4.3.2.3 Query Design

There are many query designs that can be carried out to produce many different types output. Each query has its own requirement, reason and purpose.

Table 4.8 Query Design

Type of Query	Query	Explanation
Simple Query	`SELECT Client_ID, Company FROM CLIENT`	To retrieve all client
Join Table Query	`SELECT p.Project_ID, p.Project_Name, DATE_FORMAT(p.StartDate, '%d %M %Y'), DATE_FORMAT(p.EndDate, '%d %M %Y'), s.Type, f.Name FROM project p JOIN status s ON s.Status_ID = p.Status_ID JOIN staff f ON f.Staff_ID = p.PIC ORDER BY p.Project_Name ASC`	To display all project detail and join with related table such status and staff table.
	`SELECT c.* FROM client c JOIN project p ON p.Client_ID = c.Client_ID WHERE p.Project_ID = {$projectID}`	Retrieve all data client that has tender with project.

	SELECT t.Task_ID, t.Task_Name, DATE_FORMAT(t.StartDate, '%d %M %Y'), DATE_FORMAT(t.EndDate, '%d %M %Y'), s.Name, x.\`Type\` FROM task t JOIN staff s ON s.Staff_ID = t.Staff_ID JOIN status x ON x.Status_ID = t.Status_ID WHERE Project_ID = {$projectID}	Retrieve all task that related to project table.
Aggregate and Grouping Query	SELECT t.Status_ID, s.Type, COUNT(t.Status_ID), p.Project_Name FROM task t JOIN project p ON p.Project_ID = t.Project_ID JOIN status s ON s.Status_ID = t.Status_ID GROUP by t.Status_ID, s.Type, p.Project_Name;	Display the number of task status for each project.
	SELECT t.Status_ID, s.Type, COUNT(t.Status_ID), p.Project_Name FROM task t JOIN project p ON p.Project_ID = t.Project_ID JOIN status s ON s.Status_ID = t.Status_ID WHERE t.Status_ID = 5 GROUP by t.Status_ID, s.Type, p.Project_Name;	Display tasks complete by project
	SELECT p.Project_ID, p.Project_Name, COUNT(t.Task_ID) AS total FROM task t JOIN project p ON p.Project_ID = t.Project_ID GROUP by p.Project_Name, p.Project_ID;	Display total task by project

4.3.2.4 Usage Function and Trigger

i. Function

Table 4.9 Function MySQL

Function Name	Query	Explanation

INITCAP ()	`DELIMITER $$`
	`CREATE`
	`` DEFINER=`root`@`localhost` ``
	`` FUNCTION `INITCAP`(`str` ``
	`VARCHAR(8000)) RETURNS`
	`varchar(8000) CHARSET utf8`
	`COLLATE utf8_general_ci`
	`BEGIN`
	`         DECLARE len`
	`INT DEFAULT LENGTH(str);`
	`         DECLARE pos`
	`INT DEFAULT 1;`
	`         DECLARE ch`
	`CHAR(1);`
	`         DECLARE ch_ascii`
	`INT;`
	`         DECLARE out_str`
	`VARCHAR(8000) DEFAULT '';`
	`         DECLARE`
	`prev_alphanum INT DEFAULT 0;`
	`         WHILE pos <= len`
	`         DO`
	`           SET ch =`
	`SUBSTRING(str, pos, 1);`
	`           SET ch_ascii =`
	`ASCII(ch);`
	`           IF prev_alphanum`
	`= 1 THEN`
	`             SET out_str =`
	`CONCAT(RPAD(out_str, pos -`
	`1,' '), LOWER(ch));  -- RPAD`
	`is required to append ' '`
	`           ELSE`
	`             SET out_str =`
	`CONCAT(RPAD(out_str, pos -`
	`1,' '), UPPER(ch));`
	`           END IF;`
	`           IF ch_ascii <=`
	`47 OR (ch_ascii BETWEEN 58`
	`AND 64) OR`
	`             (ch_ascii`
	`BETWEEN 91 AND 96) OR`
	`(ch_ascii BETWEEN 123 AND`
	`126) THEN`
	`             SET`
	`prev_alphanum = 0;`
	`           ELSE`
		replace the first letter of every word in a character string with an uppercase letter.

	`SET prev_alphanum = 1;`
	`END IF;`
	`SET pos = pos + 1;`
	`END WHILE;`
	`RETURN out_str;`
	`END$$`
	`DELIMITER ;`	

CompareDate	`DELIMITER $$`
	`CREATE DEFINER=`root`@`localhost` FUNCTION `CompareDates`(`startDate` DATE, `endDate` DATE) RETURNS int(11)`
	`BEGIN`
	`    DECLARE result INT;`
	`    IF endDate <= startDate THEN`
	`        SET result = true; -- endDate is less than to startDate`
	`    ELSE`
	`        SET result = false; -- endDate is greater than startDate`
	`    END IF;`
	`    RETURN result;`
	`END$$`
	`DELIMITER ;`
		Function will compare that from argument.

ii. Trigger

Table 4.10 Function MySQL

Trigger Name	**Query**	**Explanation**

Calculate_Remain	`CREATE TRIGGER `calculate_remain` AFTER INSERT ON `payment``
	 `FOR EACH ROW BEGIN`
	 `DECLARE remain DOUBLE;`
	 `DECLARE amountP DOUBLE;`
	 `DECLARE total DOUBLE;`
	 `-- Calculate Sum Amount`
	 `SELECT SUM(Amount) INTO remain FROM Payment WHERE Project_ID = NEW.Project_ID;`
	 `SELECT Amount INTO amountP FROM Project WHERE Project_ID = NEW.Project_ID;`
	 `SET total = remain - amountP;`
	 `UPDATE Project SET Remain_amount = total WHERE Project_ID = NEW.Project_ID;`
	 `END`
		The trigger will calculate the amount from payment table using SUM function and select the amount from project table. Then, it will be a minus between amount project and sum of amount payment. After that, it will update the project table column remain_amount.

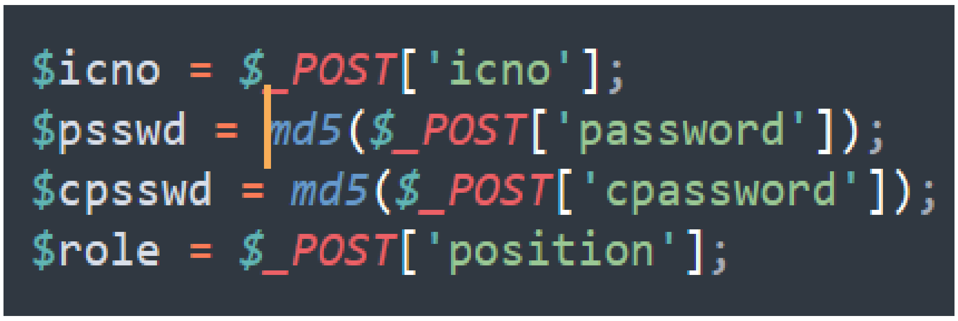
Project_Del_Status	`CREATE TRIGGER `Project_Del_Status` AFTER DELETE ON `task` FOR EACH ROW BEGIN DECLARE total_tasks INT; SET total_tasks = (SELECT COUNT(*) FROM Task WHERE project_id = OLD.project_id); IF total_tasks = 0 THEN UPDATE Project SET status_id = 1 WHERE project_id = OLD.project_id; ELSE UPDATE Project SET status_id = 2 WHERE project_id = OLD.project_id; END IF; END`	Will update status if there any deletion from task table.
Check_duedate_insert	`CREATE TRIGGER `check_duedate_insert` BEFORE INSERT ON `project` FOR EACH ROW BEGIN IF CompareDates(NEW.StartDate, NEW.EndDate) THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Due date must be in the future.'; END IF; END`	The trigger will perform when before insert into project table it will check a due date and start date by calling function CompareDate

update_project_status	`CREATE TRIGGER`
	`` `Project_Update_Status` ``
	`` AFTER UPDATE ON `task` ``
	` FOR EACH ROW BEGIN`
	`  DECLARE total_tasks INT;`
	`  DECLARE completed_tasks INT;`
	`  SET total_tasks = (SELECT COUNT(*) FROM Task WHERE project_id = NEW.project_id);`
	`  SET completed_tasks = (SELECT COUNT(*) FROM Task WHERE project_id = NEW.project_id AND status_id = 5);`
	`  IF completed_tasks = total_tasks THEN`
	`    UPDATE Project SET status_id = 5 WHERE project_id = NEW.project_id;`
	`  ELSE`
	`    UPDATE Project SET status_id = 2 WHERE project_id = NEW.project_id;`
	`  END IF;`
	`END`
		The trigger will have triggered when update on task will be calculate the complete task if all task in project have been completed then its will be update status at project will be complete.

Project_Insert_status	`CREATE TRIGGER`
	`` `Project_Insert_Status` ``
	`` AFTER INSERT ON `task` ``
	` FOR EACH ROW BEGIN`
	`  UPDATE Project SET status_id = 2 WHERE project_id = NEW.project_id;`
	`END`
		The trigger will be triggered when insert task, it will update project status to "To Do" status.

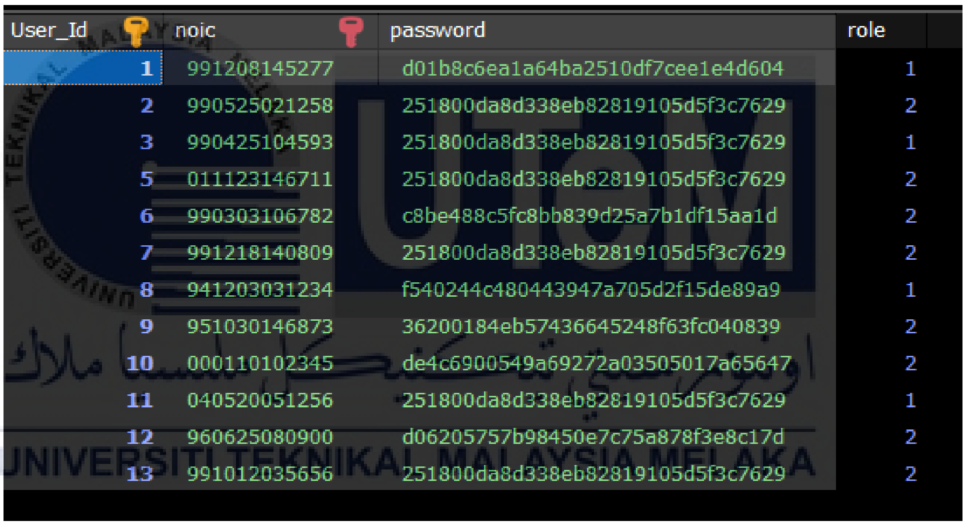
4.3.2.5 Security Mechanism

The security mechanism used in this system is encryption password before insertion into database table. The encryption that used in hash password is md5 function built-in in PHP.



```
$icno = $_POST['icno'];
$psswd = md5($_POST['password']);
$cpsswd = md5($_POST['cpassword']);
$role = $_POST['position'];
```

Figure 4.9 MD5 Encrypt Password



User_Id	noic	password	role
1	991208145277	d01b8c6ea1a64ba2510df7cee1e4d604	1
2	990525021258	251800da8d338eb82819105d5f3c7629	2
3	990425104593	251800da8d338eb82819105d5f3c7629	1
5	011123146711	251800da8d338eb82819105d5f3c7629	2
6	990303106782	c8be488c5fc8bb839d25a7b1df15aa1d	2
7	991218140809	251800da8d338eb82819105d5f3c7629	2
8	941203031234	f540244c480443947a705d2f15de89a9	1
9	951030146873	36200184eb57436645248f63fc040839	2
10	000110102345	de4c6900549a69272a03505017a65647	2
11	040520051256	251800da8d338eb82819105d5f3c7629	1
12	960625080900	d06205757b98450e7c75a878f3e8c17d	2
13	991012035656	251800da8d338eb82819105d5f3c7629	2

Figure 4.10 Password in Table

Figure 4-4 show the password in table. The password already hashed by using MD5 function built-in PHP. Any user on the database cannot view the actual password.

4.4 Graphical User Interface (GUI) Design

Graphical User Interface (GUI) design encompasses the process of determining how users will interact with a system and the specific inputs and outputs it accepts and produces. This design includes the creation of screen displays that facilitate system navigation, as well as the development of screens and forms that capture user data. GUI design can be divided into three essential components: navigation design, input design, and output design. Navigation design focuses on establishing an intuitive and user-friendly system flow, ensuring users can easily navigate through different screens and functionalities. Input design pertains to designing interfaces that enable users to input data into the system, ensuring data capture is efficient and user-friendly. Output design involves designing how the system presents information and outputs to the users, such as reports, alerts, or notifications. By carefully considering and crafting these three aspects, GUI design contributes to an enhanced user experience and effective system usability.

4.4.1 Navigation Design

The navigation component of the interface design provides straightforward access and clear direction for the user.

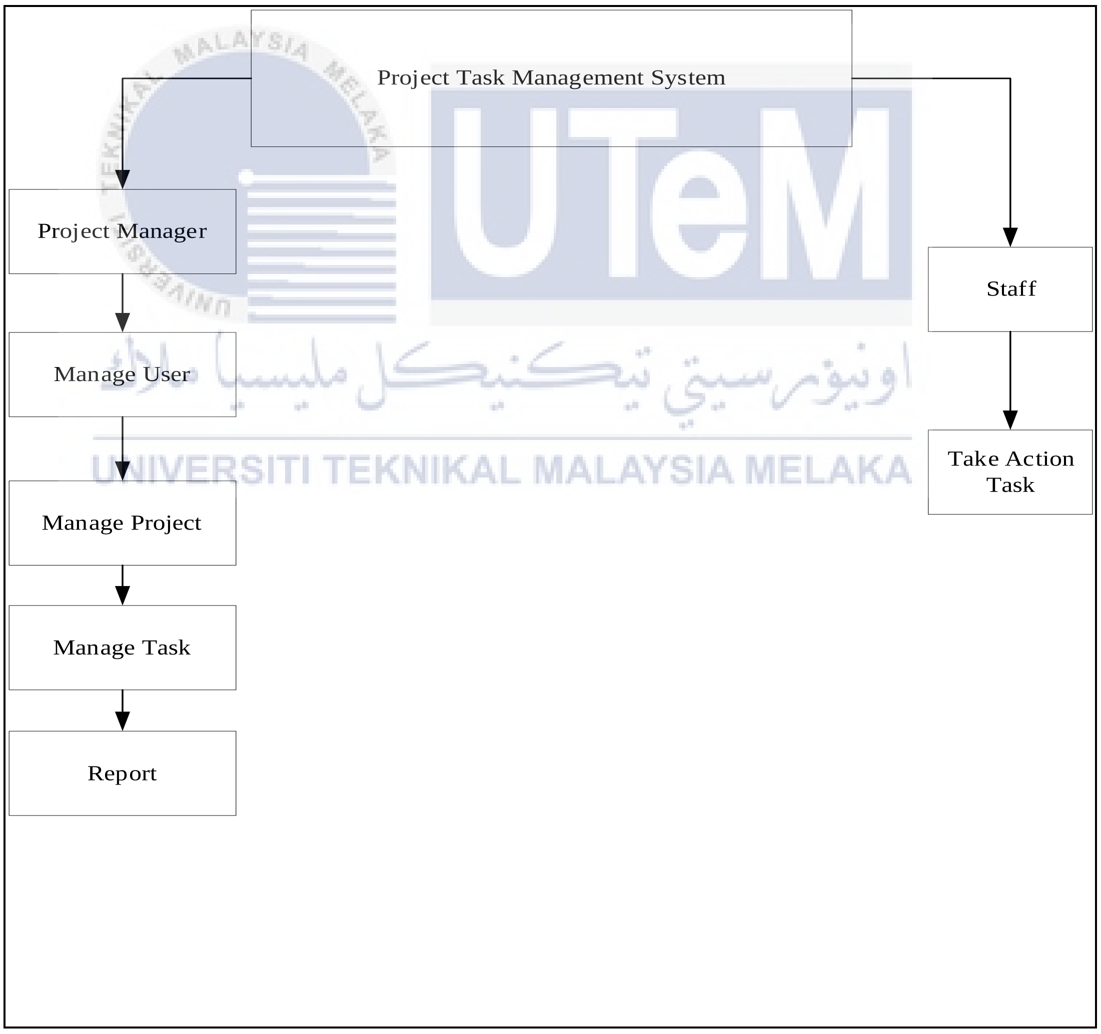


Figure 4.11 Navigation Path of Project Task Management System

4.4.2 Input Design

The input design focuses on the entry of data into the system by users in the form of structured and unstructured data. The screen and forms are designed and used to store information for the system for an action performed. Figure 4.6 until 4.18 illustrates the input design can be referred to in Appendix A.

4.4.3 Output Design

The output design concerns presenting retrieved information of the system on the screen or form. The output design was shown.

4.5 Conclusion

This chapter concludes by presenting a systematic approach to designing the Project Task Management System, ensuring that it fulfills the identified requirements and features derived from the analysis phase. The objective of the design phase is to address the problem outlined in the requirement document during the analysis phase and transition from the problem domain to the solution domain. The primary delivery of this phase is the design document, which serves as a blueprint for the subsequent phases of implementation, testing, and maintenance. By following a well-defined design process, the Project Task Management System can be developed in a structured manner, ensuring that it meets the specified requirements and facilitates efficient project and task management.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

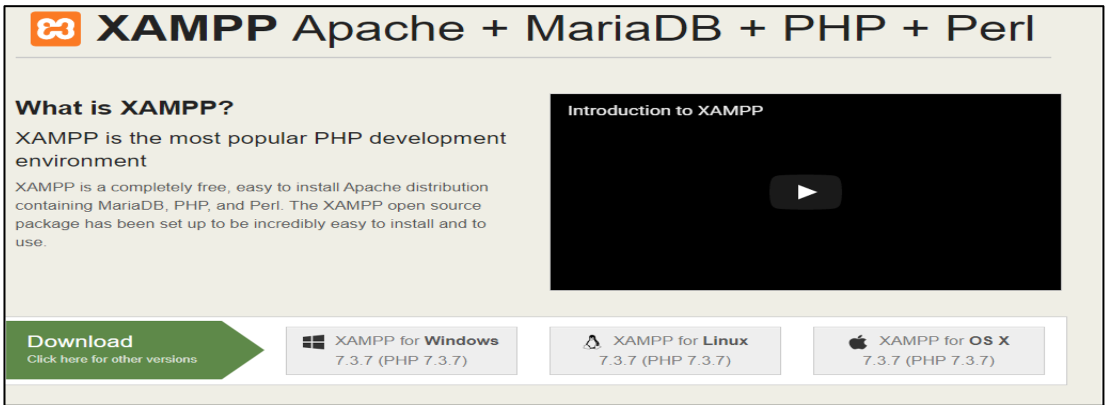
In this chapter, the process of implementing the database design is outlined. It includes the steps for database installation and configuration, specifically on the Windows 11 platform. MySQL is installed during this phase. Data Definition Language (DDL) and Data Manipulation Language (DML) operations, in the form of SQL statements, are executed within the database. The chapter also provides a detailed description of the implementation status for each module.

5.2 System Development Environment Setup

This setup involves configuring essential components for web application creation, including the Web Server with Apache, Database Server with MySQL, Web Programming Language with PHP, and Database Management Tool with HeidiSQL. For Windows platforms, users can conveniently obtain the required components through Apache Friends' free package XAMPP. Notably, the mentioned key software elements can be effortlessly installed within the XAMPP framework, streamlining the setup process for Project Task Management Task development.

5.2.1 Installation XAMPP Setup

Step 1: In web browser, go to https://www.apachefriends.org/index.html and download XAMPP for Windows.


XAMPP Apache + MariaDB + PHP + Perl

What is XAMPP?

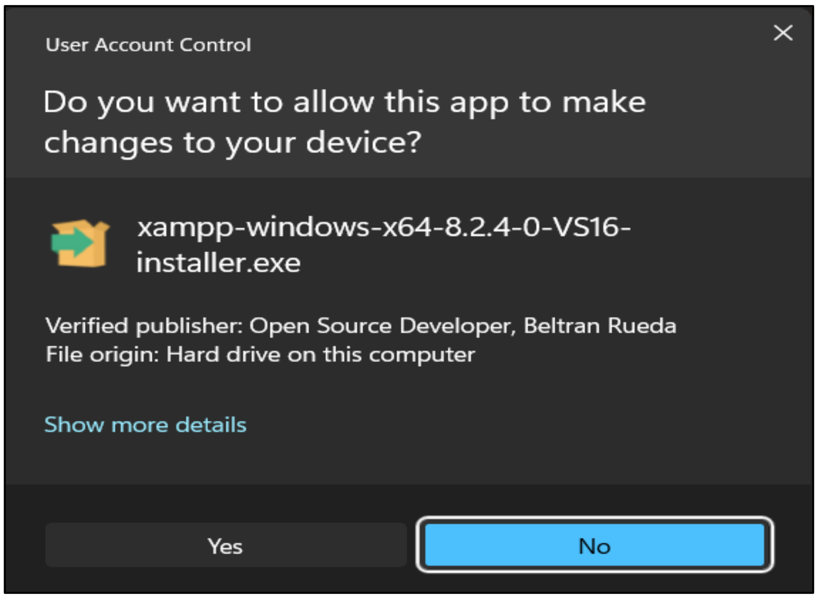
XAMPP is the most popular PHP development environment

XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use.

Introduction to XAMPP
Download
Click here for other versions
XAMPP for Windows 7.3.7 (PHP 7.3.7)
XAMPP for Linux 7.3.7 (PHP 7.3.7)
XAMPP for OS X 7.3.7 (PHP 7.3.7)

Figure 5.1 Download XAMPP for Windows

Step 2: Click the Yes button at XAMPP installer for installation in figure 5.2.



User Account Control

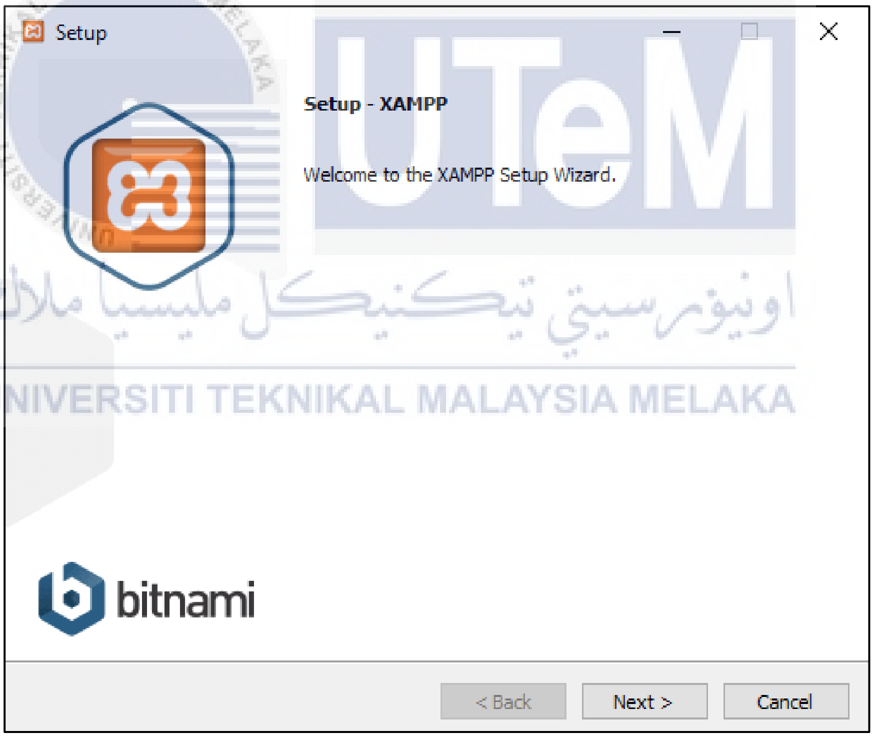
Do you want to allow this app to make changes to your device?

xampp-windows-x64-8.2.4-0-VS16-installer.exe

Verified publisher: Open Source Developer, Beltran Rueda
File origin: Hard drive on this computer

Figure 5.2 XAMPP Installer

Step 3: Click the Next button to setup XAMPP in figure 5.3.



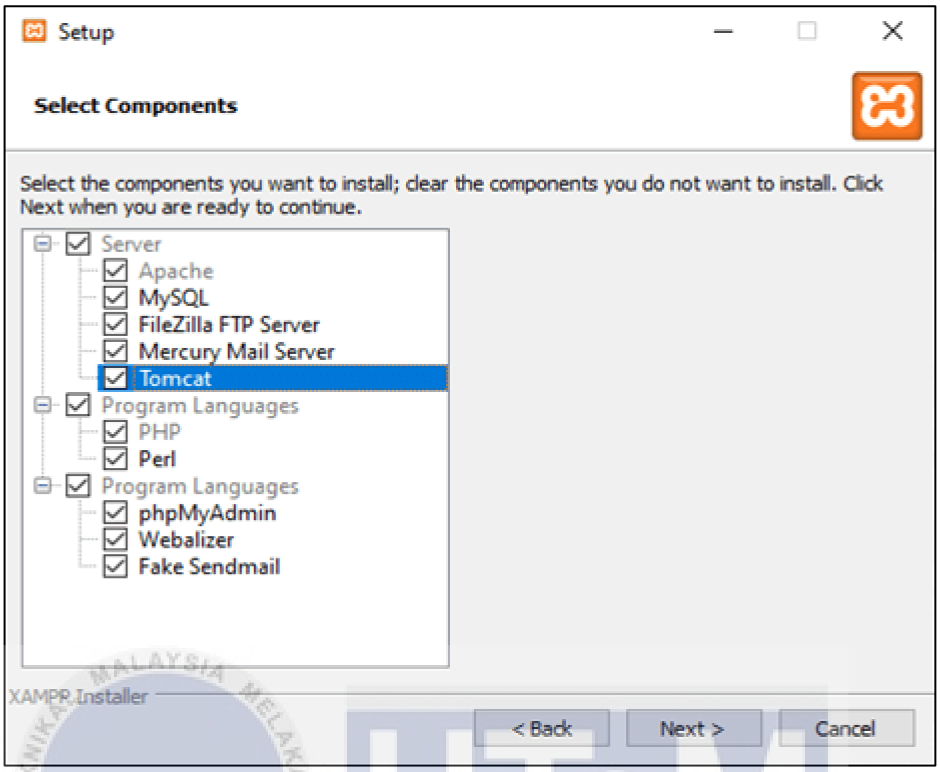
Setup

Setup - XAMPP

Welcome to the XAMPP Setup Wizard.

Figure 5.3 XAMPP Setup Wizard

Step 4: Click the Next button to continue the installation in figure 5.4.


Setup

Select Components

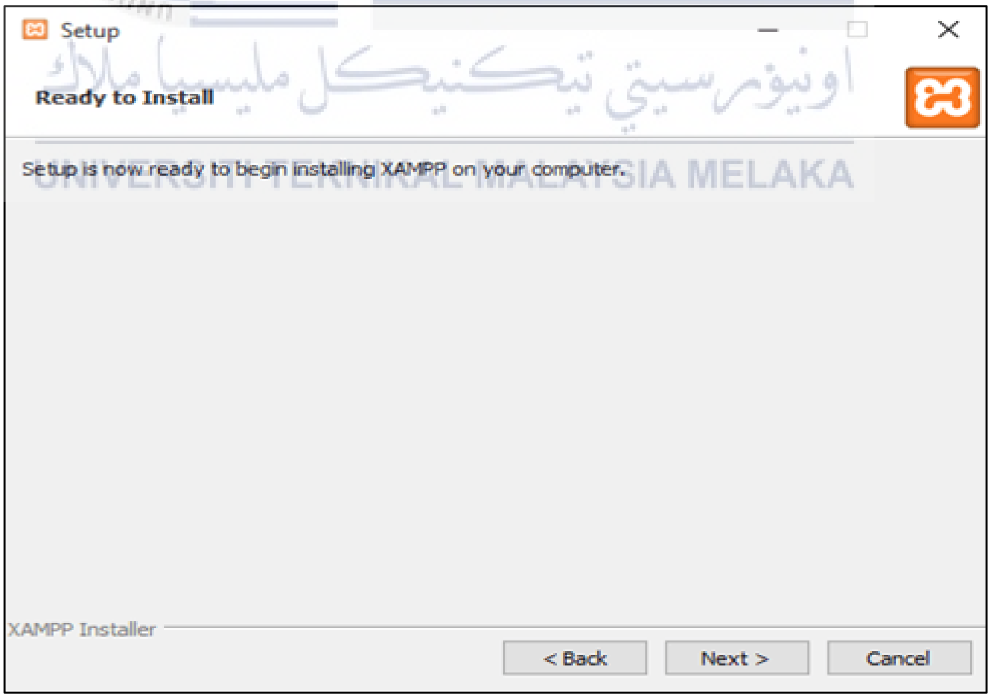
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

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

XAMPP Installer

Figure 5.4 Select Components

Step 5: Click the Next button again to continue the installation in figure 5.5.


Setup

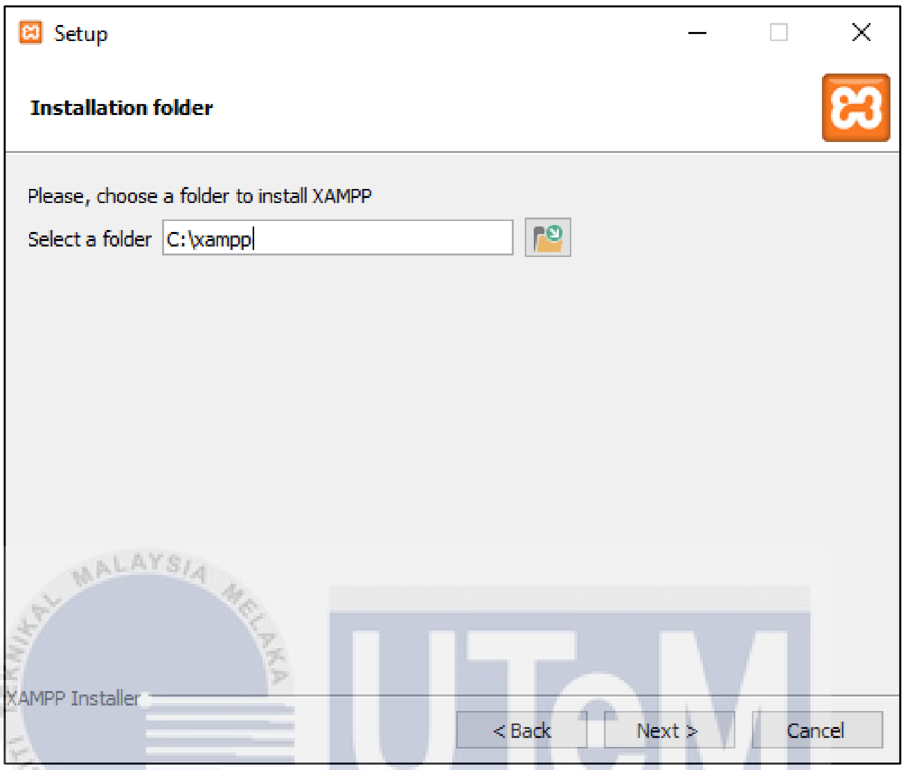
Ready to Install

Setup is now ready to begin installing XAMPP on your computer.

XAMPP Installer

Figure 5.5 Ready to Install

Step 6: Choose the location button in figure 5.6 to install XAMPP and click Next button to continue. Then, the XAMPP will start to install as shown in Figure 5.7



Setup

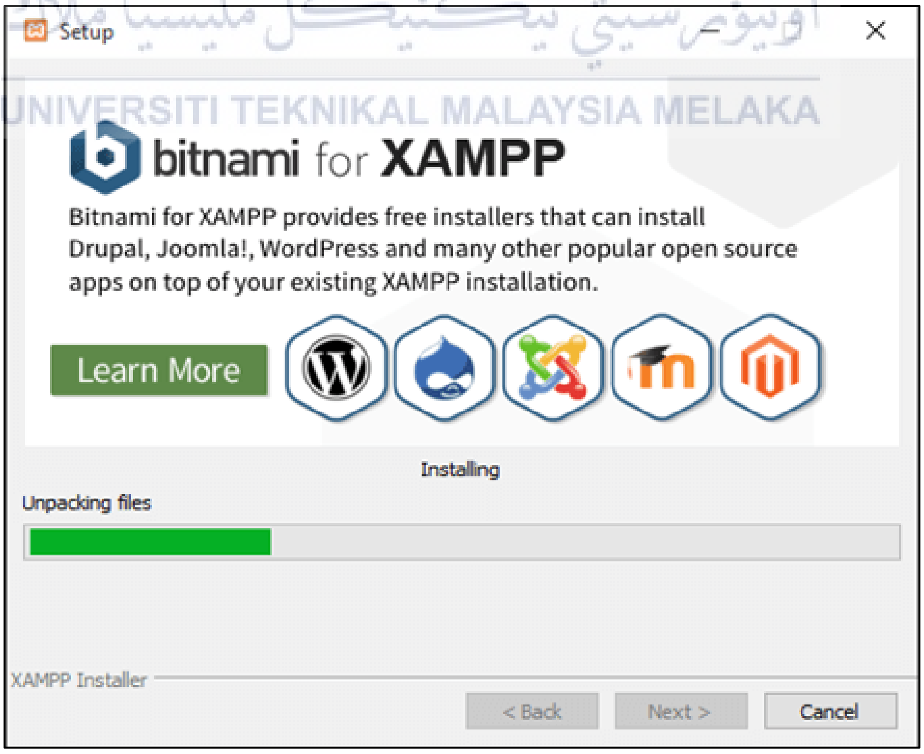
Installation folder

Please, choose a folder to install XAMPP

Select a folder C:\xampp

XAMPP Installer

Figure 5.6 Installation Path Folder



Setup

bitnami for XAMPP

Bitnami for XAMPP provides free installers that can install Drupal, Joomla!, WordPress and many other popular open source apps on top of your existing XAMPP installation.

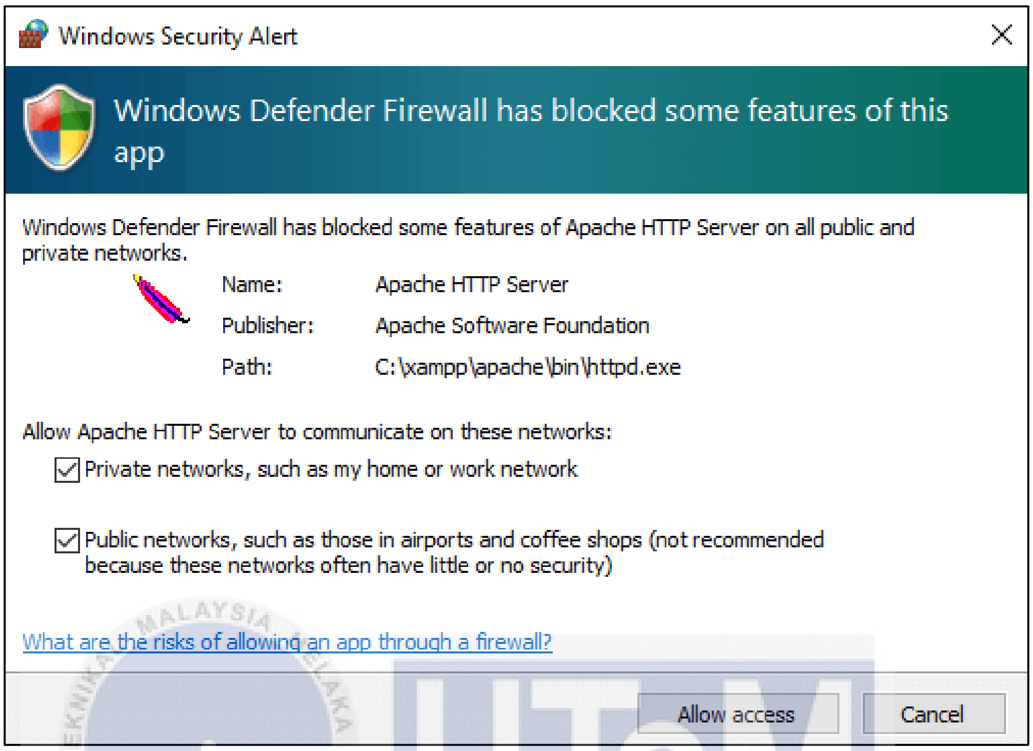
Installing

Unpacking files

XAMPP Installer

Figure 5.7 Setup Progress

Step 7: Click the Allow access button to disable the block of some features of the server from Windows Firewall in figure 5.8.



Windows Security Alert

Windows Defender Firewall has blocked some features of this app

Windows Defender Firewall has blocked some features of Apache HTTP Server on all public and private networks.

Name: Apache HTTP Server
Publisher: Apache Software Foundation
Path: C:\xampp\apache\bin\httpd.exe

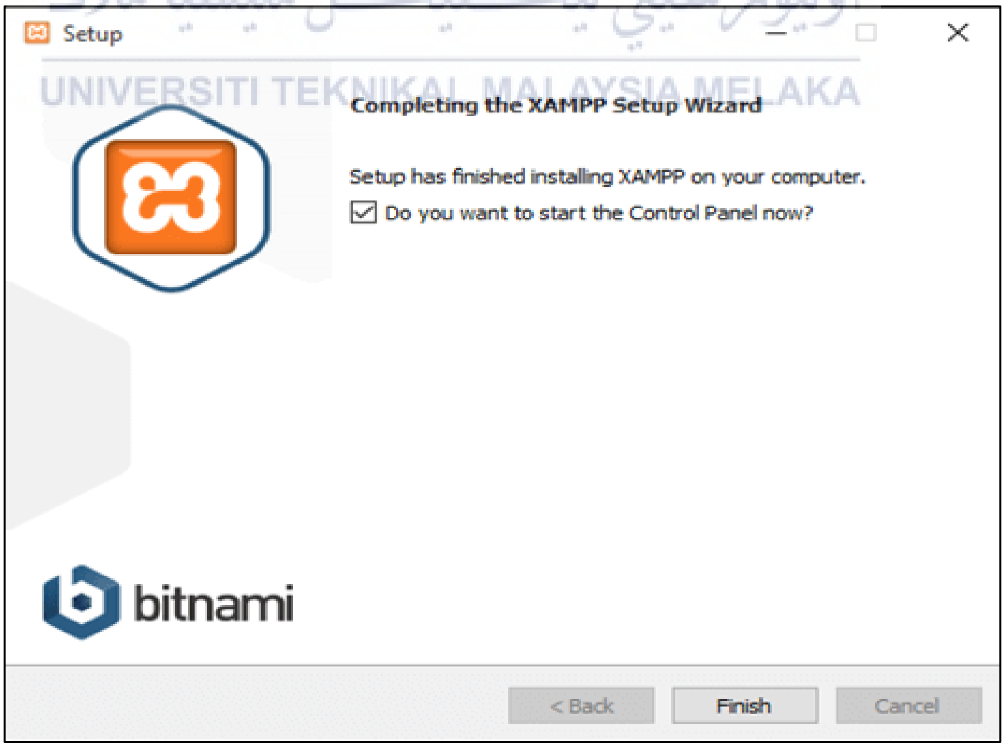
Allow Apache HTTP Server to communicate on these networks:

☑ Private networks, such as my home or work network

☑ Public networks, such as those in airports and coffee shops (not recommended because these networks often have little or no security)

Figure 5.8 Windows Firewall

Step 8: Figure 5.9 shows the XAMPP setup wizard is completing. Click the Finish button to start with XAMPP control panel.



Setup

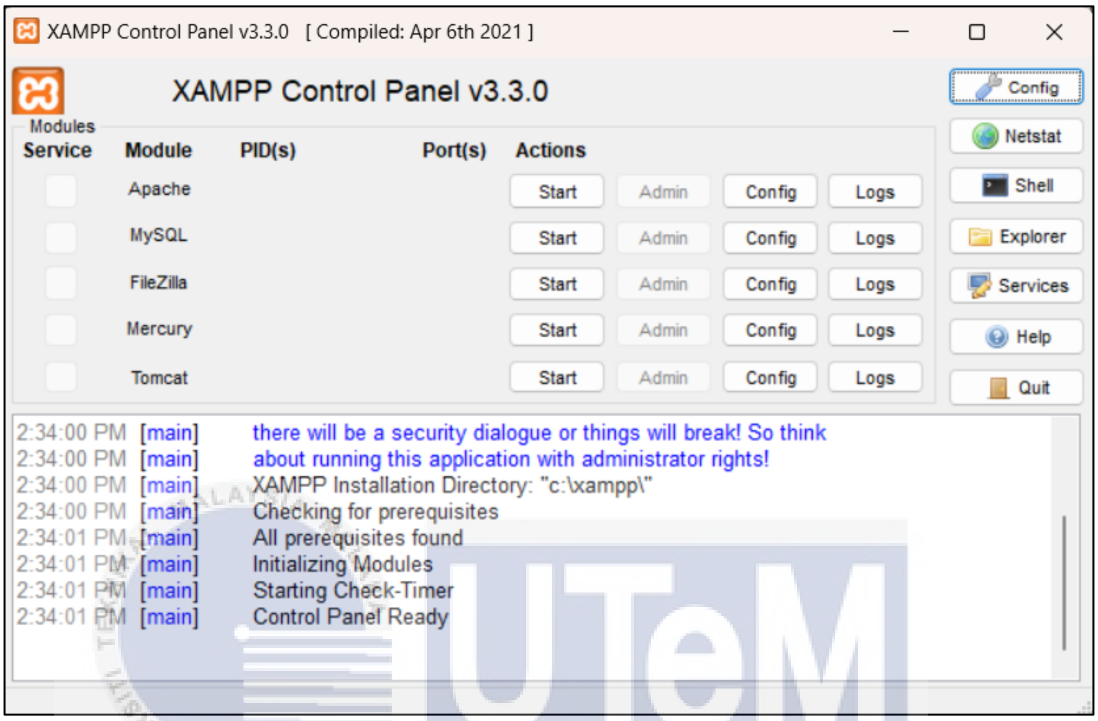
Completing the XAMPP Setup Wizard

Setup has finished installing XAMPP on your computer.

☑ Do you want to start the Control Panel now?

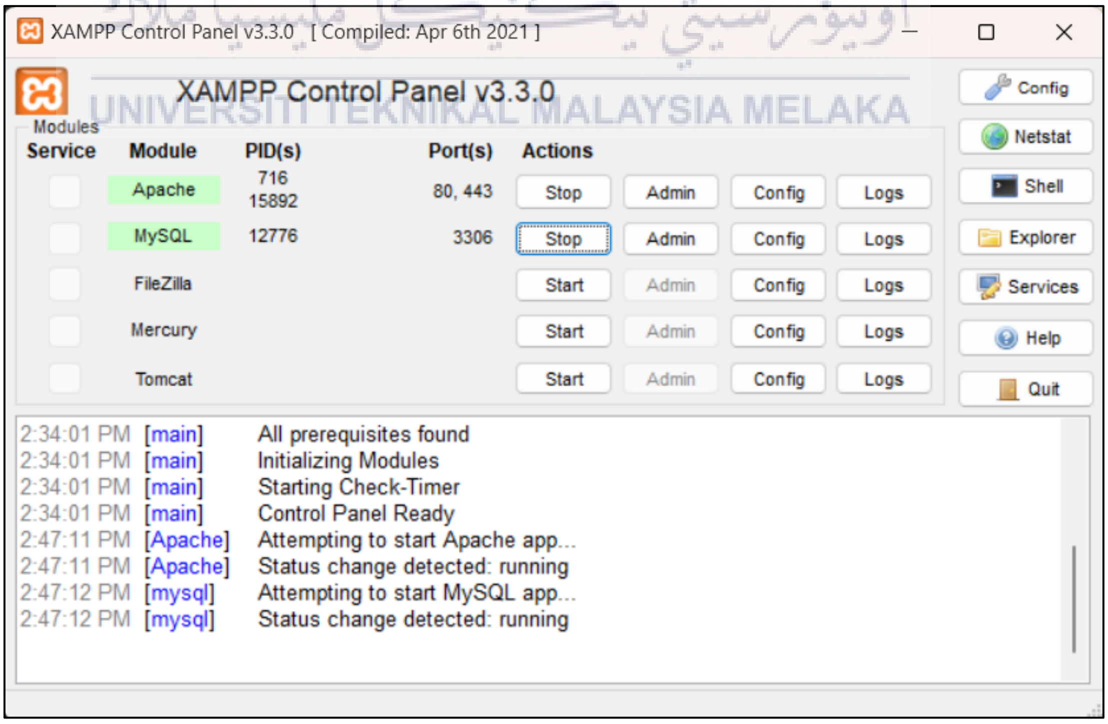
Figure 5.9 Completing Installation of XAMPP Setup Wizard

Step 9: Figure 5.10 shows the XAMPP control panel. Click the start button in Module Apache and MySQL to run the modules and wait until the modules show green color at the modules and this indicated that both modules run successfully as shown in figure 5.11.



```
2:34:00 PM  [main]   there will be a security dialogue or things will break! So think
2:34:00 PM  [main]   about running this application with administrator rights!
2:34:00 PM  [main]   XAMPP Installation Directory: "c:\xampp\"
2:34:00 PM  [main]   Checking for prerequisites
2:34:01 PM  [main]   All prerequisites found
2:34:01 PM  [main]   Initializing Modules
2:34:01 PM  [main]   Starting Check-Timer
2:34:01 PM  [main]   Control Panel Ready
```

Figure 5.10 XAMPP Contral Panel.



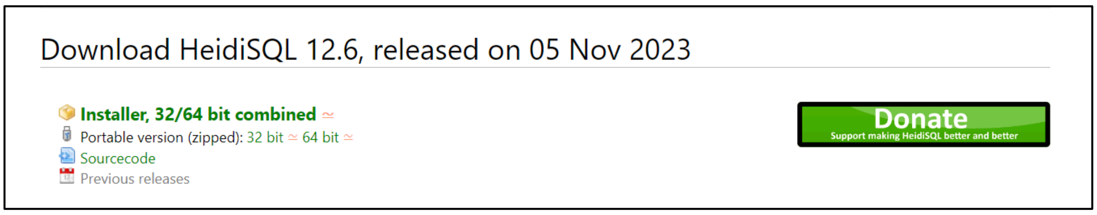
Module	PID(s)	Port(s)
Apache	716 15892	80, 443
MySQL	12776	3306
FileZilla		
Mercury		
Tomcat		

```
2:34:01 PM  [main]     All prerequisites found
2:34:01 PM  [main]     Initializing Modules
2:34:01 PM  [main]     Starting Check-Timer
2:34:01 PM  [main]     Control Panel Ready
2:47:11 PM  [Apache]   Attempting to start Apache app...
2:47:11 PM  [Apache]   Status change detected: running
2:47:12 PM  [mysql]    Attempting to start MySQL app...
2:47:12 PM  [mysql]    Status change detected: running
```

Figure 5.11 Apache and MySQL modules are running.

5.2.2 Installation of HeidiSQL

Step 1: In web browser, go to https://www.heidisql.com/download.php and click "installer, 32/64 bit combined" it will download.

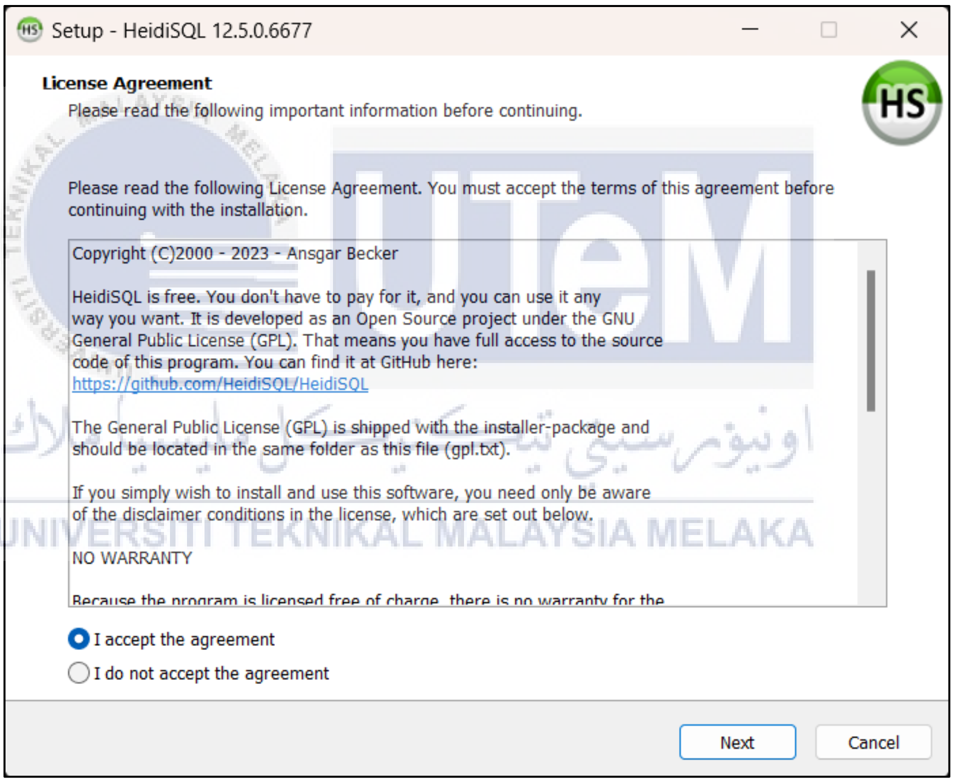


Download HeidiSQL 12.6, released on 05 Nov 2023

Installer, 32/64 bit combined
Portable version (zipped): 32 bit 64 bit
Sourcecode
Previous releases

Figure 5.12 Download HeidiSQL

Step 2: Choose "I accept the agreement" radio button and click next in figure 5.13.



Setup - HeidiSQL 12.5.0.6677

License Agreement
Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

Copyright (C)2000 - 2023 - Ansgar Becker

HeidiSQL is free. You don't have to pay for it, and you can use it any way you want. It is developed as an Open Source project under the GNU General Public License (GPL). That means you have full access to the source code of this program. You can find it at GitHub here: https://github.com/HeidiSQL/HeidiSQL

The General Public License (GPL) is shipped with the installer-package and should be located in the same folder as this file (gpl.txt).

If you simply wish to install and use this software, you need only be aware of the disclaimer conditions in the license, which are set out below.

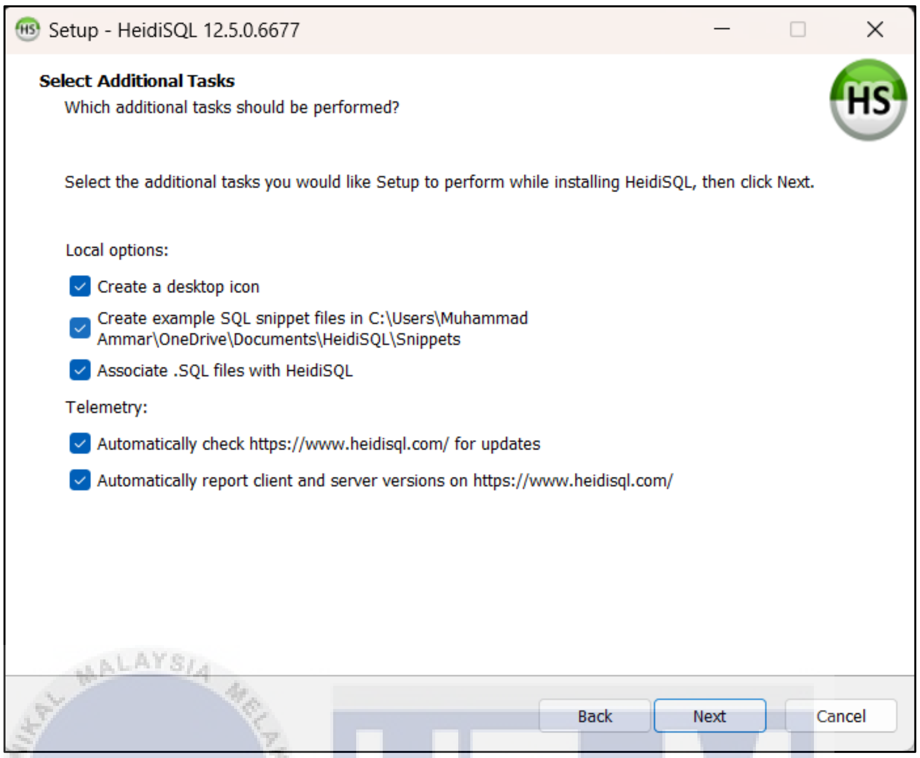
NO WARRANTY

Because the program is licensed free of charge, there is no warranty for the

[✓ I accept the agreement]
I do not accept the agreement

Figure 5.13 License Agreement

Step 3: In figure 5.13, check all the checkboxes and click next.



Setup - HeidiSQL 12.5.0.6677

Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing HeidiSQL, then click Next.

Local options:

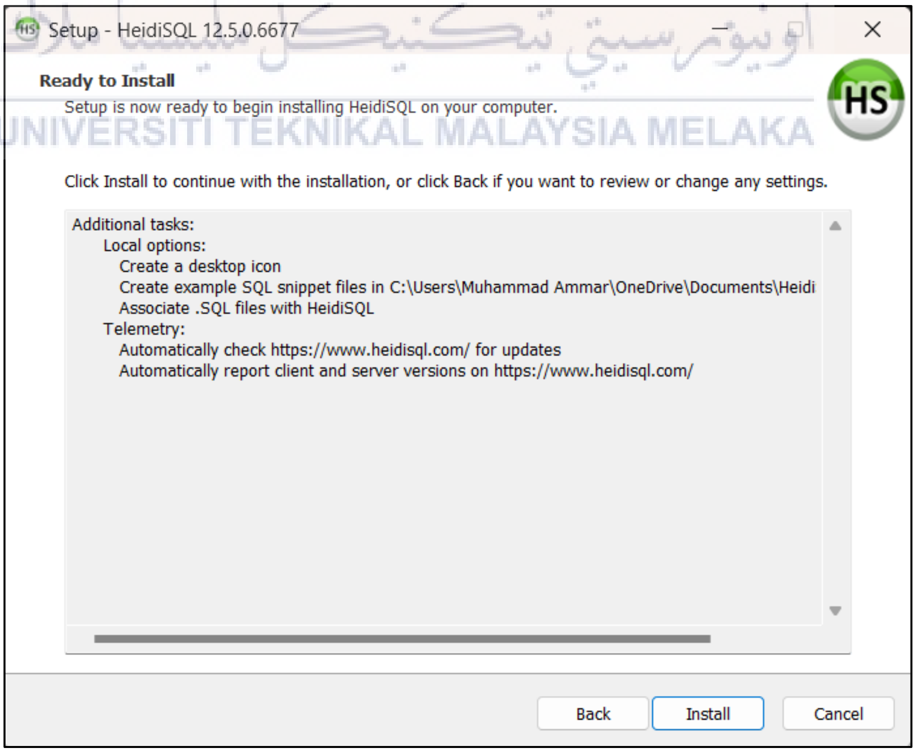
- [✓] Create a desktop icon
- [✓] Create example SQL snippet files in C:\Users\Muhammad Ammar\OneDrive\Documents\HeidiSQL\Snippets
- [✓] Associate .SQL files with HeidiSQL

Telemetry:

- [✓] Automatically check https://www.heidisql.com/ for updates
- [✓] Automatically report client and server versions on https://www.heidisql.com/

Figure 5.14 Setup HeidiSQL

Step 4: Click the install button as shown in figure 5.15. Then the HeidiSQL will start to install as shown in Figure 5.16.



Setup - HeidiSQL 12.5.0.6677

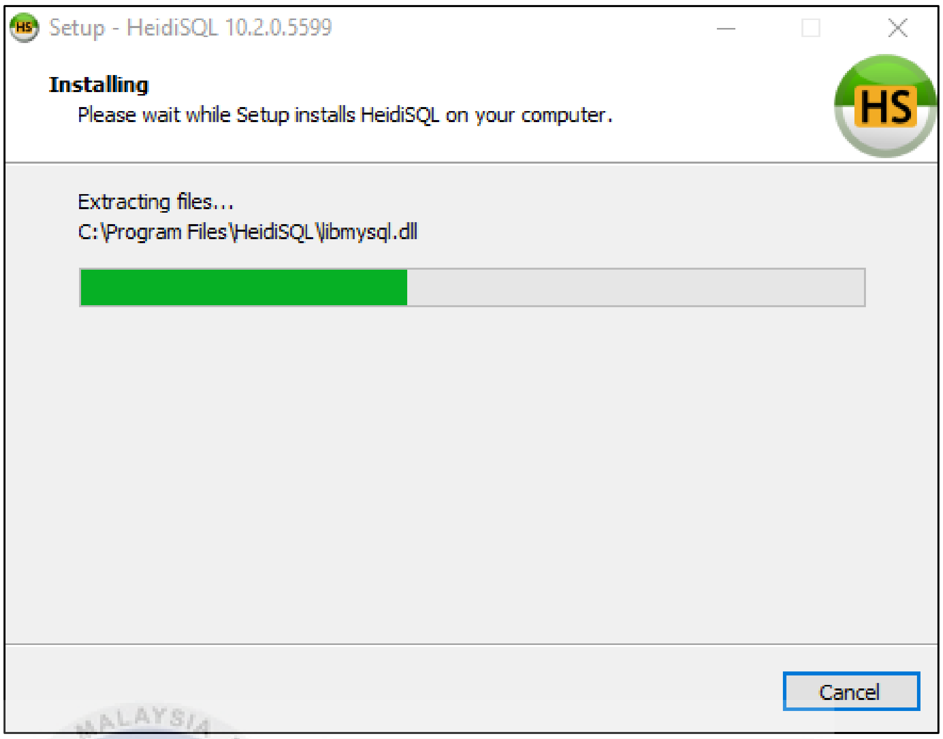
Ready to Install

Setup is now ready to begin installing HeidiSQL on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

```
Additional tasks:
    Local options:
      Create a desktop icon
      Create example SQL snippet files in C:\Users\Muhammad Ammar\OneDrive\Documents\Heidi
      Associate .SQL files with HeidiSQL
    Telemetry:
      Automatically check https://www.heidisql.com/ for updates
      Automatically report client and server versions on https://www.heidisql.com/
```

Figure 5.15 Ready to Install



Setup - HeidiSQL 10.2.0.5599

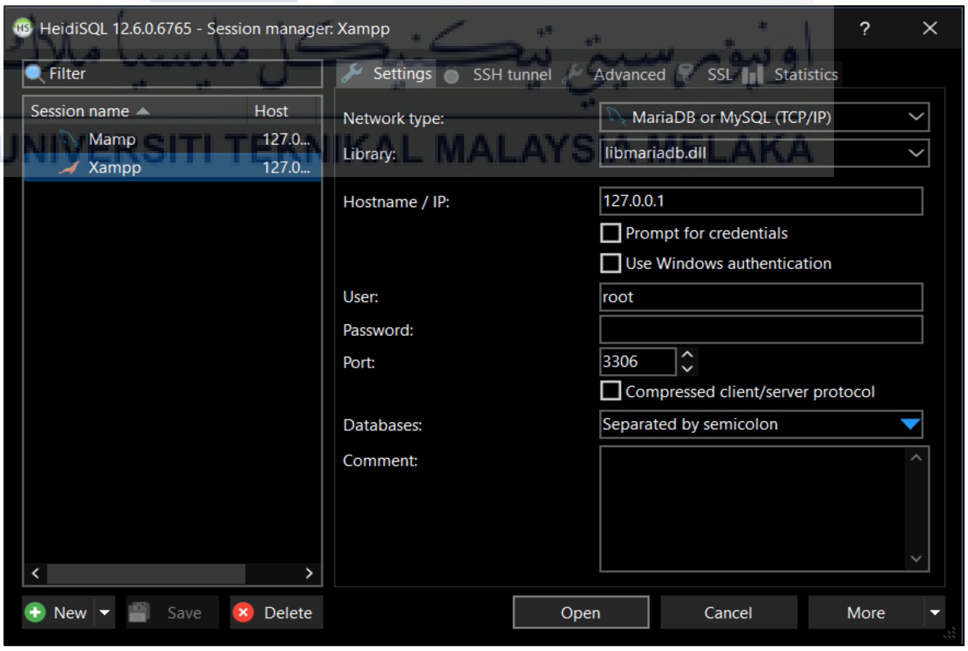
Installing

Please wait while Setup installs HeidiSQL on your computer.

Extracting files...
C:\Program Files\HeidiSQL\libmysql.dll

Figure 5.16 Installation Process

Step 5: After installation, Open HeidiSQL application then there will show as figure 5.17 and fill the form such as below and click open. Then it will open the session database successfully as shown in figure 5.18.



HeidiSQL 12.6.0.6765 - Session manager: Xampp

Session name	Host
Mamp	127.0...
Xampp	127.0...

Network type:	MariaDB or MySQL (TCP/IP)
Library:	libmariadb.dll
Hostname / IP:	127.0.0.1
	☐ Prompt for credentials
	☐ Use Windows authentication
User:	root
Password:	
Port:	3306
	☐ Compressed client/server protocol
Databases:	Separated by semicolon
Comment:	

Figure 5.17 Session Setup

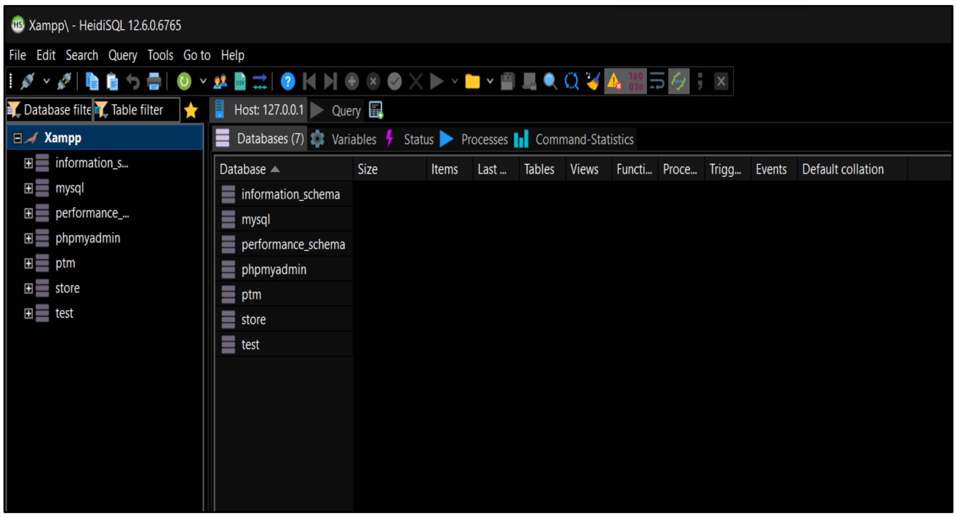


Figure 5.18 HeidiSQL Environment

5.3 Database Implementation

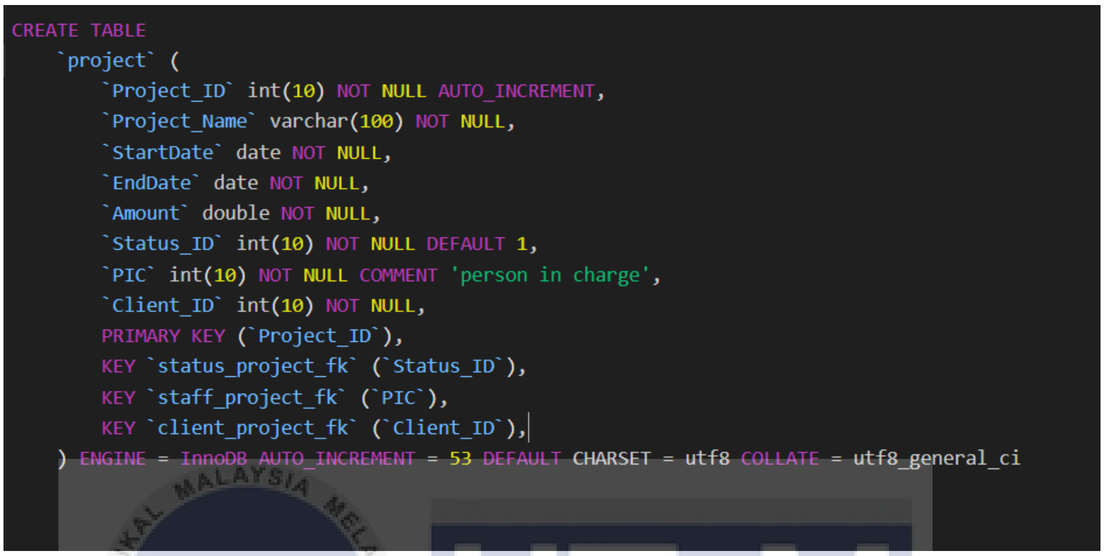
In Project Task Management System database implementation, the database is used to test the database query such as simple query, complex query, aggregate function, stored procedure, and triggers.

5.3.1 Data Definition Language (DDL)

Data Definition Language (DDL) encompasses a set of SQL commands dedicated to the creation and management of tables within a relational database. These statements, constituting the DDL, facilitate the formulation, alteration, and removal of objects integral to a database, which includes tables, indexes, and triggers. In the context of a project task management system, DDL commands play a pivotal role in establishing the structure of the underlying database. For instance, these commands are instrumental in creating tables to store project and task-related information, altering table structures to accommodate evolving requirements, and removing obsolete entities. The robust functionality of DDL in crafting and refining the database schema is fundamental to the efficiency and adaptability of a project task management system.

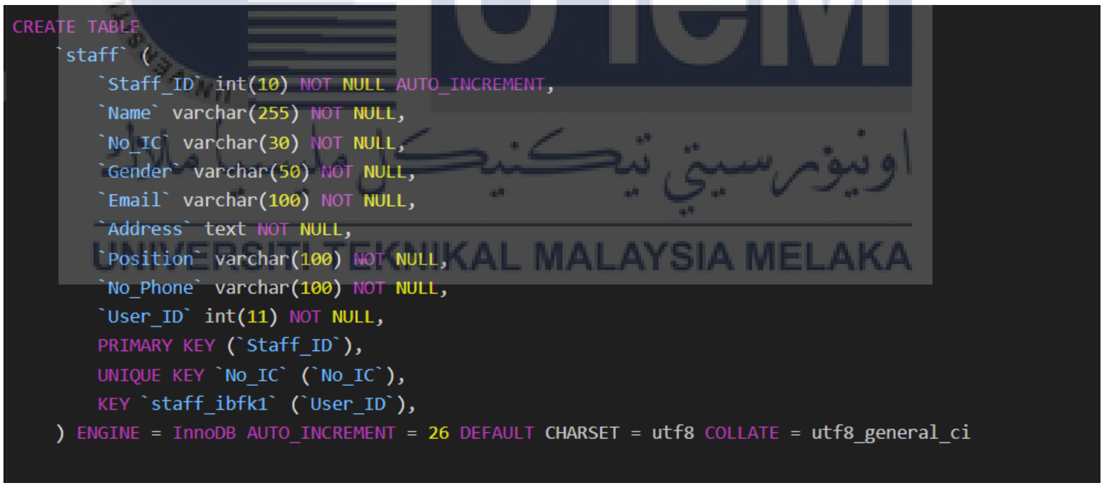
5.3.1.1 Create Table Using DDL Command

Based on figure 5.19 – figure 5.25 is showing data definition language (DDL) to create object table in database project task management system.



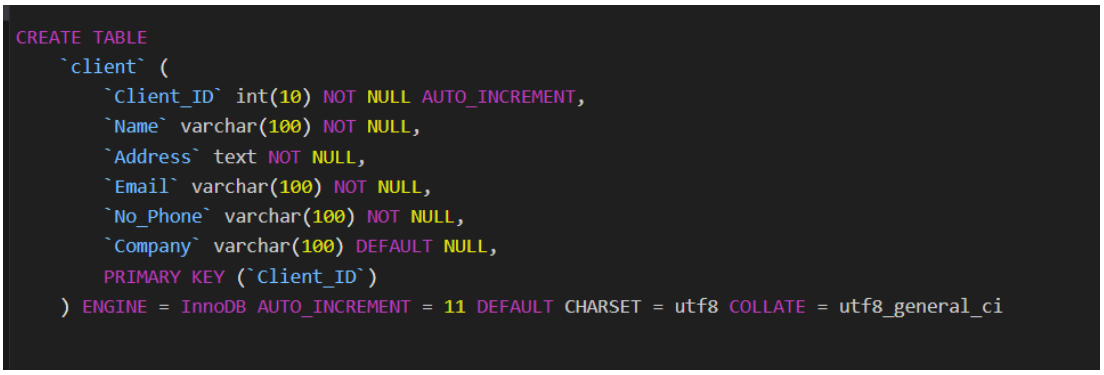
```
CREATE TABLE
    `project` (
        `Project_ID` int(10) NOT NULL AUTO_INCREMENT,
        `Project_Name` varchar(100) NOT NULL,
        `StartDate` date NOT NULL,
        `EndDate` date NOT NULL,
        `Amount` double NOT NULL,
        `Status_ID` int(10) NOT NULL DEFAULT 1,
        `PIC` int(10) NOT NULL COMMENT 'person in charge',
        `Client_ID` int(10) NOT NULL,
        PRIMARY KEY (`Project_ID`),
        KEY `status_project_fk` (`Status_ID`),
        KEY `staff_project_fk` (`PIC`),
        KEY `client_project_fk` (`Client_ID`),
    ) ENGINE = InnoDB AUTO_INCREMENT = 53 DEFAULT CHARSET = utf8 COLLATE = utf8_general_ci
```

Figure 5.19 DDL Project Table



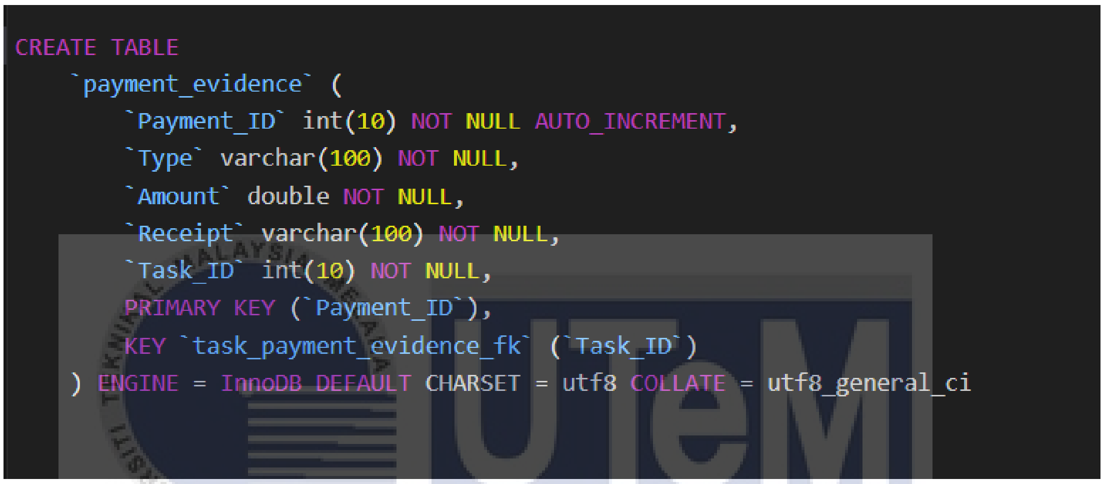
```
CREATE TABLE
    `staff` (
        `Staff_ID` int(10) NOT NULL AUTO_INCREMENT,
        `Name` varchar(255) NOT NULL,
        `No_IC` varchar(30) NOT NULL,
        `Gender` varchar(50) NOT NULL,
        `Email` varchar(100) NOT NULL,
        `Address` text NOT NULL,
        `Position` varchar(100) NOT NULL,
        `No_Phone` varchar(100) NOT NULL,
        `User_ID` int(11) NOT NULL,
        PRIMARY KEY (`Staff_ID`),
        UNIQUE KEY `No_IC` (`No_IC`),
        KEY `staff_ibfk1` (`User_ID`),
    ) ENGINE = InnoDB AUTO_INCREMENT = 26 DEFAULT CHARSET = utf8 COLLATE = utf8_general_ci
```

Figure 5.20 DDL Staff Table



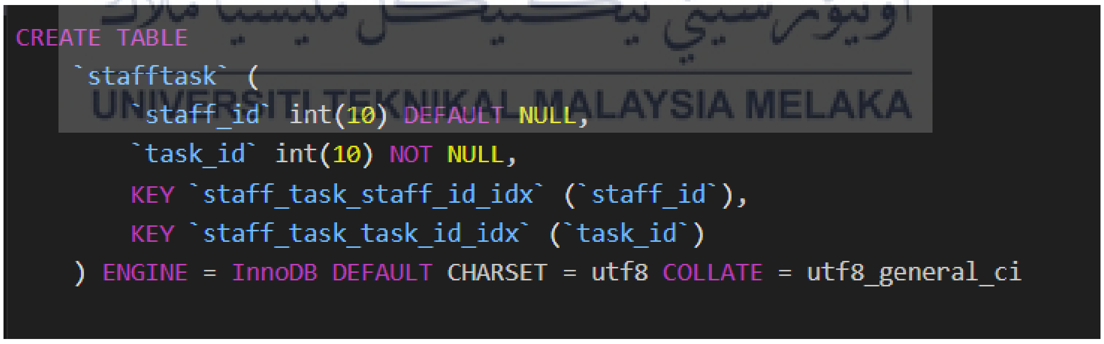
```
CREATE TABLE
    `client` (
        `Client_ID` int(10) NOT NULL AUTO_INCREMENT,
        `Name` varchar(100) NOT NULL,
        `Address` text NOT NULL,
        `Email` varchar(100) NOT NULL,
        `No_Phone` varchar(100) NOT NULL,
        `Company` varchar(100) DEFAULT NULL,
        PRIMARY KEY (`Client_ID`)
    ) ENGINE = InnoDB AUTO_INCREMENT = 11 DEFAULT CHARSET = utf8 COLLATE = utf8_general_ci
```

Figure 5.21 DDL Client Table



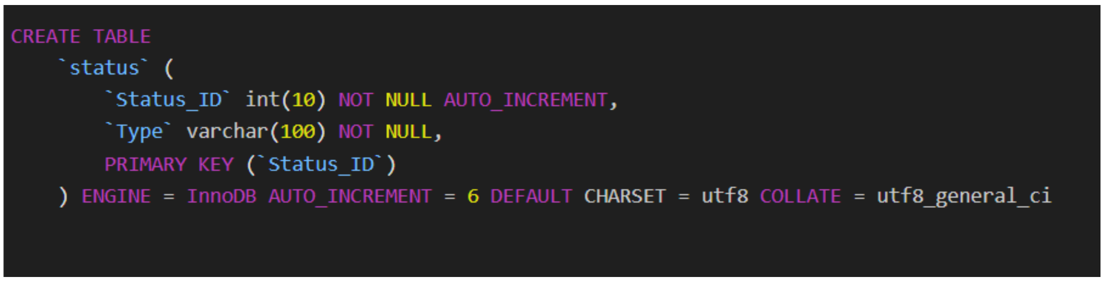
```
CREATE TABLE
    `payment_evidence` (
        `Payment_ID` int(10) NOT NULL AUTO_INCREMENT,
        `Type` varchar(100) NOT NULL,
        `Amount` double NOT NULL,
        `Receipt` varchar(100) NOT NULL,
        `Task_ID` int(10) NOT NULL,
        PRIMARY KEY (`Payment_ID`),
        KEY `task_payment_evidence_fk` (`Task_ID`)
    ) ENGINE = InnoDB DEFAULT CHARSET = utf8 COLLATE = utf8_general_ci
```

Figure 5.22 DDL Payment Table



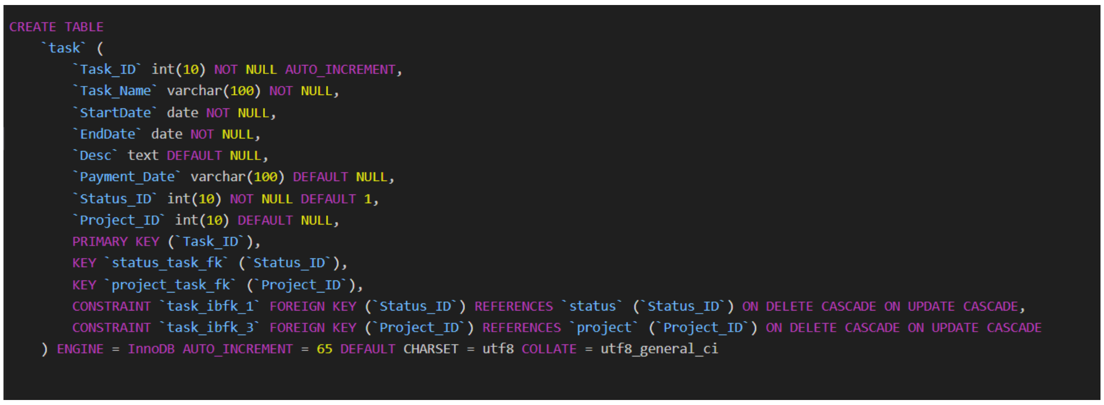
```
CREATE TABLE
    `stafftask` (
        `staff_id` int(10) DEFAULT NULL,
        `task_id` int(10) NOT NULL,
        KEY `staff_task_staff_id_idx` (`staff_id`),
        KEY `staff_task_task_id_idx` (`task_id`)
    ) ENGINE = InnoDB DEFAULT CHARSET = utf8 COLLATE = utf8_general_ci
```

Figure 5.23 DDL StaffTask Table



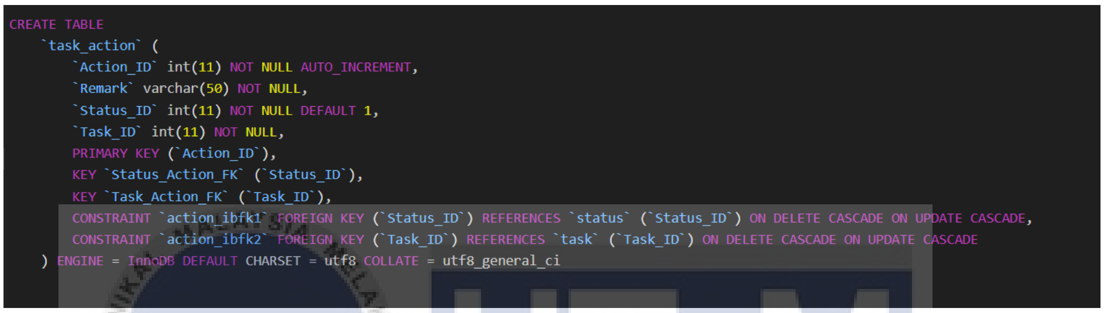
```
CREATE TABLE
    `status` (
        `Status_ID` int(10) NOT NULL AUTO_INCREMENT,
        `Type` varchar(100) NOT NULL,
        PRIMARY KEY (`Status_ID`)
    ) ENGINE = InnoDB AUTO_INCREMENT = 6 DEFAULT CHARSET = utf8 COLLATE = utf8_general_ci
```

Figure 5.24 DDL Status Table



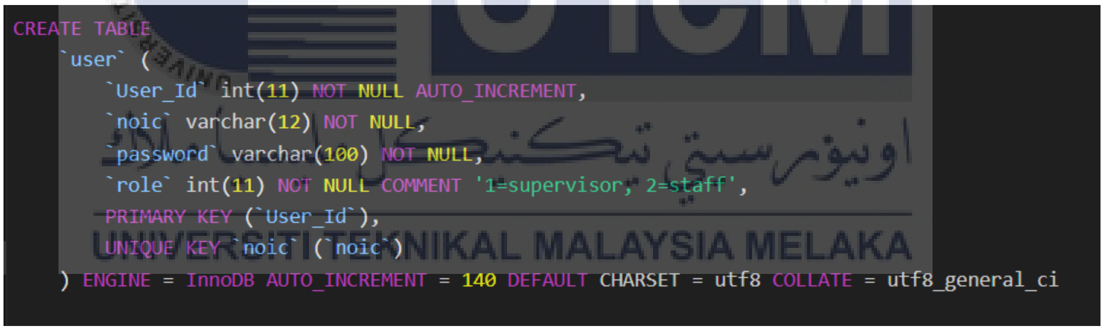
```
CREATE TABLE
    `task` (
        `Task_ID` int(10) NOT NULL AUTO_INCREMENT,
        `Task_Name` varchar(100) NOT NULL,
        `StartDate` date NOT NULL,
        `EndDate` date NOT NULL,
        `Desc` text DEFAULT NULL,
        `Payment_Date` varchar(100) DEFAULT NULL,
        `Status_ID` int(10) NOT NULL DEFAULT 1,
        `Project_ID` int(10) DEFAULT NULL,
        PRIMARY KEY (`Task_ID`),
        KEY `status_task_fk` (`Status_ID`),
        KEY `project_task_fk` (`Project_ID`),
        CONSTRAINT `task_ibfk_1` FOREIGN KEY (`Status_ID`) REFERENCES `status` (`Status_ID`) ON DELETE CASCADE ON UPDATE CASCADE,
        CONSTRAINT `task_ibfk_3` FOREIGN KEY (`Project_ID`) REFERENCES `project` (`Project_ID`) ON DELETE CASCADE ON UPDATE CASCADE
    ) ENGINE = InnoDB AUTO_INCREMENT = 65 DEFAULT CHARSET = utf8 COLLATE = utf8_general_ci
```

Figure 5.25 DDL Task Table



```
CREATE TABLE
    `task_action` (
        `Action_ID` int(11) NOT NULL AUTO_INCREMENT,
        `Remark` varchar(50) NOT NULL,
        `Status_ID` int(11) NOT NULL DEFAULT 1,
        `Task_ID` int(11) NOT NULL,
        PRIMARY KEY (`Action_ID`),
        KEY `Status_Action_FK` (`Status_ID`),
        KEY `Task_Action_FK` (`Task_ID`),
        CONSTRAINT `action_ibfk1` FOREIGN KEY (`Status_ID`) REFERENCES `status` (`Status_ID`) ON DELETE CASCADE ON UPDATE CASCADE,
        CONSTRAINT `action_ibfk2` FOREIGN KEY (`Task_ID`) REFERENCES `task` (`Task_ID`) ON DELETE CASCADE ON UPDATE CASCADE
    ) ENGINE = InnoDB DEFAULT CHARSET = utf8 COLLATE = utf8_general_ci
```

Figure 5.26 DDL Task_Action table



```
CREATE TABLE
    `user` (
        `User_Id` int(11) NOT NULL AUTO_INCREMENT,
        `noic` varchar(12) NOT NULL,
        `password` varchar(100) NOT NULL,
        `role` int(11) NOT NULL COMMENT '1=supervisor, 2=staff',
        PRIMARY KEY (`User_Id`),
        UNIQUE KEY `noic` (`noic`)
    ) ENGINE = InnoDB AUTO_INCREMENT = 140 DEFAULT CHARSET = utf8 COLLATE = utf8_general_ci
```

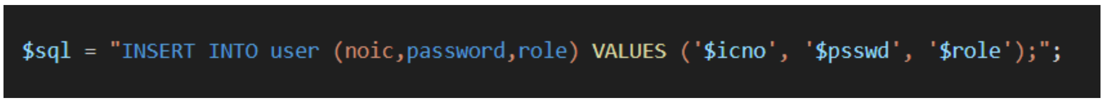
Figure 5.27 DDL User Table

5.3.2 Data Manipulation Language (DML)

Data Manipulation Language (DML) comprises SQL commands specifically designed for the manipulation of data within a relational database. DML statements are instrumental in inserting, updating, and deleting data records in tables. In the realm of a project task management system, DML commands play a crucial role in dynamically managing information. For instance, these commands are utilized to insert new tasks into a project, update the status of ongoing activities, or delete completed tasks. The versatility of DML extends to enabling the retrieval of specific data subsets through SELECT queries, thereby offering a comprehensive toolkit for seamlessly managing and interacting with the evolving dataset in a project task management environment.

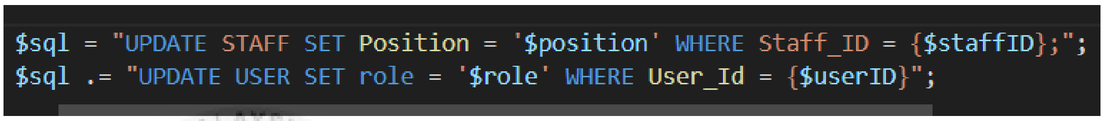
5.3.2.1 Manipulating Data Using DML

Based on figure 5.28 – figure 5.31 is showing data manipulation language (DML) to alter or manipulate data in table.



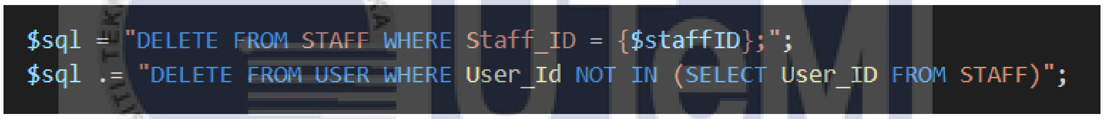
```
$sql = "INSERT INTO user (noic,password,role) VALUES ('$icno', '$psswd', '$role');";
```

Figure 5.28 Insert Statement into User Table



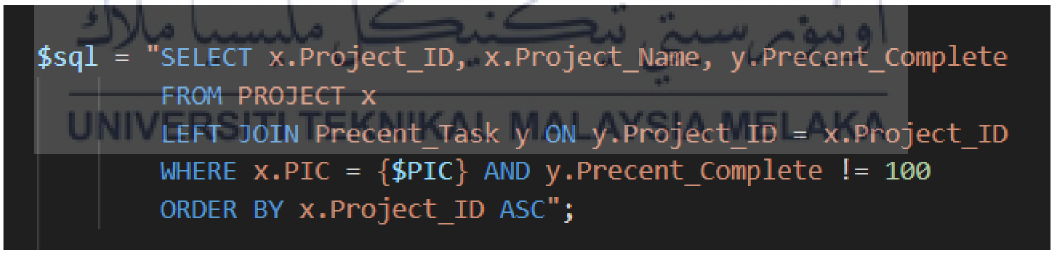
```
$sql = "UPDATE STAFF SET Position = '$position' WHERE Staff_ID = {$staffID};";
$sql .= "UPDATE USER SET role = '$role' WHERE User_Id = {$userID}";
```

Figure 5.29 Update Staff



```
$sql = "DELETE FROM STAFF WHERE Staff_ID = {$staffID};";
$sql .= "DELETE FROM USER WHERE User_Id NOT IN (SELECT User_ID FROM STAFF)";
```

Figure 5.30 Delete Staff



```
$sql = "SELECT x.Project_ID, x.Project_Name, y.Precent_Complete
        FROM PROJECT x
        LEFT JOIN Precent_Task y ON y.Project_ID = x.Project_ID
        WHERE x.PIC = {$PIC} AND y.Precent_Complete != 100
        ORDER BY x.Project_ID ASC";
```

Figure 5.31 Select data from table.

5.3.3 Stored Procedure

Stored Procedures are a category of database objects that encapsulate a set of SQL statements, creating a reusable and modular unit of functionality. In the context of a project task management system, Stored Procedures serve as a powerful tool for executing complex operations on the database. These procedures can be designed to create, modify, or delete records in tables, retrieve specific data based on predefined criteria, or perform intricate data manipulations. Additionally, Stored Procedures enhance security by allowing controlled access to the underlying database, limiting direct user interaction. Their role in a project task management system includes streamlining data operations, ensuring consistency, and providing a structured and efficient means to interact with the database.

Table 5.1 Stored Procedure Table

Description	Stored Procedure Query

The stored procedure is designed to insert data into a bridge table (StaffTask) while ensuring that the pair of foreign keys doesn't already exist in the table. If a duplicate pair is detected, it raises a custom application error.	`CREATE PROCEDURE` `` `InsertStaffTask` `` `( IN staffID INT, IN taskID INT)`
	`BEGIN`
	`      DECLARE duplicate_count INT;`
	`-- Check if the pair already exists in the bridge table`
	`      SELECT COUNT(*) INTO duplicate_count FROM stafftask WHERE staff_id = staffID AND task_id = taskID;`
	`  -- If the pair doesn't exist, insert it`
	`      IF duplicate_count = 0 THEN`
	`               INSERT INTO stafftask (staff_id, task_id)`
	`               VALUES (staffID, taskID);`
	`       ELSE`
	`-- Raise a custom application error with a specific message`
	`            SIGNAL SQLSTATE '45000'`
	`            SET MESSAGE_TEXT = 'Duplicate pair found in StaffTask Table: Cannot insert duplicate.';`
	`          END IF;`
	`END`

The stored procedure PrecentStaffTask will display the precent task based on project and the staff.	
	```
	DELIMITER $$
	CREATE DEFINER=\`root\`@\`localhost\`
	PROCEDURE \`PrecentStaffTask\`(IN staffID
	INT, IN projectID INT)
	BEGIN
	    WITH z AS (
	        SELECT p.Project_ID,
	p.Project_Name, COUNT(x.stafftask_id)
	AS Total,
	            COUNT(CASE WHEN x.status_id
	= 1 THEN 1 END) AS To_Do,
	            COUNT(CASE WHEN x.status_id
	= 2 THEN 1 END) AS In_Progress,
	            COUNT(CASE WHEN x.status_id
	= 3 THEN 1 END) AS Pending,
	            COUNT(CASE WHEN x.status_id
	= 4 THEN 1 END) AS In_Review,
	            COUNT(CASE WHEN x.status_id
	= 5 THEN 1 END) AS Complete
	        FROM stafftask x
	        JOIN task t ON t.Task_ID =
	x.task_id
	        JOIN project p ON p.Project_ID
	= t.Project_ID
	        WHERE x.staff_id = staffID AND
	t.Project_ID = projectID
	        GROUP BY p.Project_ID,
	p.Project_Name
	    )
	    SELECT

	ROUND(((z.In_Progress/z.Total)*100),0)
	AS In_Progress,

	ROUND(((z.Pending/z.Total)*100),0) AS
	Pending,

	ROUND(((z.In_Review/z.Total)*100),0) AS
	In_Review,

	ROUND(((z.Complete/z.Total)*100),0) AS
	Complete
	    FROM z;
	END$$
	DELIMITER ;
	```

5.3.4 Trigger

Triggers in a database refer to special types of stored procedures that are automatically executed or fired in response to predefined events such as INSERT, UPDATE, or DELETE operations on a specific table. Within the framework of a project task management system, triggers play a pivotal role in automating actions associated with data changes. For example, a trigger could be implemented to automatically update a timestamp or send notifications when a task status changes. Triggers contribute to maintaining data integrity, enforcing business rules, and executing actions that need to be synchronized with database modifications. In essence, triggers act as event-driven mechanisms, enhancing the responsiveness and efficiency of a project task management system by executing predefined actions in real-time based on specific database events.

Table 5.2 Trigger Table

Description	Trigger Query

The update_parent_on_insert trigger will automatically update a main record when a new related record is added. System to manage tasks in a project, this trigger ensures that when a new task is added, the overall project status or completion is updated without needing someone to do it manually.	`CREATE TRIGGER `update_parent_on_insert``
	`AFTER INSERT ON `task``
	`FOR EACH ROW`
	`BEGIN`
	`    UPDATE Project SET status_id = 2`
	`WHERE project_id = NEW.project_id;`
	`END`

The update_project_status trigger is like a helper in a database that automatically adjusts the project status when something changes. For example, if you're using a system to track tasks in a project, this trigger ensures that when a task is completed or updated, the overall project status is also automatically updated. It's a way to keep things organized without having to manually update everything.	CREATE TRIGGER update_project_status
	AFTER UPDATE ON task
	FOR EACH ROW
	BEGIN
	      DECLARE total_tasks INT;
	      DECLARE completed_tasks INT;
	      SET total_tasks = (SELECT COUNT(*)
	FROM Task WHERE project_id =
	NEW.project_id);
	      SET completed_tasks = (SELECT
	COUNT(*) FROM Task WHERE project_id =
	NEW.project_id AND status_id = 5);
	      IF completed_tasks = total_tasks
	THEN
	            UPDATE Project SET status_id
	= 5 WHERE project_id = NEW.project_id;
	      ELSE
	            UPDATE Project SET status_id
	= 2 WHERE project_id = NEW.project_id;
	      END IF
	END

The trigger will have triggered when update on task will be calculate the complete task if all task in project have been completed then its will be update status at project will be complete.	CREATE TRIGGER \`calculate_remain\` AFTER
	INSERT ON \`payment\`
	 FOR EACH ROW BEGIN
	    DECLARE remain DOUBLE;
	    DECLARE amountP DOUBLE;
	    DECLARE total DOUBLE;
	    -- Calculate Sum Amount
	    SELECT SUM(Amount) INTO remain FROM
	Payment WHERE Project_ID =
	NEW.Project_ID;
	    SELECT Amount INTO amountP FROM
	Project WHERE Project_ID =
	NEW.Project_ID;
	    SET total = remain - amountP;
	
	    UPDATE Project SET Remain_amount =
	total WHERE Project_ID =
	NEW.Project_ID;
	END

The check_due_date trigger in MySQL works before adding a new record to the Tasks table. It ensures that the due date for the task is set in the future by comparing it with the current date. If the due date is not in the future, the trigger stops the insertion and signals that the due date must be ahead for the task to be added to the Tasks table.	
	```
	CREATE TRIGGER `check_due_date`
	BEFORE INSERT ON `project`
	FOR EACH ROW
	BEGIN
	      IF   CompareDates   (NEW.StartDate,
	NEW.EndDate) THEN
	            SIGNAL SQLSTATE '45000'
	                  SET MESSAGE_TEXT = 'Due
	date must be in the future.';
	      END IF;
	END
	```

5.3.5 Function

Functions in a database refer to predefined routines that encapsulate a set of SQL statements to perform specific tasks. In the context of a project task management system, functions serve as modular units of functionality that can be utilized for data processing or computation.

Table 5.3 Function Table

Description	Function Query

The CompareDates function is a SQL function designed to compare two date values, startDate and endDate. It returns an integer result: true if endDate is greater than startDate, false if endDate is less than startDate. The function employs a basic conditional structure to determine the relationship between the provided dates, allowing for a straightforward assessment of their chronological order.	`CREATE FUNCTION `CompareDates` (`startDate` DATE, `endDate` DATE)`
	`RETURNS int(11)`
	`BEGIN`
	`      DECLARE result INT;`
	`      IF endDate <= startDate THEN`
	`               SET result = true;`
	`      -- endDate is less than to startDate`
	`       ELSE`
	`               SET result = false;`
	`      -- endDate is greater than startDate`
	`          END IF;`
	`          RETURN result;`
	`END;`

<table>
<tr><td>The function INITCAP () will replace the first letter of every word in a character string with an uppercase letter.</td><td>

```
CREATE FUNCTION `INITCAP` (`str`
VARCHAR (8000))
RETURNS varchar (8000)
BEGIN
      DECLARE len INT DEFAULT LENGTH
(str);
      DECLARE pos INT DEFAULT 1;
      DECLARE ch CHAR (1);
      DECLARE ch_ascii INT;
      DECLARE out_str VARCHAR (8000)
DEFAULT '';
      DECLARE prev_alphanum INT
DEFAULT 0;
      WHILE pos <= len
      DO
             SET ch = SUBSTRING (str,
pos, 1);
             SET ch_ascii = ASCII (ch);
             IF prev_alphanum = 1 THEN
            SET out_str = CONCAT (RPAD
(out_str, pos - 1,' '), LOWER (ch));
-- RPAD is required to append ' '
             ELSE
                  SET out_str = CONCAT
(RPAD (out_str, pos - 1,' '),
UPPER(ch));
             END IF;
             IF ch_ascii <= 47 OR
(ch_ascii BETWEEN 58 AND 64) OR
(ch_ascii BETWEEN 91 AND 96) OR
(ch_ascii BETWEEN 123 AND 126) THEN
            SET prev_alphanum = 0;
             ELSE
            SET prev_alphanum = 1;
             END IF;
             SET pos = pos + 1;
      END WHILE;
    RETURN out_str;
END;
```

</td></tr>
</table>

5.3.6 Object Views

Views in a database represent virtual tables that are derived from the result of a SELECT query and do not store the data themselves. Within the context of a project task management system, views provide a means to present a tailored perspective on the underlying data. For instance, a view could be designed to display a consolidated overview of project details or filter tasks based on specific criteria. Views enhance data abstraction, simplifying complex queries and ensuring that users interact with a more understandable and focused representation of the data. Their role in a project task management system involves offering a structured and simplified view of the underlying information, supporting efficient decision-making and facilitating a user-friendly interface for accessing relevant data.

Table 5.4 Object Views Table

Description	Views Query

This code creates a simplified summary, called status_task, combining information about projects and their tasks. For each project, it shows the total number of tasks and breaks them down by different statuses like to-do, in progress, pending, under review, and completed.	CREATE VIEW \`status_task\` AS
	`SELECT p.Project_ID, p.Project_Name,`
	`COUNT(t.Task_ID) AS Total_Task,`
	`COUNT (CASE WHEN t.Status_ID = 1 THEN 1 END) AS Total_To_Do,`
	`COUNT (CASE WHEN t.Status_ID = 2 THEN 1 END) AS Total_In_Progress,`
	`COUNT (CASE WHEN t.Status_ID = 3 THEN 1 END) AS Total_Pending,`
	`COUNT (CASE WHEN t.Status_ID = 4 THEN 1 END) AS Total_Review,`
	`COUNT (CASE WHEN t.Status_ID = 5 THEN 1 END) AS Total_Complete`
	`FROM task t`
	`JOIN project p ON p.Project_ID = t.Project_ID`
	`JOIN status s ON s.Status_ID = t.Status_ID`
	`GROUP by p.Project_Name, p.Project_ID;`

This code creates a simplified view called percent_task based on the previous view. It calculates the percentage of tasks in different states (like to-do,	`CREATE VIEW `precent_task` AS`
	`SELECT Project_ID, Project_Name,`
	`ROUND(((Total_To_Do/Total_Task)*100),0) AS Precent_ToDo,`
	`ROUND(((Total_In_Progress/Total_Task)*100),0) AS Precent_InProgress,`
	`ROUND(((Total_Pending/Total_Task)*100),0) AS Precent_Pending,`

in progress, etc.) for each project. So, for each project, it tells what percentage of tasks are in different stages. It's like a simplified report showing how much of the work is done, in progress, or pending.	`ROUND(((Total_Review/Total_Task) *100),0) AS Precent_Review, ROUND(((Total_Complete/Total_Task) *100),0) AS Precent_Complete FROM status_task;`

5.3.7 Data Loading Process

The Project Task Management System, abbreviated as PTMS, serves as a comprehensive platform for storing and processing data within the project task management framework. Its core objective is to offer valuable insights to project team members, managers, and stakeholders. All crucial project-related data, including task details, progress updates, team member contributions, and project milestones, is systematically stored in the database. In the PTMS, the Extract, Transform, Load (ETL) process is seamlessly integrated to handle large volumes of both structured and unstructured project data. Extraction involves collecting and reading data from diverse project sources. Transformation is a crucial phase where project data undergoes necessary conversions before being stored in an alternative database. Loading, the final step, represents the culmination of the process, involving the creation and insertion of project data into the designated database.

5.4 Conclusion.

In summary, this section offers insights into the setup of the software development environment for the Project Task Management System (PTMS). It outlines the steps for installing Apache, XAMPP, PHP, and MySQL on the Windows platform. Additionally, the chapter delves into the implementation of the database, essential for overseeing the system's processes. Correspondingly, the system is founded on critical business logic, encompassing commands such as Data Definition Language (DDL), Data Manipulation Language (DML), stored procedures, triggers, creation of database tables, and constraint commands.

CHAPTER 6: TESTING

6.1 Introduction

Testing plays a pivotal role in the implementation lifecycle of the Project Task Management System (PTMS), a critical phase in ensuring the system's functionality aligns with its objectives. Given PTMS's overarching goal of refining project management in academic and industry settings, meticulous testing is imperative to validate its reliability, adaptability, and performance across a spectrum of usage scenarios. This phase encompasses diverse testing methodologies, ranging from unit testing to system integration testing, to affirm that PTMS not only meets its predetermined objectives but also excels in real-world applications, fortifying its role in reshaping project management practices.

In pursuit of the robustness and effectiveness of PTMS, this introduction provides an overview of the comprehensive testing strategy implemented. The testing protocols are designed to scrutinize PTMS's capabilities and ensure it seamlessly meets the diverse needs of its users. Through this systematic testing approach, PTMS aims not just to function but to excel, delivering a transformative experience in project management within both academic and industry spheres.

6.2 Test Plans

A test plan serves as a technical document outlining the test strategy, encompassing the scope, necessary resources (including manpower, software, and hardware), testing schedule, and expected test deliverables. This comprehensive document provides a detailed comprehension of the system's workflow and functionalities. It articulates the method by which each aspect will undergo testing to ascertain if the system aligns with its design, identify any potential bugs, and uncover its practical limitations.

6.2.1 Test Organization

In the context of PTMS, the testing framework encompasses three distinct users: candidates, recruiters, and recruiter managers. The testing process will thoroughly evaluate both functional requirements and non-requirements for each of these user categories. Table 6.1 provides a visual representation of the four users who will undergo testing, delineating their respective user responsibilities.

Table 6.1 User Responsibility

Tester ID	Stakeholders	Responsible
T1	Amir Hadi as a Project Manager	• Develop a clear plan for tasks, timelines, and resources. • Monitor resource usage and adjust plans as needed to ensure optimal project execution. • Testing for black box.
T2	Erry, Auni, Anis, Najmi as a Staff	• Completed assigned tasks efficiently and within specified timelines. • Testing for black box.
T3	Ammar as a Developer	• Programmer for development PTMS. • Testing for white box.

6.2.2 Test Environment

In the context of PTMS, the Test Environment is the configuration of essential components such as software, hardware, operating systems, tools, and network settings. Tables 6.2 and 6.3 provide a comprehensive list of the required hardware and software for the test environment of PTMS.

Table 6.2 Hardware List

Environment	Description
Laptop	Honor magicbook x15
Random Access Memory (RAM)	8 GB
Printer	HP Deskjet Ink Advantage 2050

Table 6.3 Software List

Environment	Description
Database	MySQL to manage data in the database table that runs on a server
Web Server	XAMPP Server to create a local web server and use to deployment and testing purposes
Web Browser	Mozilla Firefox to use run the PHP source code and test the system interface functionality
Sublime	To use write PHP code.

6.2.3 Test Schedule

The Test Schedule for the Project Task Management System (PTMS) is a crucial aspect of the testing phase, delineating a structured plan for the systematic execution of test cases. This introduction offers an overview of the scheduled activities, milestones, and timelines devised to rigorously evaluate PTMS's functionalities. As PTMS aims to revolutionize project management practices within academic and industry settings, the test schedule becomes a roadmap for ensuring that the system not only meets its defined objectives but also stands resilient in diverse usage scenarios. This phase includes a detailed breakdown of testing activities, providing stakeholders with a clear timeline for each testing phase and ensuring a thorough evaluation of PTMS's reliability and effectiveness.

Table 6.4 Testing Schedule

Testing Task	Testing Activity	Start Date	End Date
Login	Unit Testing, Integration Testing and User acceptance	10-6-2023	12-6-2023

Testing Task	Testing Activity	Start Date	End Date
Manage Project	Unit Testing, Integration Testing and User acceptance	14-6-2023	14-6-2023
Manage Task	Unit Testing, Integration Testing and User acceptance	23-9-2023	24-9-2023
Manage Payment	Unit Testing, Integration Testing and User acceptance	25-10-2023	1-11-2023
Reporting	Unit Testing, Integration Testing and User acceptance	20-12-2023	30-1-2024

6.3 Test Strategy

Testing is a critical phase within the development of the Project Task Management System (PTMS), influencing the system's overall reliability and effectiveness. In this context, we delve into three distinctive testing strategies: White Box, Black Box, and Grey Box Testing, each playing a crucial role in ensuring the robustness of PTMS.

White Box Testing for PTMS involves a thorough examination of the internal logic and structure of the system. This strategy scrutinizes the system's code, algorithms, and data flows, aiming to identify inefficiencies, security vulnerabilities, and potential flaws within the software. By providing a detailed view of PTMS's internal workings, White Box Testing ensures a comprehensive evaluation of its structural integrity, essential for a reliable project management system.

On the other hand, Black Box Testing focuses solely on the inputs and outputs of PTMS, disregarding its internal code structure. Testers evaluate the system based on predefined specifications, simulating real-world user interactions to ensure PTMS meets user expectations and functions as intended. This approach is crucial for verifying the external behavior and functionality of PTMS under various usage scenarios, emphasizing user-centric testing.

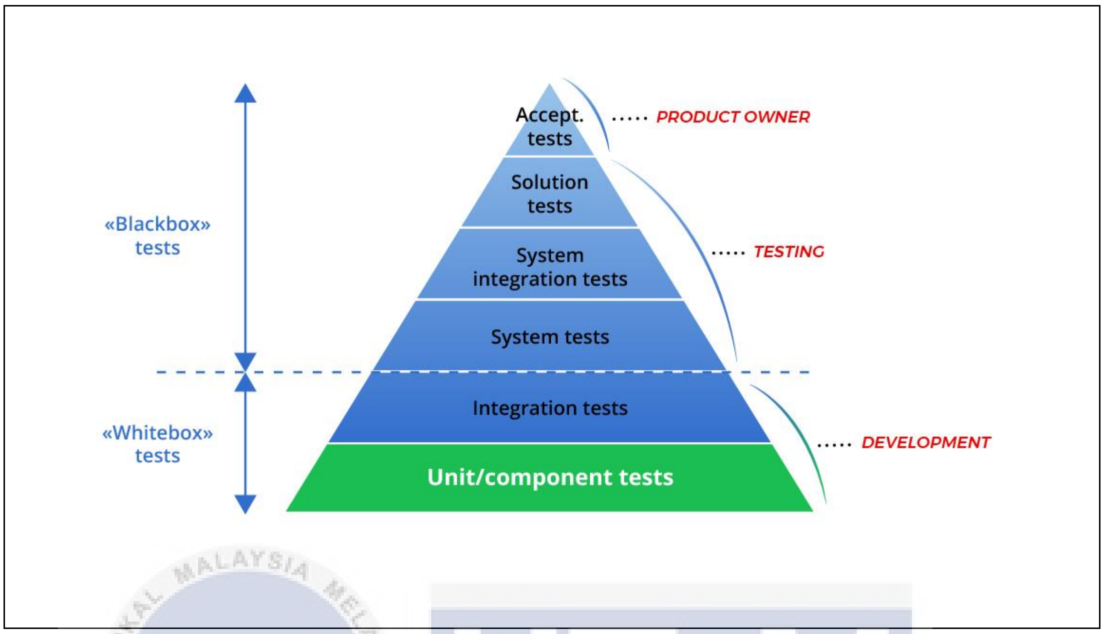


Figure 6.1 Layer of Testing [1]

6.3.1 Classes of Test

Below are several types of test class description that are being implemented on Project Task Management System (PTMS).

i. Error Handling

In the intricate realm of the Project Task Management System (PTMS), the introduction of error handling takes center stage as a fundamental element in fortifying the system's resilience and user experience. Recognizing the potential for unforeseen errors in the dynamic landscape of project management, PTMS integrates a robust error handling mechanism. This mechanism not only swiftly identifies and reports errors but also ensures the system's adept recovery, thereby sustaining uninterrupted functionality. The forthcoming sections will meticulously detail the strategies employed within PTMS for error detection, reporting, and recovery, embodying a proactive approach to address challenges and elevate the overall reliability of the system.

ii. Security Test

Security testing encompasses a comprehensive evaluation of the system's defenses, aiming to identify and rectify weaknesses in data protection, access controls, and overall system integrity. As an essential facet of PTMS development, security testing ensures that confidential information remains safeguarded, and the system operates with resilience against unauthorized access or malicious activities. The subsequent sections will delve into the methodologies and protocols employed for security testing within PTMS, emphasizing a proactive approach to fortify the system's defenses and instill confidence in its users. The security test is testing as shown in section test description Table 6.7

iii. Integration Test

Integration testing ensures that individual modules or units, when combined, function harmoniously as an integrated whole. This process assesses data flow, communication between system elements, and the overall interoperability of various functionalities within PTMS. By conducting thorough integration testing, PTMS aims to identify and resolve any inconsistencies or issues that may arise when different parts of the system come together. The upcoming sections will detail the methodologies and strategies employed for integration testing in PTMS, emphasizing the commitment to a cohesive and well-coordinated system architecture.

6.4 Test Design

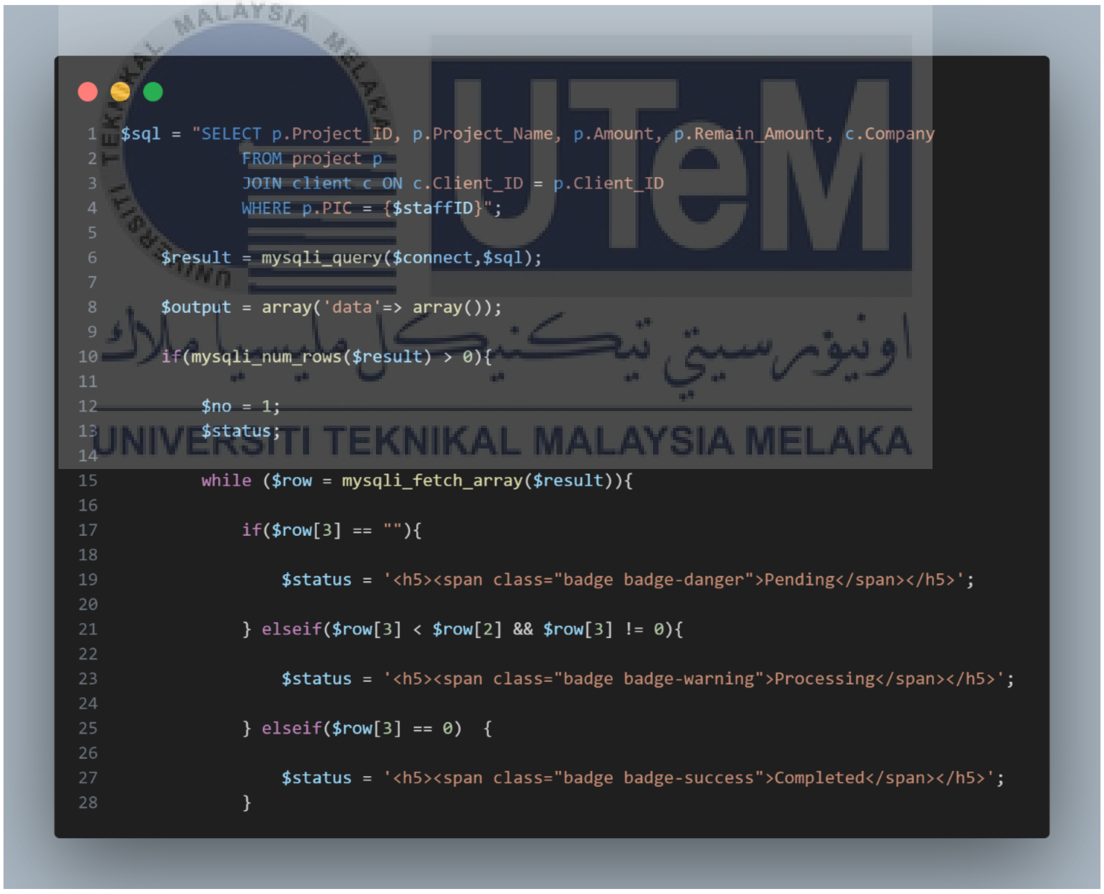
Test design for PTMS is a systematic and strategic process aimed at creating detailed test scenarios and cases to assess the diverse functionalities of the system. It involves defining clear testing objectives, identifying critical system features, and outlining specific conditions for evaluation. The two pivotal components, test description and test data, play a crucial role in guiding the testing team toward comprehensive coverage, ensuring the system meets specified requirements and performs robustly under various scenarios.

6.4.1 Test Description

The test description elucidates the identification of test cases, testing types, pre-conditions, test requirements, step-by-step procedures, and expected output results. Each module's test case is methodically designed and documented.

6.4.1.1 White Box Testing

White box testing is tested by the developer system through every code to test and debug for each error such as syntax error, logical error, compilation error and etc. The white box test has a few techniques to test the code. In the testing phase, techniques that are used to test the code such as statement coverage, and path coverage as shown in Table 6.5 and Table 6.6.



```
 1  $sql = "SELECT p.Project_ID, p.Project_Name, p.Amount, p.Remain_Amount, c.Company
 2              FROM project p
 3              JOIN client c ON c.Client_ID = p.Client_ID
 4              WHERE p.PIC = {$staffID}";
 5
 6      $result = mysqli_query($connect,$sql);
 7
 8      $output = array('data'=> array());
 9
10      if(mysqli_num_rows($result) > 0){
11
12          $no = 1;
13          $status;
14
15          while ($row = mysqli_fetch_array($result)){
16
17              if($row[3] == ""){
18
19                  $status = '<h5><span class="badge badge-danger">Pending</span></h5>';
20
21              } elseif($row[3] < $row[2] && $row[3] != 0){
22
23                  $status = '<h5><span class="badge badge-warning">Processing</span></h5>';
24
25              } elseif($row[3] == 0)  {
26
27                  $status = '<h5><span class="badge badge-success">Completed</span></h5>';
28              }
```

Figure 6.2 Testing Code

Based on Figure 6.2 testing code, on line 17 until line 28 to check the project payment status through the remaining amount of project whether the project payment is pending (not make any payment), processing (has remain amount) and completed.

- **Statement Coverage**

  Statement Coverage testing is a test case design method concentrated on executing every executable statement within the source code at least once. This approach ensures comprehensive coverage of all lines, statements, and paths in the software or application's source code [1]. Statement Coverage = (Number of statements executed/ Total number of statements in the code) * 100

Table 6.5 Statement Coverage Table

Test ID	TID_1 – Statement Coverage		
Testing Strategy	White Box		
Test Case ID	**Input**	**Description**	**Statement Coverage**
TSC_1	Let's assume there is a query call project that has amount and remaining amount. $row[3] = null $row[2]=24000	On line 17 will check is $row[3] equal to empty or null if right the statement will print the inside of condition.	Statement coverage = (17/28) *100 = 60.72%
TSC_2	$row[3] = 2300 $row[2]=24000	The statement will check next condition which is $row[3] less then $row[2] and not equal to 0.	Statement coverage = (23/28) *100 = 82.14%

- **Path Coverage**

  Path coverage, as a test case design technique, thoroughly examines all pathways within the code. This robust strategy ensures that every program route or path undergoes testing at least once. The effectiveness and breadth of coverage in this technique surpass that of multiple-condition coverage [1].

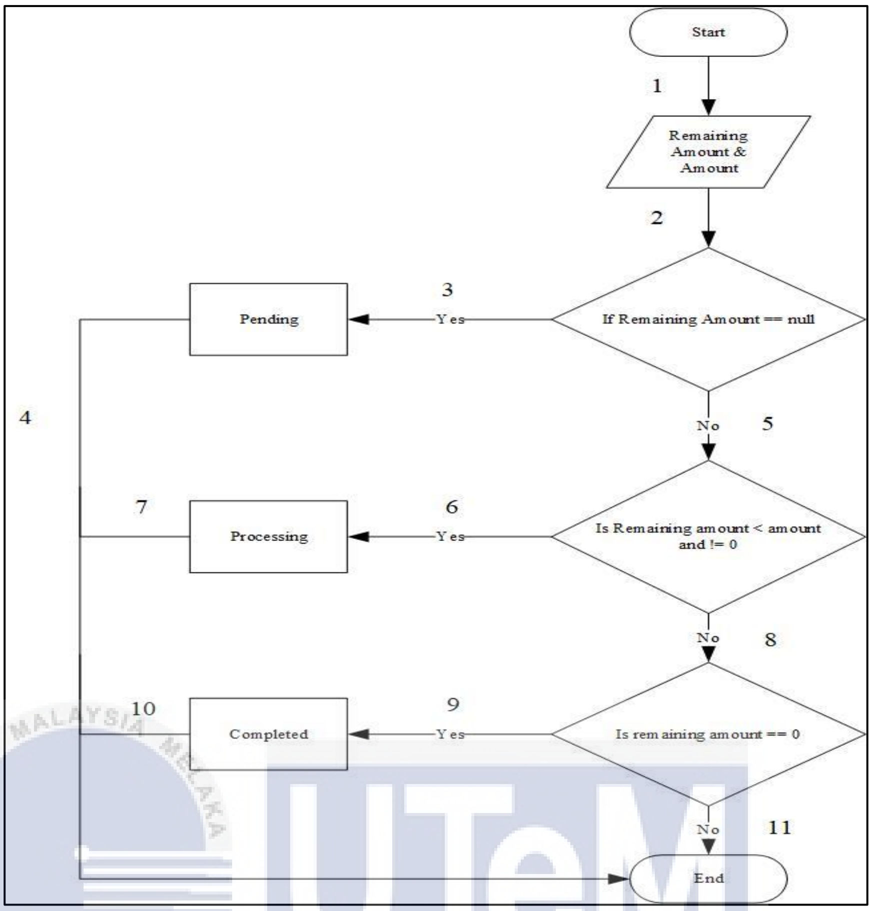

Start
1
Remaining Amount & Amount
2
If Remaining Amount == null
3
Yes
Pending
4
No
5
Is Remaining amount < amount and != 0
6
Yes
Processing
7
No
8
Is remaining amount == 0
9
Yes
Completed
10
No
11
End

Figure 6.3 Flowchart for Path Coverage

Based on Figure 6.3 flowchart for Figure 6.2 Testing Code from line 17 to line 18. The Figure shown as below to testing using technique Path Coverage as show in Table 6.6

Table 6.6 Path Coverage Table

Test ID	TID_2 – Path Coverage		
Testing Strategy	White Box		
Test Case ID	**Input**	**Expected Output**	**Path Taken**
TPC_1	Remaining amount = null	Pending	1 > 2 > 3 > 4
TPC_2	Remaining amount = 3500 and amount = 24000	Processing	1 > 2 > 5 > 6 > 7
TPC_3	Remaining amount = 0	Completed	1 > 2 > 5 > 8 > 9 > 10

6.4.1.2 Black Box Testing

Black box testing is tested by all end users such as Project Manager and Staff Member. In Table 6.7 until Table 6.12 is showing result during testing phase.

Table 6.7 Login Page (Project Manager)

Test ID	T001 – Login			
Testing Type	Unit testing and integration testing			
Test Strategy	Black Box Testing			
Test Class	Security and error handling testing			
Test Case ID	**Test Requirements**	**Pre-Condition**	**Step Procedure**	**Expected Output**
TC1_1	Validate the login function is working if the identity card number and password provided that has been recorded in database.	User has valid identity card number and password	1. Navigate to login page. 2. Provide a valid identity card number. 3. Provide a valid password. 4. Click on login button	Login Successful
TC1_2	Validate the login function if the input of identity card number and password are empty.		1. Navigate to login page. 2. Click on login button	Login failed display error message “Identity No. and password required”
TC1_3	Validate the login function if the input field is wrong and not match.	User has field wrong identity no. and password	1. Navigate to login page. 2. Provide valid identity no. 3. Provide invalid password. 4. Click on login button.	Login failed display error message “Incorrect identity no. or password”

Table 6.8 Manage Staff

Test ID	T002 – Manage Staff		
Testing Type	Unit testing and error handling testing		
Test Strategy	Black Box Testing		
Pre-Condition	User must login as role project manager		
Test Case ID	**Test Requirements**	**Step Procedure**	**Expected Output**
TC2_1	Validate the register staff function is available if all the input data are valid.	1. Navigate to add staff page. 2. Enter correct data to input text field. 3. Click on save button.	Insert a new staff successful and display success message "Successful add new staff"
TC2_2	Validate the staff registration function with empty data that mandatory to fill into the input field	1. Navigate to add staff page. 2. Click on save button.	Registration staff failed. Display error message.
TC2_3	Validate the registration staff function is not available if the certain data format is invalid such as identity card number.	1. Navigate to registration staff page. 2. Enter wrong format identity card number. Example less than 12 character.	The registration staff failed. Display error message "identity card number must be 12 characters"
TC2_4	Validate the update staff function is working	1. Navigate to manage staff page. 2. Choose one staff that has registered and click the button action edit. 3. Update the position staff. 4. Click save button.	The update successful and display success message.
TC2_5	Validate the delete staff function is working	1. Navigate to manage staff page. 2. Choose one member of staff that has registered. Click icon trash. 3. Click the confirmation.	The deletion is successfully.

Table 6.9 Manage Project

Test ID	T003 – Manage Project		
Testing Type	Unit testing and error handling testing		
Test Strategy	Black Box Testing		
Pre-Condition	User must login as role project manager		
Test Case ID	**Test Requirements**	**Step Procedure**	**Expected Output**
TC3_1	Validate add project function is working if all the input data are valid.	1. Navigate to add project page. 2. Enter correct data align with input field. 3. Click add project button.	Insert new project has successful and display success message "Successful add project"
TC3_2	Validate add project function is working if all the input data are empty.	1. Navigate to add project page. 2. Click add project button.	Inserting new project failed.
TC3_3	Validate add project function is working if certain input data is invalid such as end date is less than start date.	1. Navigate to add project page. 2. Enter all input text with correct data. 3. Select end date less than start date. 4. Click add project button.	Inserting new project failed and display error message "End Date must be Future from start date".
TC3_4	Validate update project function is available.	1. Navigate to manage project. 2. Choose a project that has success add. 3. Click button edit. Change the data by fill input field. 4. Click save button.	Updating project is successfully and display message.
TC3_5	Validate update project function is not available with invalid data.	1. Navigate to manage project. 2. Choose a project that has success add. 5. Click button edit. Select end date less than start date. 3. Click save button.	Updating project failed and display error message "End Date must be Future from start date".

TC3_6	Validate delete project function is available.	1. Navigate to manage project. 2. Choose a project that has success add. 3. Click button trash. 4. Click the confirmation	Deleting project is successfully

Table 6.10 Manage Task

Test ID	T004 – Manage Task		
Testing Type	Unit testing, Error handling and Integration system testing		
Test Strategy	Black Box Testing		
Pre-Condition	1. User must login as role project manager 2. The project must already be added to the system. 3. The registration staff must be done.		
Test Case ID	**Test Requirements**	**Step Procedure**	**Expected Output**
TC4_1	Validate add task in project function is working if all the input data are valid.	1. Navigate to manage project. 2. Click on the action button and click task project. 3. Click on the add task button. 4. Fill the form with correct data. 5. Click save button.	Inserting new task in project is successfully.
TC4_2	Validate add task in project function is working if all the input data are empty.	1. Navigate to manage project. 2. Click on the action button and click task project. 3. Click on the add task button. 4. Click save button.	Inserting new tasks in a project is unsuccessful. Display error message to fill the form.
TC4_3	Validate add task in project function is working if certain input data is invalid such as end date is less than start date.	1. Navigate to manage project. 2. Click on the action button and click task project. 3. Enter all input text with correct data. 4. Select end date less than start date. 5. Click add project button.	Inserting new tasks in project failed and display error message "End Date must be Future from start date".

TC4_4	Validate assign staff to task function.	1. Navigate to manage project. 2. Click on the action button and click task project. 3. Enter all input text with correct data. 4. Pick the staff from the select element. 5. Click save button.	The new task and assign staff to the staff successful and display message successful.
TC4_5	Validate update task in project function is available.	1. Navigate to manage project. 2. Click on the action button and click task project. 3. Choose a task that has success added. 4. Click button edit. Change the data by fill input field. 5. Click save button.	Updating task in project is successfully and display message.
TC4_6	Validate update task in project function is not available with invalid data.	1. Navigate to manage project. 2. Click on the action button and click task project. 3. Choose a task that has success added. 4. Click button edit. Select end date less than start date. 5. Click save button.	Updating task failed and display error message "End Date must be Future from start date".
TC4_7	Validate delete task in project function is available.	1. Navigate to manage project. 2. Click on the action button and click task project. 3. Choose a task that has success added. 4. Click the trash button. 5. Click the confirmation.	Deleting task in project is successfully

Table 6.11 Take Action Task

Test ID	T005 – Take Action task		
Testing Type	Unit testing, Error handling and Integration system testing		
Test Strategy	Black Box Testing		
Pre-Condition	1. User must login as role staff 2. The task must be assigned to staff.		
Test Case ID	**Test Requirements**	**Step Procedure**	**Expected Output**
TC5_1	Validate take action on task that assign from project manager.	1. Navigate to manage project. 2. Click one of project that assign. 3. Click one of task that assign. 4. Pick the status and remark if any on the task that has been assigned. 5. Click save button.	Updating on task that has been assign has successfully and display success message.

Table 6.12 Dashboard and Reporting

Test ID	T006 – Dashboard and Reporting		
Testing Type	User Acceptance Testing (UAT)		
Test Strategy	Black Box Testing		
Pre-Condition	1. User must be login. 2. The project status must be updated. 3. Add new task in project.		
Test Case ID	**Test Requirements**	**Step Procedure**	**Expected Output**
TC6_1	Validate the dashboard and reporting function is working if have a project and task on project.	1. Navigate to manage project. 2. Click on the action button and click task project. 3. Choose a task that has success added. 4. Click button edit. Update the status task. 5. Click the save button. 6. Navigate to dashboard page.	The dashboard will display successfully all the project and task.

6.4.2 Test Data

Test data is a critical component in evaluating the functionality of the Project Task Management System (PTMS). It involves the systematic generation and utilization of input values to validate the system's responses and overall performance. This section outlines the strategies and datasets employed for comprehensive test scenarios within PTMS.

Table 6.13 Test Data for Login (Manager)

Test Data ID	Identity Card No.	Password
TD1_1	990425104593	Abcd@123
TD1_2	Null	Null
TD1_3	990425104593	1234aAb

Table 6.14 Test Data for Manage Staff

Test Data ID	Name	No_IC	Gender	Position
TD2_1	Muhammad Afiq	990425104593	Male	Supervisor
TD2_2	Null	Null	Null	Null
TD2_3	Muhammad Afiq	99042510	Male	Supervisor
TD2_4	Null	Null	Null	Staff
TD2_5	-	-	-	-

Table 6.15 Test Data for Manage Project

Test Data ID	Project_Name	Start_Date	Due_Date	Amount	PIC	Client ID
TD3_1	E-Kehadiran	2023-10-03	2024-04-12	25000	1	2
TD3_2	Null	Null	Null	Null	Null	Null

Test Data ID	Project_Name	Start_Date	Due_Date	Amount	PIC	Client ID
TD3_3	E-Kehadiran	2024-10-03	2023-04-12	25000	1	2
TD3_4	E-Kehadiran	2023-10-03	2024-04-12	-	-	-
TD3_5	E-Kehadiran	2024-10-03	2023-04-12	-	-	-
TD3_6	-	-	-	-	-	-

Table 6.16 Test Data for Manage Task

Test Data ID	Task	Start Date	End Date	Desc	Status
TD4_1	Print	2024-10-03	2023-04-12	10 copies	1
TD4_2	Null	Null	Null	Null	Null
TD4_3	Print	2023-10-03	2024-04-12	-	-
TD4_5	Develop php	2024-10-03	2023-04-12	Null	1
TD4_6	Document SRS	2023-10-03	2024-04-12	-	-
TD4_7	-	-	-	-	-

Table 6.17 Test Data for Assign Staff

Test Data ID	Task_ID	Staff_ID
TD4_4	1	7

Table 6.18 Test Data for Take Action Task

Test Data ID	Task_ID	Staff_ID	Status_ID
TD5_1	1	7	3

6.5 Test Result & Analysis

Test results and analysis to test and get the output by matching the test data and test case scenario.

Table 6.19 Result for Login (Manager)

Test Case ID	Test Data ID	Expected Output	Actual Output	Result
TC1_1	TD1_1	Login Successful	Directly to main page which dashboard page	Pass
TC1_2	TD1_2	Login failed display error message “Identity No. and password required”	Display error message “Identity No. and password required”	Pass
TC1_3	TD1_3	Login failed display error message “Incorrect identity no. or password”	Display error message “Incorrect identity no. or password”	Pass

Table 6.20 Result for Manage Staff

Test Case ID	Test Data ID	Expected Output	Actual Output	Result
TC2_1	TD2_1	Insert a new staff successful and display success message “Successful add new staff”	Display message “Successful Add Staff”	Pass
TC2_2	TD2_2	Registration staff failed. Display error message.	The input field will turn to danger color and display message to fill the input text	Pass

TC2_3	TD2_3	The registration staff failed. Display error message “Insert Identity No. Properly”.	Display error message “Insert Identity No. Properly”	Pass
TC2_4	TD2_4	The update successful and display success message.	Display message “Successful Update staff”	Pass
TC2_5	TD2_5	The deletion is successfully.	Display message “successful remove staff”	Pass

Table 6.21 Result for Manage Project

Test Case ID	Test Data ID	Expected Output	Actual Output	Result
TC3_1	TD3_1	Insert new project has successful and display success message “Successful add project”	Display message “Successful Add Project”	Pass
TC3_2	TD3_2	Inserting new project failed.	The input field will turn to danger color and display message to fill the input text	Pass
TC3_3	TD3_3	Inserting new project failed and display error message “End Date must be Future from start date”.	Display error message “End date must be future from Start Date”	Pass
TC3_4	TD3_4	Updating project is successfully and display message.	Display message “successful update project”	Pass
TC3_5	TD3_5	Updating project failed and display error message “End Date must be Future from start date”.	Display error message “End date must be future from Start Date”	Pass
TC3_6	TD3_6	Deleting project is successfully	Display message “successful delete project”	Pass

Table 6.22 Result for Manage Task

Test Case ID	Test Data ID	Expected Output	Actual Output	Result
TC4_1	TD4_1	Inserting new task in project is successfully.	Display message “Successful New Task”	Pass
TC4_2	TD4_2	Inserting new tasks in a project is unsuccessful. Display error message to fill the form.	The input field will turn to danger color and display message to fill the input text	Pass
TC4_3	TD4_3	Inserting new tasks in project failed and display error message “End Date must be Future from start date”.	Display error message “End date must be future from Start Date”	Pass
TC4_4	TD4_4	The new task and assign staff to the staff successful and display message successful.	Display message “Successful update task and staff”	Pass
TC4_5	TD4_5	Updating task in project is successfully and display message.	Display message “successful update task”	Pass
TC4_6	TD4_6	Updating task failed and display error message “End Date must be Future from start date”.	Display error message “End date must be future from Start Date”	Pass
TC4_7	TD4_7	Deleting task in project is successfully	Display message “successful delete task”	Pass

Table 6.23 Result for Take Action Task

Test Case ID	Test Data ID	Expected Output	Actual Output	Result
TC5_1	TD5_1	Updating on task that has been assign has successfully and display success message.	Display message “Successful Update Task”	Pass

6.6 Conclusion

In conclusion, the testing phase for the Project Task Management System (PTMS) has played a pivotal role in ensuring the system's functional integrity, user-centric performance, and overall optimization. Rigorous test scenarios and meticulous designs have scrutinized every aspect of PTMS's functionalities, detecting and rectifying potential defects to guarantee seamless operation. The testing process prioritized delivering an intuitive and satisfying user experience, simulating real-world usage scenarios to validate the system's responsiveness. Beyond defect identification, testing has contributed significantly to system optimization, addressing performance bottlenecks and scalability concerns. As PTMS progresses towards implementation, the insights gleaned from testing not only ensure the system's current reliability but also lay the groundwork for scalability, adaptability, and excellence within the academic and industrial domains.

CHAPTER 7: PROJECT CONCLUSION

7.1 Introduction

In this dedicated section, we embark on a thorough exploration of the conclusive aspects of the Project Task Management System (PTMS). Herein, we conduct a meticulous analysis of the system, scrutinizing both its formidable strengths and areas where improvement is warranted. Beyond a mere examination, we actively put forth proposals for augmentations grounded in the insights derived from this evaluative process. Furthermore, we shine a light on the notable contributions made to the advancement and refinement of this project.

In the subsequent paragraphs, we intricately dissect the robust elements that fortify the Project Task Management System (PTMS) and bring attention to the aspects where refinement is essential. By leveraging the findings of this analysis, we strategically propose enhancements designed to elevate the system's efficacy. These enhancements are not arbitrary but are meticulously tailored to address specific weaknesses identified during the evaluation. Moreover, we illuminate the ways in which these proposed improvements contribute to the overarching goals and objectives of the project.

Beyond a mere discussion of strengths and weaknesses, we proactively lay out a roadmap for the evolution of the Project Task Management System (PTMS). Our proposed enhancements are not only pragmatic but are aligned with the core objectives of optimizing task management and project coordination. We delve into the practicalities of implementing these improvements, emphasizing their potential impact on streamlining processes, enhancing user experience, and fortifying the overall efficacy of the system.

7.2 Observations on weaknesses and strengths

In the continual pursuit of optimizing operational efficiency and refining project management strategies, an insightful examination of the strengths and weaknesses of the Project Task Management System (PTMS) becomes imperative. This analysis serves as a compass, guiding our endeavors toward harnessing existing strengths for sustained excellence and addressing identified weaknesses for targeted improvements. Through astute observations, we aim to fortify the robust facets of the system while strategically navigating challenges, ultimately steering the trajectory of PTMS towards heightened efficacy and user satisfaction. This exploration into the system's strengths and weaknesses forms the cornerstone of our commitment to delivering a project management tool that not only meets but exceeds the evolving needs of our users and stakeholders.

7.2.1 Strengths

In the relentless pursuit of operational excellence and enhanced project management, PTMS exhibits several commendable strengths. Its centralized task management hub provides real-time visibility, fostering transparency and accountability. The system's adaptability to agile methodologies empowers users to navigate dynamic project requirements effortlessly, ensuring flexibility in project execution. Furthermore, PTMS facilitates seamless collaboration between academia and industry, bridging the gap between theoretical knowledge and real-world practices. This adaptability and integration capability have streamlined industry-oriented projects, establishing a dynamic connection between UTeM and industry practices. Overall, PTMS stands strong in delivering transparency, adaptability, and effective collaboration in project management.

7.2.2 Weaknesses

While PTMS boasts notable strengths, certain weaknesses warrant attention for refinement. The system, in some instances, may encounter challenges in handling a vast volume of data, necessitating optimization for scalability. Additionally, user interface intricacies may pose a hurdle for seamless navigation, requiring enhancements for a more intuitive experience. Continuous efforts are essential to address these weaknesses, ensuring PTMS aligns seamlessly with user expectations and offers an enhanced, user-friendly interface.

7.3 Propositions for Improvement

i. **Backup and Recovery**

Enhancing the system involves incorporating an automated backup feature. This entails executing automatic backups daily at 12:00 AM to maintain the currency of backup data. The recruiter manager retains control over the backup frequency, adjusting it as needed—whether daily, weekly, or monthly—aligned with system requirements. In the event of system corruption, the system seamlessly initiates an automatic recovery process.

ii. **Administrative User**

The admin, often referred to as the system administrator, holds authority over the system's configuration, user management, and overall functionality. Key responsibilities include creating and managing user accounts, defining access permissions, and ensuring the system's smooth operation. The admin also addresses technical issues, implements system updates, and oversees security measures. This central role empowers the admin to maintain the integrity and efficiency of PTMS, fostering effective project management within the organization. Admin has the capability to perform CRUD (Create, Read, Update, Delete) operations on the payment module. This involves managing financial transactions, creating new payment entries, updating existing records, retrieving payment information, and handling deletions when necessary. The admin's multifaceted role extends to maintaining financial data integrity and ensuring the seamless operation of the payment module within PTMS.

7.4 Project Contributions

The Project Task Management System (PTMS) implemented at UTeM's Faculty of Information and Communication Technology has been a transformative force, contributing significantly to the university's project management landscape. The system's introduction has enhanced task visibility and coordination among faculty members, fostering a collaborative environment for more efficient project execution. PTMS's centralized hub for task management ensures that academic projects are seamlessly tracked, facilitating streamlined communication, and promoting a shared understanding of project progress.

Moreover, within the academic context, PTMS has played a crucial role in supporting agile project management methodologies. Its adaptability to changes, iterative processes, and evolving project requirements aligns with the dynamic nature of academic projects. This contribution ensures that faculty members can easily navigate project complexities, promoting a more agile and responsive approach to academic project management.

In collaboration with SE Infinity Sdn Bhd, the implementation of PTMS has extended beyond academia, creating a symbiotic relationship between industry and education. The system's adaptability and integration capabilities have provided a seamless platform for faculty members and students to collaborate with SE Infinity on industry-oriented projects. This dynamic connection between UTeM and SE Infinity underscores PTMS's role in bridging academic knowledge with real-world industry practices, contributing to a holistic and experiential learning environment for students at the Faculty of Information and Communication Technology.

7.5 Conclusions

The implementation of the Project Task Management System (PTMS) at University Technical Malaysia Melaka (UTeM), Faculty of Information and Communication Technology, marks the successful realization of carefully articulated objectives aimed at advancing project management capabilities. The project's primary goals were to enhance task visibility, streamline collaboration, and optimize resource utilization in both academic and industry contexts.

Achieving the initial objective, PTMS has notably enhanced task visibility in the academic domain. The system's centralized task management hub has provided faculty members with a seamless platform for monitoring and tracking project progress. This advancement fosters transparency and accountability, aligning with the overarching goal of fostering a more informed and collaborative academic environment. The system's adaptability to agile project management methodologies is a standout accomplishment, empowering faculty members and students to navigate dynamic project requirements effortlessly. This achievement is especially crucial in the constantly evolving landscapes of academia and industry, where adaptability plays a paramount role.

Furthermore, PTMS has efficiently fostered collaboration between academia and industry, narrowing the divide between theoretical knowledge and real-world applications. In collaboration with SE Infinity Sdn Bhd, the system's adaptability and integration capabilities have streamlined engagement in industry-oriented projects, establishing a dynamic connection between UTeM and industry practices. Beyond achieving specific objectives, PTMS has left a lasting imprint on project management practices at UTeM and its collaborations with SE Infinity. The system's streamlined processes and improved project outcomes have contributed to a culture of innovation, collaboration, and experiential learning in both academic and industrial spheres. In conclusion, the successful implementation of PTMS not only met its defined objectives but has also laid the foundation for a more efficient, collaborative, and adaptive approach to project management. As UTeM continues to embrace this technological advancement, the enduring impact of PTMS is set to shape a future where project management excellence is synonymous with success in academia and industry alike.

REFERENCES

1. Deo, T. K. (2023, November 16). Test case design techniques for smart software testing | LambdaTest. LambdaTest. https://www.lambdatest.com/blog/test-case-design-techniques/

2. Wikis Engrade. (2016). Dbase Initial System. Retrieved on March 2019 from https://wikis.engrade.com/databaselifecycledblc/dbase-initial-study

3. Computers Professor. (2019). Explain Database Life Cycle DBLC. Retrieved on March 2019 from http://www.computersprofessor.com/2019/01/explain-database-life-cycle-dblc.html

4. Smart Draw. (2017). Data Flow Diagram. Retrieved March 2019 from https://www.smartdraw.com/data-flow-diagram/

5. Visual Paradigm. (2012). Data Flow Diagram. Retrieved March 2019 from https://www.visual-paradigm.com/tutorials/data-flow-diagram-dfd.jsp

6. Tutorialspoint.com. (2011.). Software Testing Quick Guide. Retrieved from https://www.tutorialspoint.com/software_testing/software_testing_quick_guide

7. Difference between Black Box and White Box Testing: Practice: GeeksforGeeks. (n.d.). Retrieved from https://practice.geeksforgeeks.org/problems/difference-between-black-box-and-white-box-testing

APPENDIX A

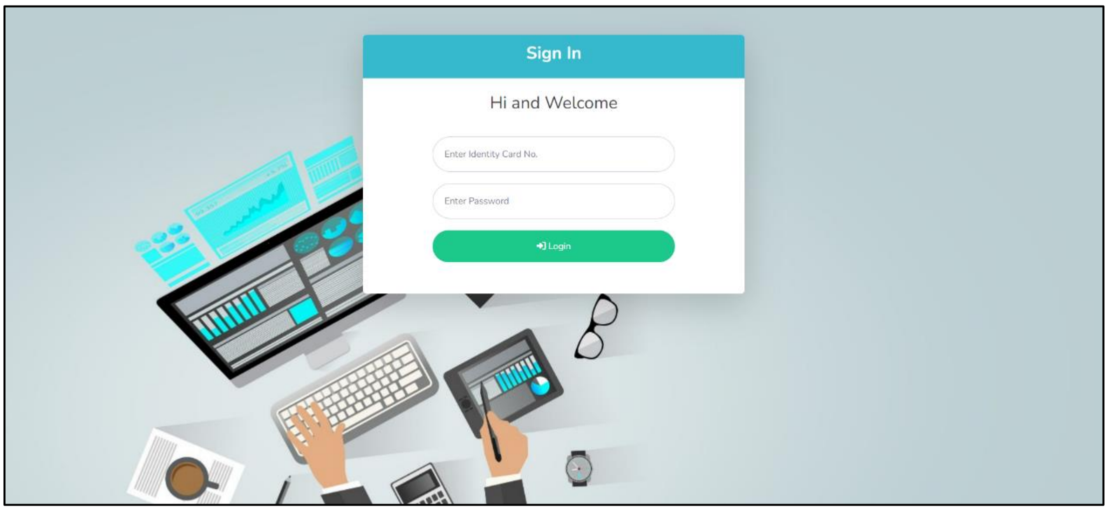


Figure A.1 Login Page

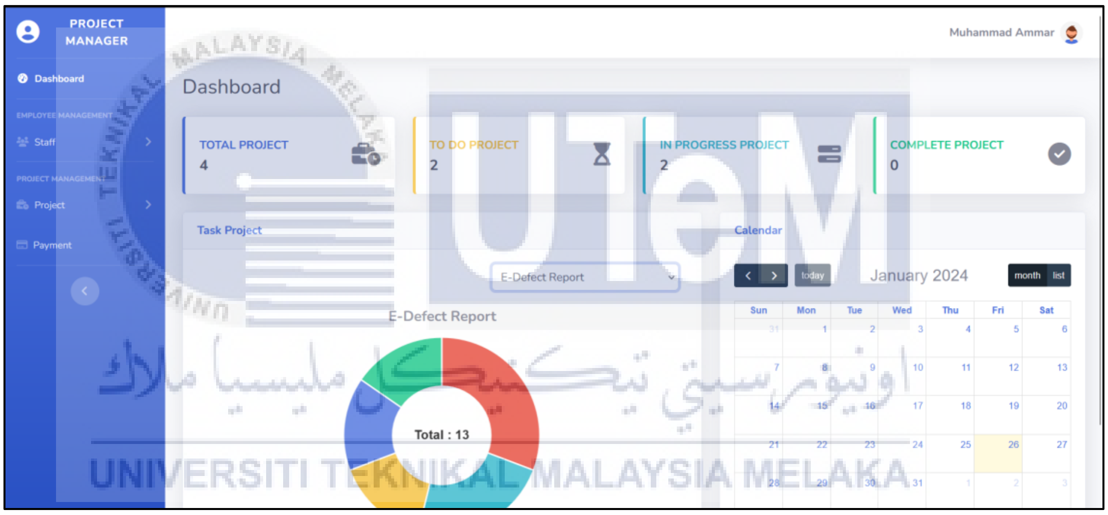


Figure A.2 Dashboard Page (Manager)

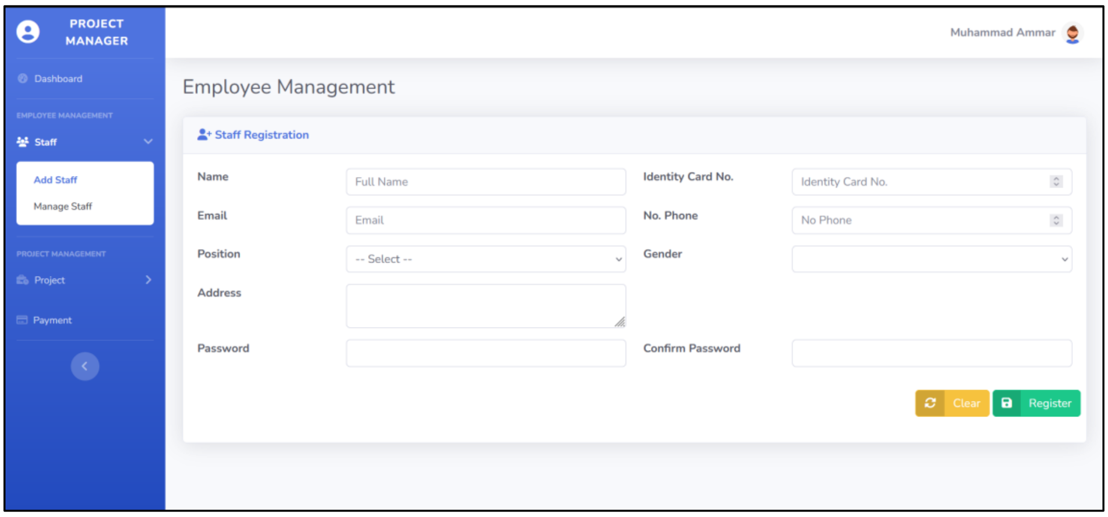
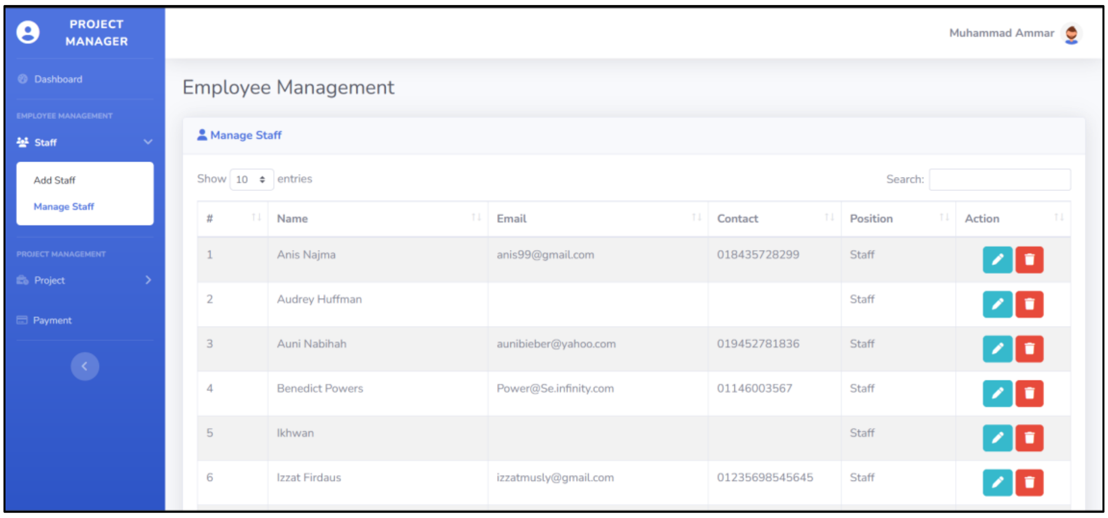


Figure A.3 Staff Registration Page



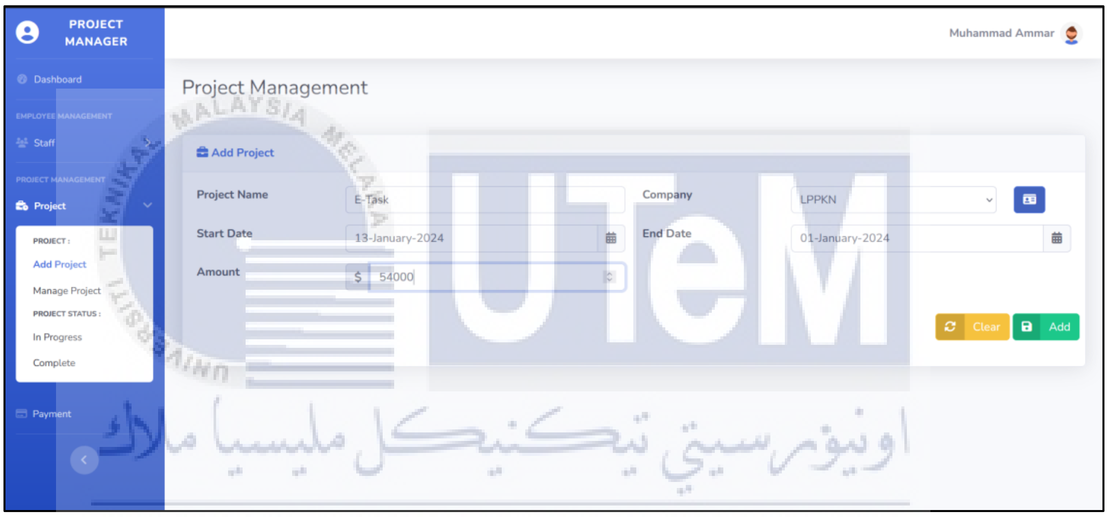
Employee Management

Manage Staff

Show 10 entries Search:

#	Name	Email	Contact	Position	Action
1	Anis Najma	anis99@gmail.com	018435728299	Staff	
2	Audrey Huffman			Staff	
3	Auni Nabihah	aunibieber@yahoo.com	019452781836	Staff	
4	Benedict Powers	Power@Se.infinity.com	01146003567	Staff	
5	Ikhwan			Staff	
6	Izzat Firdaus	izzatmusly@gmail.com	01235698545645	Staff	

Figure A.4 Manage Staff Page

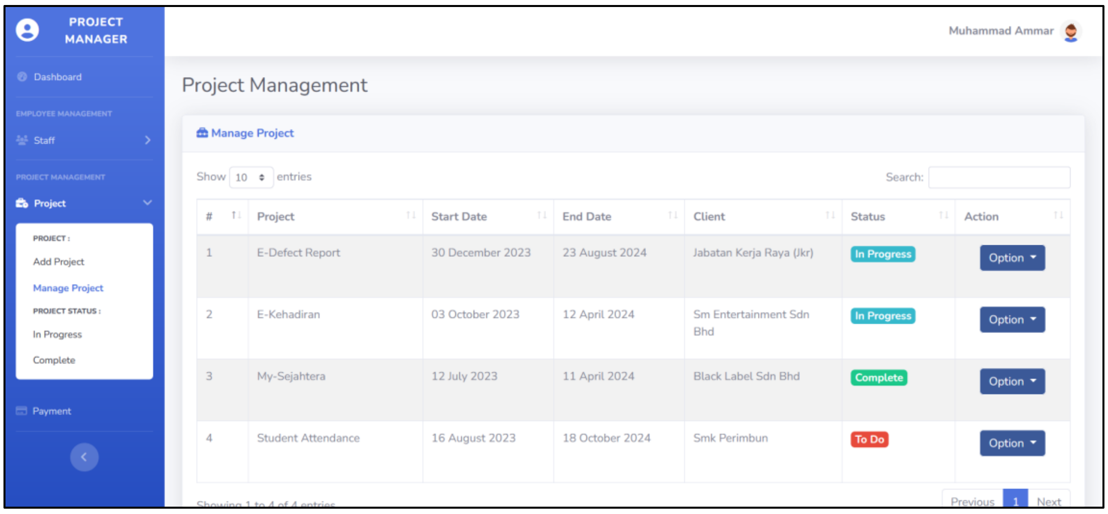


Project Management

Add Project

Project Name	E-Task	Company	LPPKN
Start Date	13-January-2024	End Date	01-January-2024
Amount	$ 54000		

Figure A.5 Add Project Page



Project Management

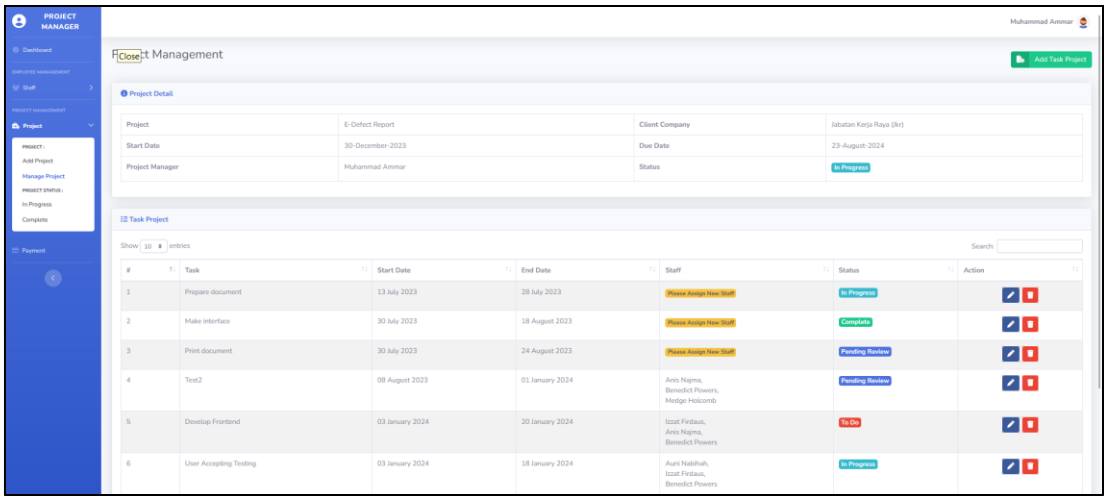
Manage Project

Show 10 entries Search:

#	Project	Start Date	End Date	Client	Status	Action
1	E-Defect Report	30 December 2023	23 August 2024	Jabatan Kerja Raya (Jkr)	In Progress	Option
2	E-Kehadiran	03 October 2023	12 April 2024	Sm Entertainment Sdn Bhd	In Progress	Option
3	My-Sejahtera	12 July 2023	11 April 2024	Black Label Sdn Bhd	Complete	Option
4	Student Attendance	16 August 2023	18 October 2024	Smk Perimbun	To Do	Option

Showing 1 to 4 of 4 entries Previous 1 Next

Figure A.6 Manage Project Page



Project Management

Project Detail

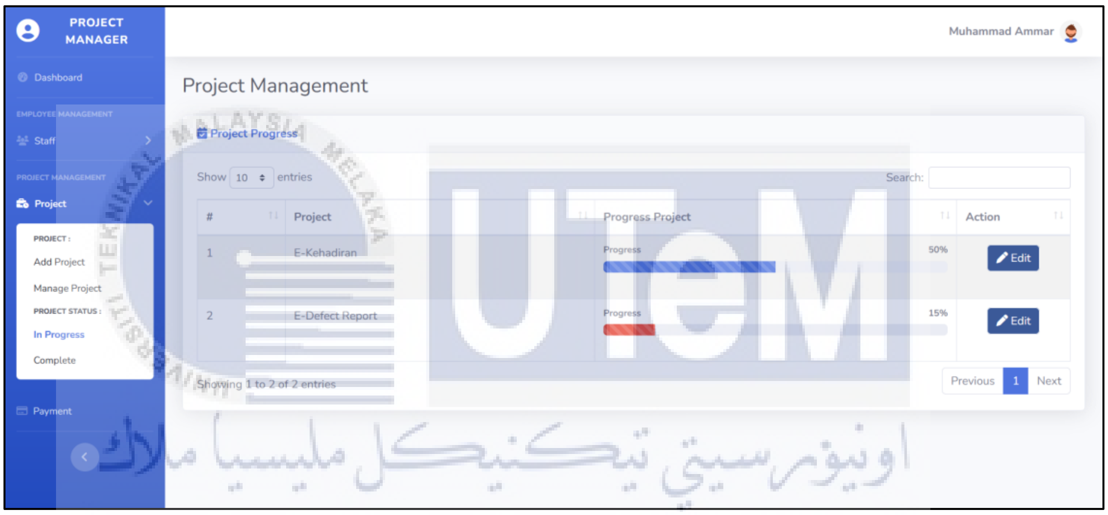
Project	E-Defect Report	Client Company	Jabatan Kerja Raya (Jkr)
Start Date	30-December-2023	Due Date	23-August-2024
Project Manager	Muhammad Ammar	Status	In Progress

Task Project

Show 10 entries Search:

#	Task	Start Date	End Date	Staff	Status	Action
1	Prepare document	13 July 2023	28 July 2023	Please Assign New Staff	In Progress	
2	Make interface	30 July 2023	18 August 2023	Please Assign New Staff	Complete	
3	Print document	30 July 2023	24 August 2023	Please Assign New Staff	Pending Review	
4	Test2	08 August 2023	01 January 2024	Anis Najma, Benedict Powers, Medge Holcomb	Pending Review	
5	Develop Frontend	03 January 2024	20 January 2024	Izzat Firdaus, Anis Najma, Benedict Powers	To Do	
6	User Accepting Testing	03 January 2024	18 January 2024	Auni Nabihah, Izzat Firdaus, Benedict Powers	In Progress	

Figure A.7 Manage Task Page



Project Management

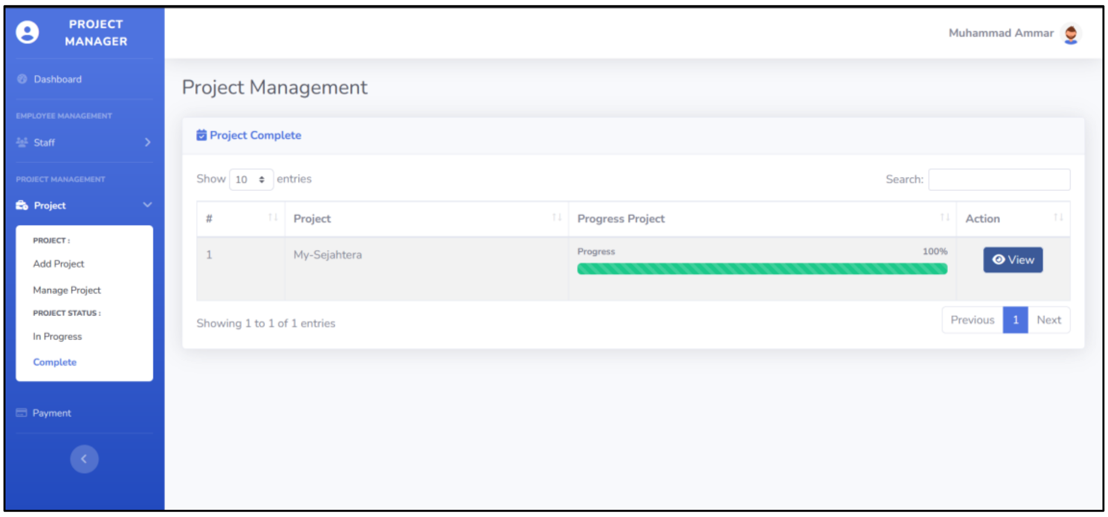
Project Progress

Show 10 entries Search:

#	Project	Progress Project	Action
1	E-Kehadiran	Progress 50%	Edit
2	E-Defect Report	Progress 15%	Edit

Showing 1 to 2 of 2 entries Previous 1 Next

Figure A.8 In Progress Page



Project Management

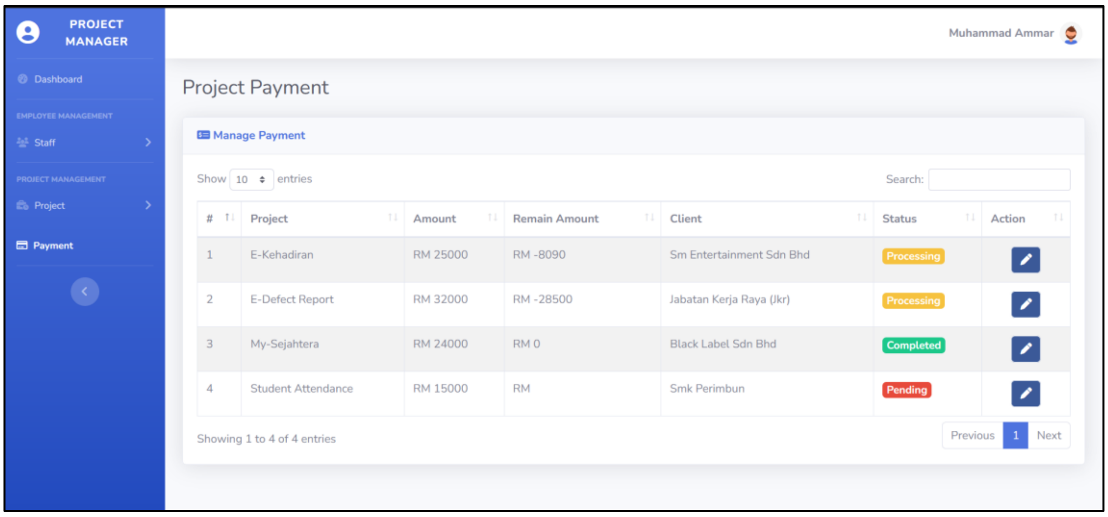
Project Complete

Show 10 entries Search:

#	Project	Progress Project	Action
1	My-Sejahtera	Progress 100%	View

Showing 1 to 1 of 1 entries Previous 1 Next

Figure A.9 Complete Project



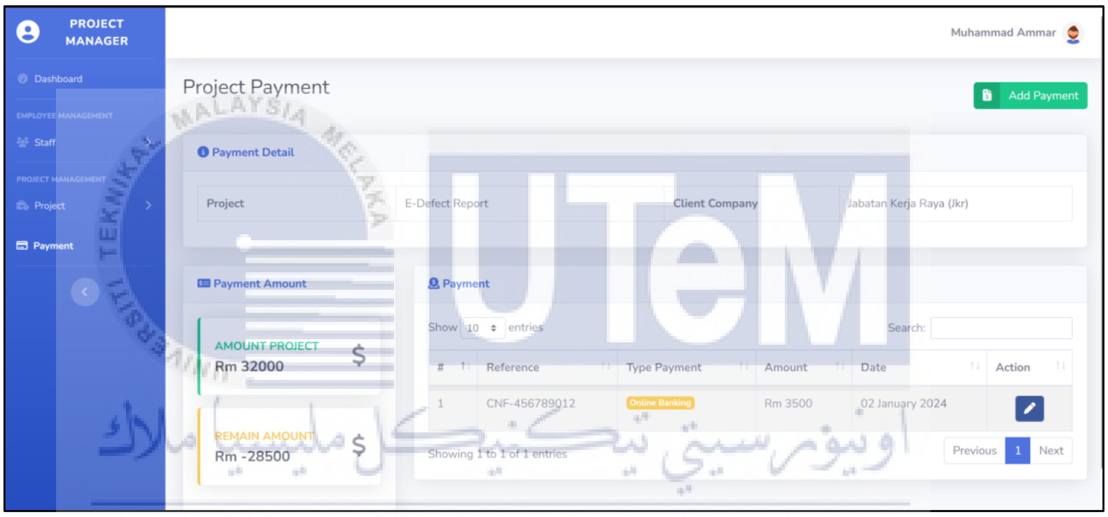
Project Payment

Manage Payment

#	Project	Amount	Remain Amount	Client	Status	Action
1	E-Kehadiran	RM 25000	RM -8090	Sm Entertainment Sdn Bhd	Processing	
2	E-Defect Report	RM 32000	RM -28500	Jabatan Kerja Raya (Jkr)	Processing	
3	My-Sejahtera	RM 24000	RM 0	Black Label Sdn Bhd	Completed	
4	Student Attendance	RM 15000	RM	Smk Perimbun	Pending	

Showing 1 to 4 of 4 entries

Figure A.10 Payment Page



Project Payment

Payment Detail

Project	E-Defect Report	Client Company	Jabatan Kerja Raya (Jkr)

Payment Amount

AMOUNT PROJECT
Rm 32000

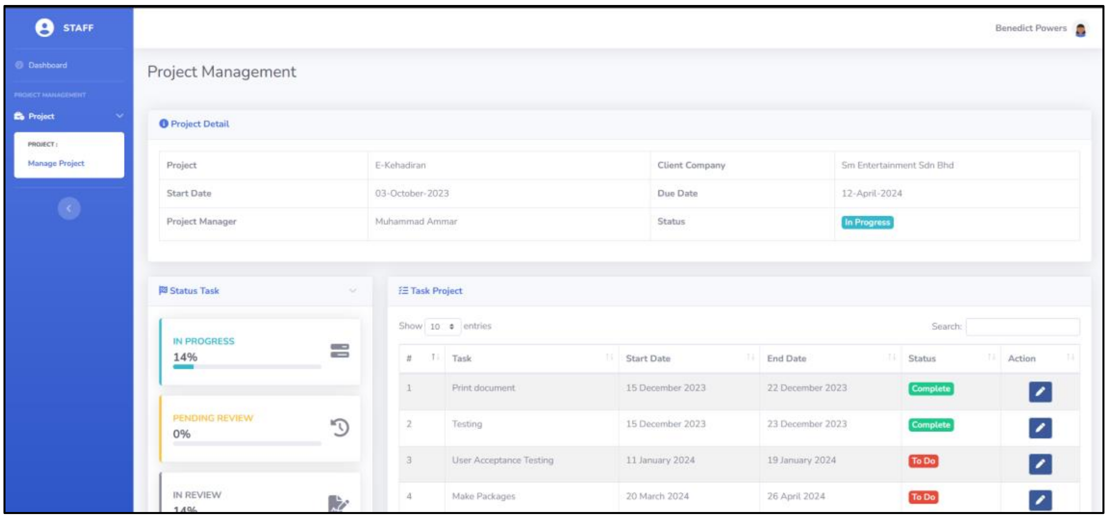
REMAIN AMOUNT
Rm -28500

Payment

#	Reference	Type Payment	Amount	Date	Action
1	CNF-456789012	Online Banking	Rm 3500	02 January 2024	

Showing 1 to 1 of 1 entries

Figure A.11 Payment Details Page



Project Management

Project Detail

Project	E-Kehadiran	Client Company	Sm Entertainment Sdn Bhd
Start Date	03-October-2023	Due Date	12-April-2024
Project Manager	Muhammad Ammar	Status	In Progress

Status Task

IN PROGRESS
14%

PENDING REVIEW
0%

IN REVIEW
14%

Task Project

#	Task	Start Date	End Date	Status	Action
1	Print document	15 December 2023	22 December 2023	Complete	
2	Testing	15 December 2023	23 December 2023	Complete	
3	User Acceptance Testing	11 January 2024	19 January 2024	To Do	
4	Make Packages	20 March 2024	26 April 2024	To Do	

Figure A.12 Staff Task Page