CAREERNEST MANAGEMENT SYSTEM

NURHAZWANI BINTI OMAR

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CAREERNEST MANAGEMENT SYSTEM

NURHAZWANI BINTI OMAR

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

CAREERNEST MANAGEMENT SYSTEM

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

STUDENT : ______________________________ Date: 14SEPTEMBER 2024
(NURHAZWANI BINTI OMAR

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

SUPERVISOR : ______________________________ Date: 14 SEPTEMBER 2024
(DR NUR ATIKAH BINTI ARBAIN)

DEDICATION

This project is dedicated to my beloved parents, whose unwavering support, encouragement, and sacrifices have made this journey possible. Your love and guidance have been my greatest motivation, and your constant belief in my abilities has been a source of immense strength and inspiration.

To my esteemed professors and mentors, I extend my deepest gratitude for your invaluable guidance and wisdom. Your dedication to imparting knowledge and your insightful feedback have significantly shaped the quality of this work. Your support has been instrumental in my academic and personal growth.

Lastly, to my dear friends and classmates, thank you for the camaraderie and shared experiences that have made this journey truly memorable. Your friendship, support, and the joyous moments I have shared have enriched this experience beyond measure.

ACKNOWLEDGEMENTS

I would like to convey my heartfelt appreciation to Dr Nur Atikah binti Arbain for her significant support and direction in successfully finishing this assignment. Her profound insight and unwavering encouragement have been instrumental in shaping this work to its current form. Her dedication and commitment to guiding me through this project have been invaluable, and I am sincerely grateful for her patience and wisdom throughout this journey.

I am deeply thankful to my dear parents for their unfailing encouragement and support during this journey. Their trust and belief in my abilities have been my greatest source of strength and motivation. Their constant love and sacrifice have provided me with the resilience needed to persevere through the challenges I encountered. I am eternally grateful for their steadfast presence in my life.

Furthermore, I want to express my deepest gratitude to my teachers and mentors for their wisdom and insightful input, which have substantially improved the quality of my research. Their dedication to imparting knowledge and their constructive feedback have greatly enriched my understanding and development. I am immensely thankful for their time, effort, and the invaluable lessons they have imparted to me.

Finally, I wish to extend my heartfelt thanks to my friends and classmates for their camaraderie and support, which have made this experience unforgettable. Their companionship and encouragement have provided me with a sense of community and belonging, making this journey not only bearable but also enjoyable. I am truly blessed to have such wonderful individuals by my side, whose support has been a pillar of strength throughout this endeavor.

ABSTRACT

CareerNest Management System is a comprehensive job portal application designed to facilitate the recruitment process for administrators, jobseekers, and employers. Developed using modern tools like Laragon, PHP, HTML, phpMyAdmin, Visual Studio Code, and Microsoft Power BI, the system provides a robust platform for managing job postings, applications, and user profiles. Administrators have full control over company, jobseeker, and employer data, performing essential tasks such as updating records and using Power BI for detailed system-wide visualizations. Jobseekers benefit from an intuitive interface that allows them to search for jobs, update their profiles, upload resumes, and track their application statuses. The system ensures application accuracy by prompting jobseekers to delete old submissions before reapplying with updated information, thus preventing duplicate entries. Employers can efficiently manage their profiles, create, and manage job postings, and review applications submitted by jobseekers. The platform streamlines decision-making processes, enabling employers to approve or reject applications with ease. To ensure the system meets user needs, User Acceptance Testing (UAT) was conducted using a Google Form survey, gathering valuable feedback that helped refine the system. With its well-architected design, CareerNest Management System ensures scalability, reliability, and ease of maintenance, making it an indispensable tool in the recruitment and hiring process.

ABSTRAK

Sistem Pengurusan CareerNest adalah sebuah aplikasi portal pekerjaan yang komprehensif, direka untuk memudahkan proses pengambilan pekerja bagi pentadbir, pencari kerja, dan majikan. Dibangunkan menggunakan alat moden seperti Laragon, PHP, HTML, phpMyAdmin, Visual Studio Code, dan Microsoft Power BI, sistem ini menyediakan platform yang kukuh untuk mengurus iklan pekerjaan, permohonan, dan profil pengguna. Pentadbir mempunyai kawalan penuh ke atas data syarikat, pencari kerja, dan majikan, melaksanakan tugas penting seperti mengemas kini rekod dan menggunakan Power BI untuk visualisasi terperinci seluruh sistem. Pencari kerja menikmati antara muka yang intuitif yang membolehkan mereka mencari pekerjaan, mengemas kini profil, memuat naik resume, dan menjejaki status permohonan mereka. Sistem ini memastikan ketepatan permohonan dengan menggesa pencari kerja untuk memadamkan penghantaran lama sebelum memohon semula dengan maklumat yang dikemas kini, sekali gus mengelakkan kemasukan duplikat. Majikan boleh mengurus profil mereka dengan cekap, mencipta dan mengurus iklan pekerjaan, serta menyemak permohonan yang dihantar oleh pencari kerja. Platform ini memudahkan proses membuat keputusan, membolehkan majikan meluluskan atau menolak permohonan dengan mudah. Untuk memastikan sistem memenuhi keperluan pengguna, Ujian Penerimaan Pengguna (UAT) telah dijalankan menggunakan tinjauan Google Form, mengumpul maklum balas yang berharga yang membantu memperhalusi sistem. Dengan reka bentuknya yang tersusun dengan baik, Sistem Pengurusan CareerNest memastikan skalabiliti, kebolehpercayaan, dan kemudahan penyelenggaraan, menjadikannya alat yang penting dalam proses pengambilan pekerja dan penggajian.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

FYP	-	Final Year Project
DBLC	-	Database Life Cycle
PBI	-	Power BI
UAT	-	User Acceptance Testing
GUI	-	Graphical User Interface

Chapter 1: INTRODUCTION

1.1 Introduction

The hiring process is crucial to an organization's competitiveness and success in today's fast-paced commercial world. With the rapidly evolving job market and the increasing demand for specialized skills, conventional hiring practices frequently fall short in meeting the dynamic needs of both organizations and job seekers. Traditional methods, often characterized by manual processes and fragmented systems, can lead to inefficiencies, delays, and missed opportunities for both employers and candidates.

Acknowledging the significance of proficient and fruitful talent acquisition, the creation of an advanced CareerNest Management System offers a compelling solution to streamline and enhance the recruiting process. An effective CareerNest Management System not only addresses the shortcomings of traditional hiring practices but also leverages technology to provide a more agile, transparent, and efficient approach to recruitment.

The primary goal of this system is to bridge the gap between job seekers and employers, creating a seamless and engaging platform for both parties. For organizations, the CareerNest Management System facilitates comprehensive management of the hiring process, from job posting to candidate selection, thereby reducing administrative burdens and accelerating time-to-hire. By providing a centralized dashboard, administrators can easily manage company profiles, job postings, and candidate applications, ensuring that the recruitment process is both organized and efficient.

In summary, the development of this advanced CareerNest Management System represents a significant leap forward in the field of talent acquisition. By harnessing the power of modern technologies and addressing the limitations of traditional hiring practices, the CareerNest Management System offers a comprehensive and efficient solution for both job seekers and employers. This system not only enhances the recruitment process but also contributes to the overall competitiveness and success of organizations in the modern business landscape.

1.2 Problem statement(s)

- Decentralized candidate data: In many organizations, candidate data is scattered across multiple systems, spreadsheets, and email accounts, leading to a decentralized and fragmented data management approach. This decentralization makes it challenging to maintain accurate and up-to-date candidate information, complicates the process of tracking applicant progress, and hinders effective communication between hiring teams. The lack of a centralized repository for candidate data results in inefficiencies, increased risk of data loss, and difficulty in ensuring data consistency across the recruitment process.

- Inefficiency in managing job applications: Traditional hiring practices involve manual processes for managing job applications, which can be time-consuming and prone to human error. The absence of a streamlined, automated system for tracking and processing applications leads to inefficiencies and delays in the hiring process. This inefficiency hampers an organization's ability to quickly identify and secure top talent, potentially causing them to lose out to competitors with more agile recruitment systems.

- Lack of application tracking for job seekers: Job seekers are unable to track the status of their applications effectively. The current system does not provide clear and timely updates on application statuses, leading to frustration and uncertainty among job seekers about the progress of their job applications.

1.3 Objective(s)

This project embarks on the following objectives:

1. Centralize candidate data: Develop a centralized repository within the CareerNest Management System to store and manage all candidate information. This will ensure data consistency, enhance data integrity, and streamline the process of tracking applicant progress. By centralizing candidate data, the system aims to improve the efficiency of communication among hiring teams and reduce the risk of data loss.

2. Automate and streamline job application management: Implement automated processes for handling job applications to eliminate manual errors and inefficiencies. This includes features for tracking application status, notifying candidates, and managing submissions. The objective is to accelerate the hiring process, enabling organizations to quickly identify and secure top talent, thereby enhancing their competitive edge in the job market.

3. Comprehensive application tracking for job seekers: implement a robust application tracking system that allows job seekers to view the status of their applications in real-time. This includes updates on whether applications are received, under review, accepted, or rejected, providing job seekers with transparency, and reducing uncertainty throughout the application process.

1.4 Scope

The CareerNest Management System aims to develop a comprehensive, efficient, and secure job portal application to facilitate the recruitment process for administrators, job seekers, and employers. The scope of the project includes the following modules and functionalities:

1. Admin module:

   - Dashboard: Provide a centralized dashboard for administrators to monitor and manage system activities.
   - User management: Enable admins to view, update, and delete company, jobseeker, and employer details.
   - Data management and reporting: Utilize Microsoft Power BI for advanced data visualization and reporting capabilities within the CareerNest Management System, ensuring comprehensive insights and analytics to enhance decision-making processes.

2. Jobseeker module:

   - Job search: Implement a job search feature that allows job seekers to find and browse available job listings.
   - Profile management: Allow job seekers to create, update, and manage their profiles, including uploading resumes.
   - Application management: Enable job seekers to apply for jobs, view application status, and manage their applications. Ensure that job seekers can only apply once per job and must delete a previous application to reapply with an updated resume to prevent duplicate submissions.

3. Employer module:

   - Profile management: Allow employers to create, update, and manage their profiles.
   - Job posting: Enable employers to post new job listings and manage existing job posts.

- Application review: Provide a feature for employers to view applications received for their job postings and take actions such as approving or rejecting candidates.

4. Security and data protection:

- Data encryption: Implement data encryption to protect sensitive candidate information.
- Access control: Establish role-based access control to ensure that only authorized users can access specific functionalities and data.

5. User experience and interface design:

- Responsive design: Ensure that the application is responsive and provides a seamless experience across different devices and screen sizes.
- User-friendly interface: Design intuitive interfaces for all user roles (admin, jobseeker, employer) to facilitate ease of use and enhance user satisfaction.

1.5 Project Significance

The development and implementation of the CareerNest Management System hold substantial significance for various stakeholders involved in the recruitment process. This project is poised to deliver a range of benefits that enhance the efficiency, effectiveness, and overall experience of job seekers, employers, and administrators.

1. Enhanced efficiency in recruitment:

- Streamlined processes: By automating and centralizing job application management, the CareerNest Management System significantly reduces the time and effort required to manage recruitment activities. This allows organizations to expedite their hiring processes and respond more quickly to talent needs.

- Reduced administrative burden: Administrators can manage job postings, candidate applications, and data visualization through a single, integrated platform. This minimizes manual data entry and the potential for human errors, allowing administrative resources to be allocated more effectively.

2. Improved candidate experience:

- User-friendly interface: The system offers a seamless and intuitive interface for job seekers to search for jobs, apply, and track their application status. This enhances the overall user experience, making it easier for candidates to engage with potential employers.
- Timely updates and feedback: Job seekers receive timely notifications about their application status, ensuring transparency and keeping them informed throughout the recruitment process. This can lead to higher levels of satisfaction and engagement among candidates.

3. Empowered employers:

- Effective job posting and management: Employers can easily create, update, and manage job postings, as well as review applications from potential candidates. This empowers employers to take control of their recruitment processes and make informed hiring decisions more efficiently.
- Streamlined applicant review: The system provides employers with a comprehensive view of all applications, allowing them to quickly review and take actions on candidate submissions. This reduces the time-to-hire and helps in securing top talent faster.

1.6 Expected Output

The implementation of the CareerNest Management System is expected to yield significant improvements in the recruitment process, benefiting both job seekers and employers. The expected outcomes include:

1. Centralized and standardized candidate data:
   - Standardized data storage: The CareerNest Management System will systematically organize all candidate information, including contact details, resumes, and other relevant data. By centralizing candidate data in a standardized format, the system streamlines data management processes, reduces duplication, and enhances accessibility and organization for recruiters and hiring managers.

2. Improved quality of hires:
   - Informed hiring decisions: With enhanced search and filtering capabilities, recruiters can identify and select candidates who closely match job requirements and organizational needs. This targeted approach improves the quality of hires, ensuring that organizations onboard candidates who are best suited for the roles and contribute positively to organizational success.

3. Streamlined recruitment processes:
   - Efficiency and productivity: The CareerNest Management System automates and streamlines various aspects of the recruitment process, reducing administrative burdens and manual errors. It accelerates time-to-hire by facilitating faster candidate identification, application processing, and decision-making, thereby increasing recruitment efficiency and productivity.

1.7 Conclusion

The CareerNest Management System represents a significant advancement in modernizing the recruitment process, addressing longstanding inefficiencies, and enhancing user experience and data security. By leveraging technologies such as Laragon, phpMyAdmin, PHP, HTML, Visual Studio Code, and Microsoft Power BI, the system offers a robust platform that streamlines job searching, application management, and candidate evaluation for administrators, job seekers, and employers alike.

This system not only centralizes candidate data and automates application management but also empowers organizations to make informed hiring decisions swiftly and effectively. The user-friendly interfaces and comprehensive functionalities cater to the diverse needs of stakeholders, fostering efficiency, transparency, and reliability throughout the recruitment lifecycle.

Moving forward, the implementation of the CareerNest Management System promises to deliver tangible benefits, including improved recruitment efficiency, enhanced candidate experience, and strengthened organizational competitiveness. By embracing modern technology and best practices, this system sets a new standard in talent acquisition, positioning organizations to thrive in today's dynamic business environment.

In summary, CareerNest is poised to revolutionize how organizations approach recruitment, offering a strategic tool that not only meets current demands but also anticipates future trends in talent management and acquisition.

CHAPTER 2: PROJECT METHODOLOGY AND PLANNING

2.1 Introduction

Chapter 2 of the CareerNest Management System project discusses the methodology and planning, focusing on the Database Lifecycle (DBLC). The DBLC is a structured approach that guides the development and management of the system's database, ensuring it is robust, scalable, and secure. This methodology outlines the steps from the initial requirements analysis through design, implementation, testing, and maintenance, ensuring the database aligns with the organization's goals and user needs.

The DBLC process begins with the database initial study, where the need for the database is identified, and the requirements are analyzed. Next is the database design phase, where the blueprint for the database is created, focusing on the structure and storage of data. After design, the implementation and loading phase brings the database to life, followed by the testing and evaluation phase, where the system is thoroughly tested to ensure it meets the necessary standards.

Finally, the database enters the operation phase, where it is used in daily activities, and the maintenance and evolution phase, where ongoing updates and improvements are made. These phases ensure the database remains functional and adapts to changing organizational needs, making the DBLC an essential part of the project’s success.

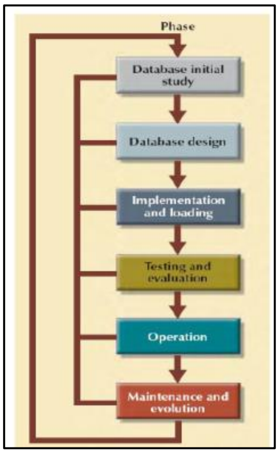


Figure 2.1: Database Life Cycle (DBLC)

2.2 Project Methodology

Chapter 2 outlines the methodology adopted for the development and implementation of the CareerNest Management System, integrating specific phases and approaches to optimize the creation of a reliable recruitment portal:

1. Requirement analysis:

   - Conduct thorough analysis to gather and document functional and non-functional requirements for the CareerNest Management System.

   - Engage stakeholders, including administrators, job seekers, and employers, to understand their needs and expectations.

   - Define system objectives, features, and constraints to establish a clear project scope.

2. Database Lifecycle (DBLC) approach:

   - Implement the DBLC methodology, encompassing stages from requirements analysis to maintenance.

   - Begin with conceptual design to define data entities, relationships, and attributes based on gathered requirements.

   - Progress to logical design to translate conceptual models into normalized database schemas, ensuring data integrity and efficiency.

   - Develop physical design to specify database structures, indexes, and storage considerations for optimal performance.

3. Development environment and tools:

   - Utilize Laragon as the local development environment, integrating Apache, MySQL, PHP, and phpMyAdmin for database management.

- Employ Visual Studio Code as the primary integrated development environment for coding PHP and HTML.

- Incorporate Microsoft Power BI for data visualization and reporting capabilities.

4. Project planning and management:

   - Establish a project timeline with defined milestones for each phase of development, testing, and deployment.

   - Allocate resources effectively, including human resources, budget, and infrastructure, to meet project deadlines and objectives.

   - Implement agile project management methodologies to adapt to changing requirements and optimize project outcomes.

5. Testing and quality assurance:

   - Conduct rigorous testing throughout the development lifecycle, including unit testing, integration testing, and user acceptance testing (UAT).

   - Ensure comprehensive testing of database functionalities, user interfaces, and system integrations to identify and rectify any issues.

   - Validate system performance, security, and usability to deliver a stable and reliable CareerNest Management System.

6. Documentation and training:

   - Maintain detailed documentation of system architecture, database schemas, technical specifications, and user manuals.

- Provide training sessions and support materials for administrators, job seekers, and employers to effectively use the CareerNest Management System.

- Ensure documentation and training materials are updated regularly to reflect system enhancements and changes.

7. Deployment and maintenance:

- Deploy the CareerNest Management System ensuring scalability, reliability, and security considerations.

- Monitor system performance, security vulnerabilities, and user feedback post-deployment to proactively address issues.

- Establish protocols for ongoing maintenance, updates, and enhancements to continuously improve system functionality and user experience.

By following this structured methodology, the CareerNest Management System aims to deliver a scalable, secure, and user-friendly recruitment portal that meets the needs of stakeholders and supports organizational goals effectively.

2.2.1 Database Initial Study

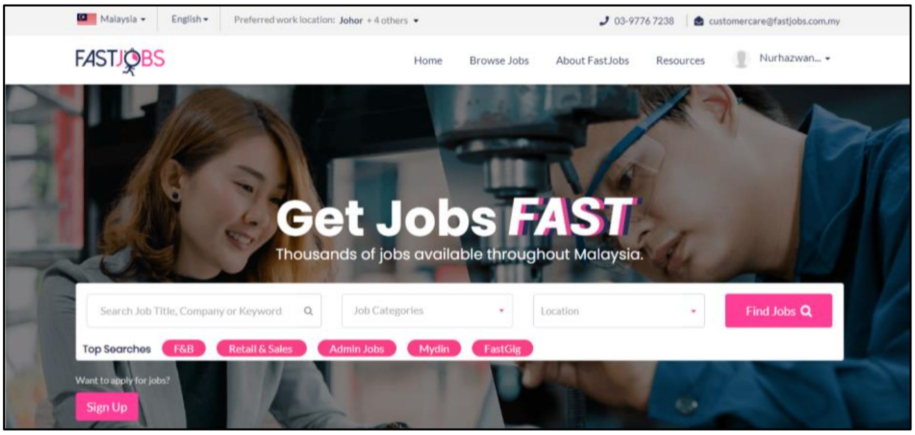


Figure 2.2: FastJobs Homepage

FastJobs is a prominent online job portal designed to simplify the job search and recruitment process throughout Southeast Asia. In this phase, my primary focus is on thoroughly analyzing the functionality and features of FastJobs to understand how they contribute to the job search and recruitment process. The first step involves evaluating the user experience, particularly the platform's user-friendly interface optimized for mobile devices. This analysis will help determine how effectively the interface enhances the job search experience for users who prefer accessing job listings and managing applications on their smartphones.

Next, the focus shifts to the job listing management aspect, examining how FastJobs' robust search filters and management tools assist job seekers in quickly finding relevant opportunities. Additionally, the exploration includes how employers utilize these tools to efficiently post job ads and manage applications, ensuring they can attract and select suitable candidates with minimal effort.

Another critical area of focus is the platform's communication and scheduling features, which allow employers to seamlessly review resumes, communicate with applicants, and schedule interviews. Understanding these features helps assess how they streamline interactions between employers and job seekers, contributing to a smoother recruitment process.

Furthermore, the assessment focuses on FastJobs' accessibility and usability, particularly in the competitive Southeast Asian job market. This includes evaluating how well the platform serves both job seekers and employers, ensuring it meets their needs effectively. Based on this comprehensive analysis, key strengths of FastJobs, such as its mobile optimization and efficient recruitment tools, are identified, along with suggestions for areas of improvement to enhance functionality and user satisfaction.

2.2.2 Database Design

The database design for the CareerNest Management System ensures the platform runs smoothly and meets all operational requirements. This phase includes conceptual design, logical design, and physical design. In the conceptual design stage, create an Entity-Relationship Diagram (ERD) to visualize the data and relationships. Entities such as Jobseeker, Employer, Company, Advertisement, Applications, and Resume are identified, and their attributes and relationships are defined. Normalization is applied to maintain data integrity and reduce data redundancy.

MySQL is chosen as the Database Management System (DBMS) for CareerNest due to its robustness, reliability, and ability to handle large datasets efficiently. MySQL’s performance and scalability are ideal for managing the platform’s diverse and dynamic data needs. During the logical design phase, the ERD is translated into a detailed relational data model, defining the database structure, including tables, primary keys, foreign keys, and their relationships. Business logic and validation rules are also set to ensure data accuracy and consistency. The main tables are Jobseeker, Employer, Company, Advertisement, Applications, and Resume.

Finally, the physical design phase converts the logical model into a physical schema for MySQL. This includes defining tables, columns, data types, indexes, sequences, and constraints to optimize performance and ensure data integrity. For example, tables such as Jobseeker, Employer, Company, Advertisement, Applications, and Resume are created with appropriate primary keys, foreign keys, and constraints to maintain referential integrity and enhance query performance. This comprehensive database design ensures that the CareerNest Management System's data is well-organized, easily accessible, and efficiently managed, facilitating smooth operation and optimal performance of the platform.

2.2.3 Implementation and Loading

The implementation and loading phase of the CareerNest Management System is crucial for transitioning from design to an operational platform. This phase involves the actual creation of the database and the development of application components, followed by populating the database with initial data. Implementation begins with setting up the MySQL database according to the physical design schema. This includes creating tables for entities like Jobseeker, Employer, Company, Advertisement, Applications, and Resume. Each table is defined with appropriate columns, data types, indexes, and constraints to ensure data integrity and optimize performance. Stored procedures, triggers, and views are also implemented to enforce business rules and facilitate complex queries.

Once the database structure is in place, the next step is to load the initial data. This involves inserting baseline data into the tables to set up the system for use. For example, the Company table is populated with details of various companies, the Jobseeker and Employer tables with user profiles, and the Advertisement table with job listings. Data loading is typically done through bulk insert operations to ensure accuracy and efficiency. After the initial data load, the system undergoes rigorous testing to verify that all components function correctly and data flows seamlessly between different parts of the system. This includes testing data integrity, application performance, and user interface interactions to ensure the system meets the specified requirements and is ready for deployment.

Overall, the implementation and loading phase ensures that the CareerNest Management System is fully equipped with the necessary data and functionalities to provide a robust and efficient platform for job seekers and employers.

2.2.4 Testing and Evaluation

Testing and evaluation are critical phases in the development of the CareerNest Management System, ensuring that the system operates correctly, efficiently, and meets all specified requirements. These phases involve multiple testing strategies and evaluation techniques to identify and resolve any issues before the system goes live. The testing phase begins with unit testing, where individual components of the system, such as database queries, stored procedures, and application modules, are tested in isolation. This ensures that each part functions as intended. Next, integration testing is conducted to verify that different modules and components work together seamlessly. For example, interactions between the jobseeker application form and the applications table are tested to ensure data is correctly saved and retrieved.

Functional testing is then performed to validate that the system meets all functional requirements. Test cases are designed to cover all aspects of the system, such as job listing creation, profile management, application submission, and data retrieval. User acceptance testing (UAT) follows, involving real users who test the system in scenarios that simulate actual use. This phase helps identify any usability issues or discrepancies from user expectations. Performance testing is also essential to evaluate how the system handles high loads and stress conditions. This ensures that the system remains responsive and stable under peak usage. Security testing is conducted to identify and fix vulnerabilities, ensuring data protection and user privacy.

The evaluation phase involves reviewing test results and gathering user feedback. Any identified issues are prioritized and addressed. Evaluations include assessing the system’s reliability, efficiency, and user satisfaction. Overall, the testing and evaluation phases are vital for delivering a high-quality, reliable CareerNest Management System. They ensure that the system is robust, secure, and user-friendly, providing a seamless experience for job seekers and employers alike.

2.2.5 Operation

The operation phase of the CareerNest Management System ensures smooth and effective operation for all users. This encompasses routine maintenance, monitoring, support, updates, and enhancements to maintain reliability and meet user needs.

Routine maintenance includes regular checks such as system logs for errors, monitoring performance metrics like response times and server load, and ensuring data integrity through data visualization using Microsoft Power BI. Data visualization plays a crucial role in presenting actionable insights to administrators. Moreover, regular updates to software and database components ensure the system remains current and secure. Updates are carefully planned and tested to minimize disruption, with users informed in advance of scheduled maintenance to mitigate inconvenience. Monitoring tools provide real-time alerts to administrators, enabling swift resolution of issues to prevent downtime or slow performance. This proactive approach ensures the system operates within optimal parameters.

Support and troubleshooting are central to user satisfaction. A dedicated support team uses a ticketing system to manage and resolve user issues promptly, whether related to job applications, employer postings, or system navigation. Finally, continuous updates and enhancements are driven by user feedback to improve functionality, add features, and enhance security. These updates are part of ongoing efforts to ensure the CareerNest Management System evolves to meet the dynamic needs of job seekers and employers.

In conclusion, the operation phase is pivotal in maintaining the reliability and effectiveness of the CareerNest Management System, supporting its role as a trusted platform for recruitment and job management.

2.2.6 Maintenance and Evolution

Maintenance and evolution are critical phases in the lifecycle of the CareerNest Management System, ensuring its reliability, efficiency, and alignment with evolving user needs and technological advancements.

In terms of maintenance, the system undergoes regular checks and updates to uphold performance and data integrity. This includes routine monitoring of system logs for errors and performance metrics to detect and address issues promptly. It leverages Microsoft Power BI for data visualization, which plays a crucial role in presenting actionable insights to administrators for informed decision-making and strategic planning. This proactive approach helps prevent downtime and ensures seamless operation for administrators, job seekers, and employers alike.

Evolution in the CareerNest Management System is driven by user feedback and technological advancements. Updates and enhancements are meticulously planned and tested to introduce new features, improve existing functionalities, and enhance overall user experience. User feedback guides the prioritization of updates, ensuring that the system evolves in ways that directly benefit its stakeholders. These updates are communicated transparently to users to minimize disruption and maximize the adoption of new features.

In conclusion, maintenance and evolution are integral to the ongoing success of the CareerNest Management System. By maintaining high standards of performance and responsiveness through proactive maintenance and evolving to meet user expectations and technological innovations, the system remains a robust and valuable tool in the recruitment and job management landscape.

2.3 Project Schedule and Milestone

Table 2.1: Gantt Chart

Year	2024						
Month **Activities**	March	April	May	June	July	August	September
Database Initial Study	▨						
Database Design		▨					
Implementation and Loading			▨	▨	▨		
Testing and Evaluation						▨	
Operation						▨	
Maintenance and Evolution						▨	▨

2.4 Conclusion

Chapter 2 has detailed the methodology and planning essential for developing the CareerNest Management System. Emphasizing the Database Lifecycle (DBLC) approach, the chapter ensures systematic design, implementation, and management of the system's database infrastructure. The DBLC methodology guides every phase from requirements analysis to testing and deployment, ensuring alignment with organizational goals and user needs. Effective project planning highlighted the importance of clear objectives, scope definition, resource allocation, and adherence to timelines. This groundwork sets the stage for the successful implementation of CareerNest, aiming to deliver a secure, user-friendly recruitment portal that meets stakeholders' expectations. Chapter 2 provides a structured framework for disciplined project execution, crucial for achieving project milestones and goals.

CHAPTER 3: ANALYSIS

3.1 Introduction

Chapter 3 of the CareerNest Management System project centers on analyzing the current system, proposing improvements, and defining requirements for future development. This chapter is critical in identifying the strengths and weaknesses within the current recruitment process, with the goal of enhancing efficiency and user experience. The analysis begins by evaluating the existing system to pinpoint its limitations and identify areas needing improvement. This evaluation forms the basis for proposing targeted enhancements to address specific challenges and inefficiencies faced by stakeholders. Furthermore, Chapter 3 outlines proposed improvements for CareerNest, including both technical optimizations and functional enhancements. These improvements are designed to streamline operations and improve usability across the recruitment portal.

The chapter also details the requirements for the future system, drawing on stakeholder feedback, industry standards, and technological advancements. These requirements encompass both functional features and capabilities, as well as non-functional considerations, ensuring a comprehensive foundation for system development. To support this analysis, Chapter 3 utilizes Entity-Relationship Diagrams (ERD) and Data Flow Diagrams (DFD). ERDs are employed to model the data structure and relationships between various entities within the system, providing a clear understanding of how data is organized. DFDs are used to map the flow of data through the system, highlighting how information is processed and transferred between different components. Additionally, normalization techniques are applied to optimize the database design by reducing redundancy and ensuring data integrity. These diagrams and techniques collectively contribute to a thorough analysis, forming the foundation for proposing targeted improvements and defining precise requirements for the future system.

In summary, Chapter 3 provides a strategic analysis of the current system, proposes enhancements, and defines requirements essential for designing an effective and user-friendly CareerNest Management System. This sets the stage for the subsequent design and development phases, aiming to deliver a solution that meets organizational goals and exceeds user expectations.

.

3.2 Problem Analysis

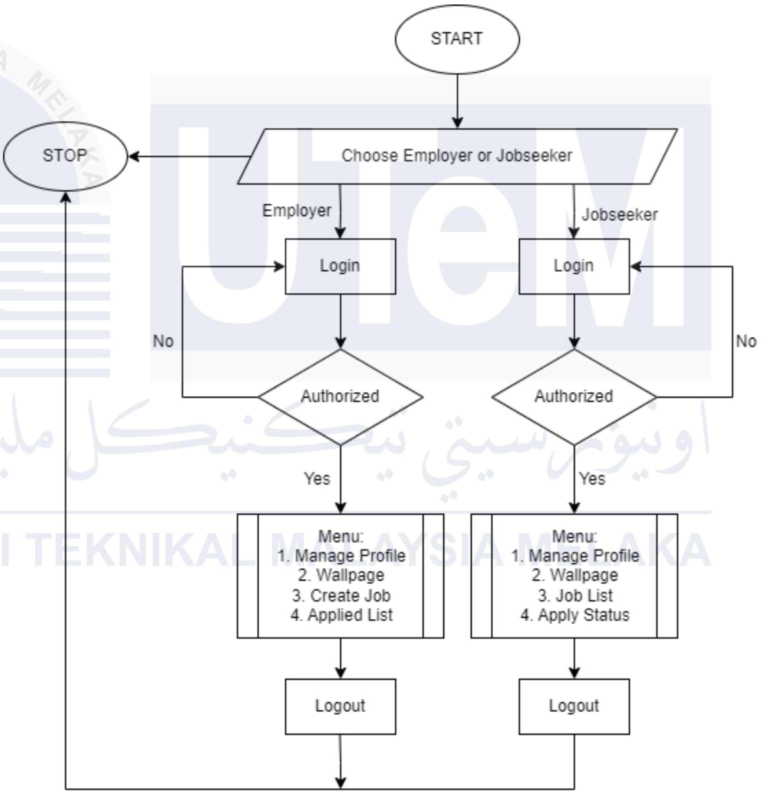


Figure 3.1: Flowchart of current system

The current system, as illustrated in Figure 3.1, follows a structured process where users begin by selecting their role as either an Employer or a Jobseeker. Each role has a separate login page, leading to a specific workflow for verifying credentials and accessing relevant functionalities. For Employers, successful login grants access to options like managing profiles, creating job listings, and viewing applicants. Similarly, Jobseekers can manage their profiles, browse job listings, and check the status of their applications upon successful login.

However, this system presents several challenges. First, having separate login pages for Employers and Jobseekers complicates the user experience. A more efficient approach would be to provide a single, unified login page where users can log in and have their roles authorized by the system, streamlining the process.

Another significant issue is the absence of an application history feature. Jobseekers currently lack the ability to view their past applications, making it difficult to track their job search progress. Furthermore, the system does not clearly indicate whether an employer has reviewed their application or if there are any upcoming steps, such as an interview session. Addressing these shortcomings would greatly improve the overall user experience, providing clarity and convenience for both Employers and Jobseekers.

3.3 The proposed improvement / solutions

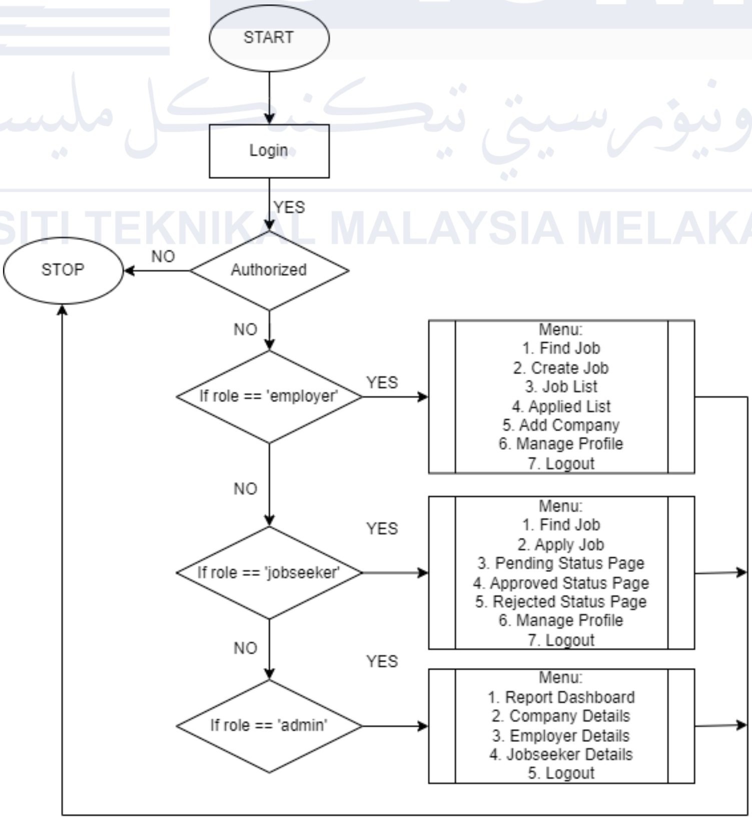


Figure 3.2: Flowchart of proposed system

In the proposed system, as depicted in Figure 3.2, several enhancements are introduced to address the challenges identified in the current system. One of the key improvements is the introduction of a unified login page that serves both Employers and Jobseekers. This single-entry point eliminates the need for separate login pages, simplifying the login process. The system automatically identifies the user’s role whether they are an employer or a jobseeker and grants access to the appropriate functionalities, thereby improving user experience and reducing confusion.

Another significant enhancement is the addition of an application history feature for Jobseekers. This feature allows Jobseekers to view a comprehensive record of all their job applications, addressing the current system's lack of visibility into past applications. Jobseekers can now easily track their progress and revisit previous applications as needed.

Furthermore, the proposed system introduces dedicated status pages pending, approved, and rejected that allow jobseekers to distinguish between different stages of their applications. The pending status page displays applications still under review, the approved status page highlights accepted applications, and the rejected status page shows unsuccessful ones. These status-specific pages offer a clear and organized overview of the job search process, making it easier for jobseekers to manage and track their applications.

Overall, the proposed system effectively addresses the shortcomings of the current system by streamlining the login process, introducing a comprehensive application history feature, and improving the clarity and organization of application status updates. These enhancements significantly improve the user experience for both Employers and Jobseekers.

3.4 Requirement analysis of the to-be system

Requirement analysis for the CareerNest Management System involves understanding what administrators, job seekers, and employers need from the system. Administrators need a dashboard to oversee company profiles, job postings, and applicant data, using tools like Microsoft Power BI for insights. Job seekers need an easy-to-use interface for browsing jobs, applying, and tracking applications. Employers require tools for posting jobs, managing candidates, and reviewing applications efficiently. By focusing on these needs, the system aims to improve user experience, streamline operations, and support effective recruitment processes.

3.4.1 Functional Requirement (Process Model)

1. Jobseeker Module

i. Job search process

- Allow job seekers to search for jobs based on various criteria such as job title, location, salary range, and category.
- Display job listings with detailed job descriptions, salary range, and company.

ii. Application process

- Enable job seekers to create and manage their profiles securely.
- Facilitate resume upload, with delete resumes.
- Restrict job seekers to apply for each job only once to prevent duplicate applications.
- Provide notifications for application status updates (e.g., application received, under review, accepted, rejected).

iii. Job application tracking

- Allow job seekers to view their application history, including applied jobs, status updates, and previous applications.

2. Employer Module

1. Job posting
    - Enable employers to create, edit, and manage job postings.
    - Include fields for job title, description, requirements, location, and salary range.
    - Allow employers to view their job history that they are created with status (e.g., active, and inactive).

2. Application management

    - Provide employers with a dashboard to view and manage job applications.
    - Allow employers to review applicant profiles, and resumes.
    - Enable employers to take actions on applications such as accepting or rejecting candidates.

3. Admin Module

- User and data management
    - Administer user accounts, roles, and permissions.
    - Manage employer and jobseeker profiles, ensuring data accuracy and security.
    - Enable editing or deletion of user accounts as required.
- System monitoring and maintenance
    - Utilize Microsoft Power BI for data visualization
    - Monitor system performance, identify bottlenecks, and optimize system resources.
    - Ensure data security through measures like data encryption, access controls, and vulnerability assessments.

3.4.1.1 Data Flow Diagram (DFD):

i. Context Diagram

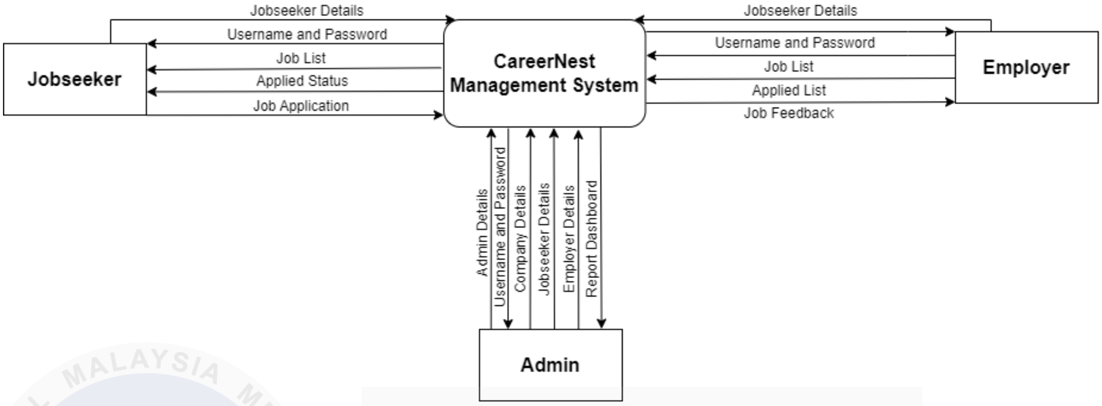


Figure 3.3: Context Diagram

Figure 3.3 illustrates a context diagram for the CareerNest Management System, providing a high-level overview of the system and its interactions with external entities. The diagram clearly outlines how the system exchanges information with three key groups: jobseekers, employers, and administrators.

In this context, the jobseeker interacts with the system by inputting details such as their username and password. They can also browse the job list, submit job applications, and monitor their application status through the system. This flow of information is critical for enabling jobseekers to manage their job search efficiently. The employer also engages with the system by providing login credentials and accessing the job list. Employers are responsible for managing the applied list, which includes reviewing applications from jobseekers, and providing job feedback through the system. This interaction helps streamline the recruitment process from the employer's perspective.

Additionally, the admin plays a crucial role in maintaining the system by managing jobseeker details, company details, and employer information. The admin is also responsible for generating and managing reports via the dashboard, which is essential for overseeing the system’s operations and ensuring that all processes run smoothly.

ii. Data Flow Diagram (DFD) Level 1

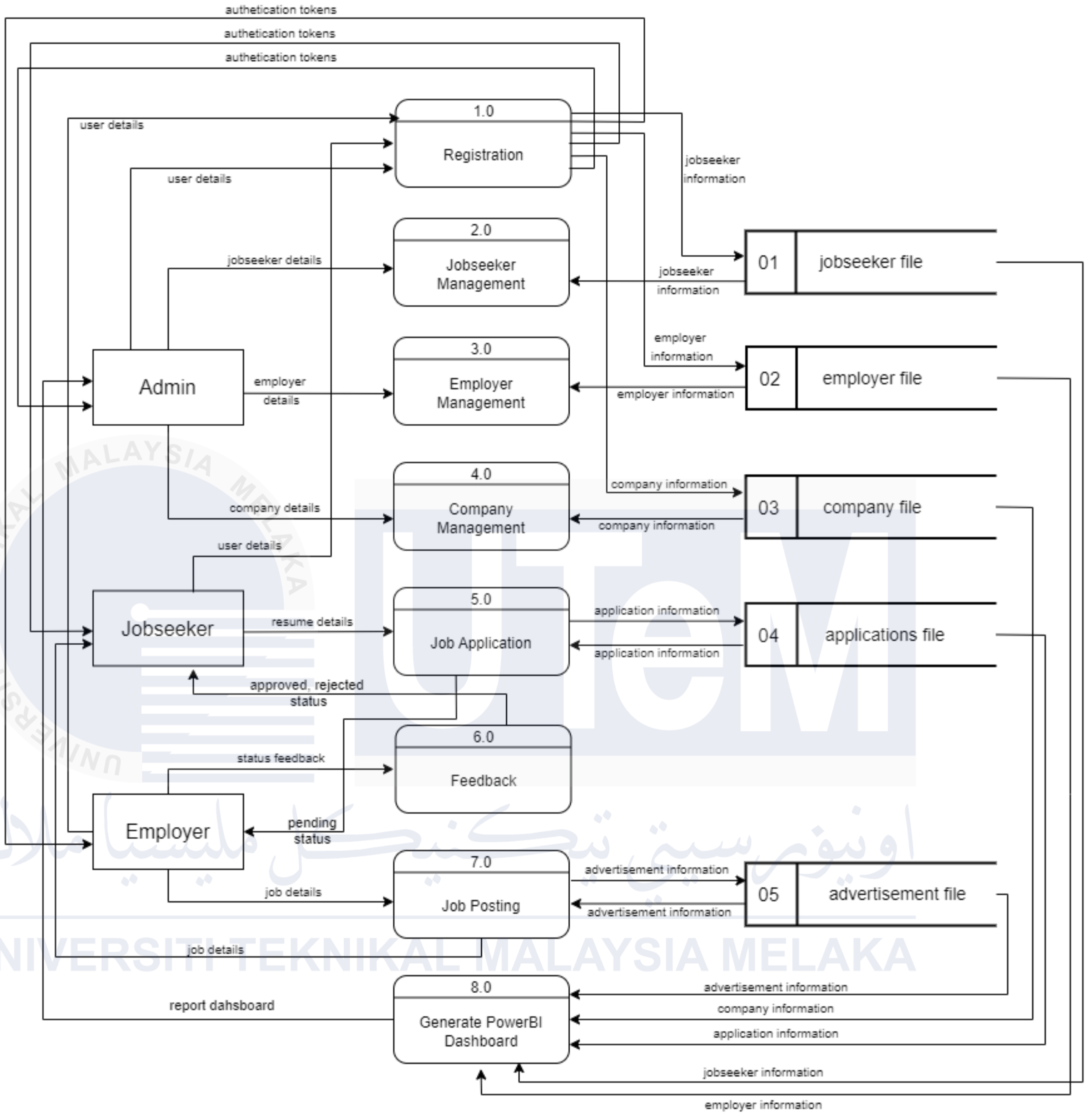


Figure 3.4: Data Flow Diagram Level 1

3.4.2 Non-functional requirement

3.4.2.1 Quality Requirements:

i. Accuracy**:** The system must accurately record and manage data related to user profiles, job postings, and applications. This ensures that information such as job details and applicant qualifications are stored correctly and processed without errors.

ii. Security**:** The system ensuring the security of user data is paramount. The system implements secure login mechanisms, encrypts sensitive information, and employs access controls to prevent unauthorized access to personal and organizational data. This safeguards the privacy and integrity of user information across all interactions within the platform.

iii. Reliability**:** The system is designed for high reliability, minimizing downtime to ensure continuous availability for administrators, employers, and job seekers. It leverages Microsoft Power BI for data visualization and monitoring. This enhances operational reliability by providing insights into system performance and ensuring effective management of system resources.

iv. Usability**:** The system prioritizes usability with an intuitive interface tailored for admin, employers, and job seekers. This design facilitates easy navigation and efficient use of features such as job searching, application management, and profile updates, enhancing overall user satisfaction and productivity.

3.4.2.2 Performance Requirements:

i. Response time**:** The system maintains fast response times to user actions such as job searches, application submissions, and data updates. This ensures a seamless user experience where interactions with the platform are processed promptly and efficiently.

ii. Scalability**:** Designed with scalability in mind, the CareerNest Management System can accommodate growth in user base and data volume over time. It can expand its resources and functionalities to handle increased traffic and user demands without compromising performance.

iii. Capacity**:** The system efficiently manages large amounts of data and supports concurrent user access. It is equipped to store and process diverse information related to job listings, user profiles, and application statuses, maintaining optimal performance under varying workloads.

3.4.3 Other Requirements

3.4.3.1 Software Requirements:

Table 3.1: Software Requirements

Software	Benefit	Reason
VS Code	Visual Studio Code is a lightweight but powerful source code editor which runs on my desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and has a rich ecosystem of extensions for other languages	VS Code provides a robust and convenient environment for coding, debugging, and version control
Laragon	Laragon is a lightweight and easy-to-use package that includes Apache, MySQL, and PHP. It also includes several additional tools and features that make it easy to set up and manage local development environments. One of the key advantages of Laragon is its "Quick Create" feature, which allows me to create a new WordPress or Laravel project with just a few clicks.	Laragon includes a user-friendly control panel that allows users to start, stop, and configure their local servers with just a few clicks.
phpMyAdmin	phpMyAdmin is a free and open-source administration tool for MySQL and MariaDB. It provides a web-based interface that simplifies database management tasks such as creating, modifying, and deleting databases, tables, fields, and executing SQL queries.	phpMyAdmin offers an intuitive interface, making it easier for users to manage their databases without needing extensive SQL knowledge.
Microsoft Power BI	Power BI is a business analytics tool that enables users to create interactive reports and dashboards. It allows for real-time data access, advanced data visualization, and seamless integration with other Microsoft products like Excel and Azure.	Power BI helps organizations to analyze and visualize data effectively, leading to more informed decision-making and better insights.

3.4.3.2 Hardware Requirements:

Table 3.2: Hardware Requirements

Hardware	Specification	Reason
Laptop Lenovo	Windows 11 Home Single Language	Windows 11 is the latest operating system from Microsoft, offering improved performance, security, and a user-friendly interface
	Intel(R) Core(TM) i3-8130U CPU @ 2.20GHz	The Intel Core i3 processor provides sufficient processing power for developing and testing the system. It can handle multiple tasks and processes without lagging, ensuring a smooth development experience.
	4.00 GB (3.3 GB usable)	4GB of RAM is enough for running the necessary software for development like Visual Studio Code, Laragon, and web browsers for testing. It allows for efficient multitasking and quick access to data and applications.

3.5 Conclusion

Chapter 3 has provided a comprehensive analysis of the current CareerNest Management System, focusing on identifying areas for improvement and defining requirements for its enhancement. By evaluating the existing system, this chapter highlighted its strengths and weaknesses, paving the way for targeted improvements aimed at enhancing efficiency and user experience within the recruitment process. The proposed enhancements outlined in the chapter, including technical optimizations and functional additions, aim to streamline operations and improve usability across the platform. Moreover, the detailed requirements derived from stakeholder feedback, industry standards, and technological advancements set a solid foundation for the design and development phases of the project. Ultimately, Chapter 3 serves as a critical phase in the project lifecycle, laying the groundwork for the development of an effective and user-friendly CareerNest Management System that aligns with organizational goals and exceeds user expectations.

CHAPTER 4: DESIGN

4.1 Introduction

Chapter 4 focuses on the design phase of the CareerNest Management System, detailing the conceptual, logical, and physical aspects of both the database and system architecture. This chapter delves into the database design, outlining how data is structured and organized to support the system's functionalities effectively. It also covers normalization, which is crucial for ensuring data integrity and minimizing redundancy by organizing data into related tables.

Additionally, the chapter explores query design, emphasizing the creation and optimization of SQL queries to efficiently retrieve and manipulate data. This ensures that the CareerNest platform can handle complex data requests and perform efficiently. Database objects such as triggers, and stored procedures are also addressed. Triggers automate certain actions based on specific events, while stored procedures encapsulate SQL code for reuse and complex operations. These elements help streamline processes and enforce business rules.

Furthermore, the chapter examines the Graphical User Interface (GUI) design, highlighting a user-centric approach to interface development aimed at enhancing user interaction and experience. By detailing these design elements, Chapter 4 aims to lay the groundwork for implementing a robust, scalable, and user-friendly CareerNest platform that meets the needs of job seekers, employers, and administrators alike.

4.2 System's Architecture

The architecture of the CareerNest Management System is designed with scalability, maintainability, and efficiency in mind, structured around a three-tier architecture model.

- **Presentation Layer:** This layer encompasses user interfaces tailored for administrators, employers, and job seekers. Developed using HTML, CSS, and JavaScript, it ensures a responsive and user-friendly experience, facilitating seamless navigation and interaction within the system.

- **Business Logic Layer:** Known as the operational layer in this context, this tier houses the core functionalities critical to the system's operations. It includes tasks such as user authentication, job posting management, application tracking, and dashboard visualization using Microsoft Power BI. These functionalities are implemented using PHP, ensuring robust integration with the database and efficient handling of system operations.

- **Data Layer:** At the foundation of the system is the data layer, responsible for managing and storing all system data securely. MySQL with phpMyAdmin serves as the database management system, ensuring data integrity, security, and optimal data retrieval performance. This layer supports the storage of user profiles, job postings, application records, and other essential system data, crucial for seamless system operation and user interaction.

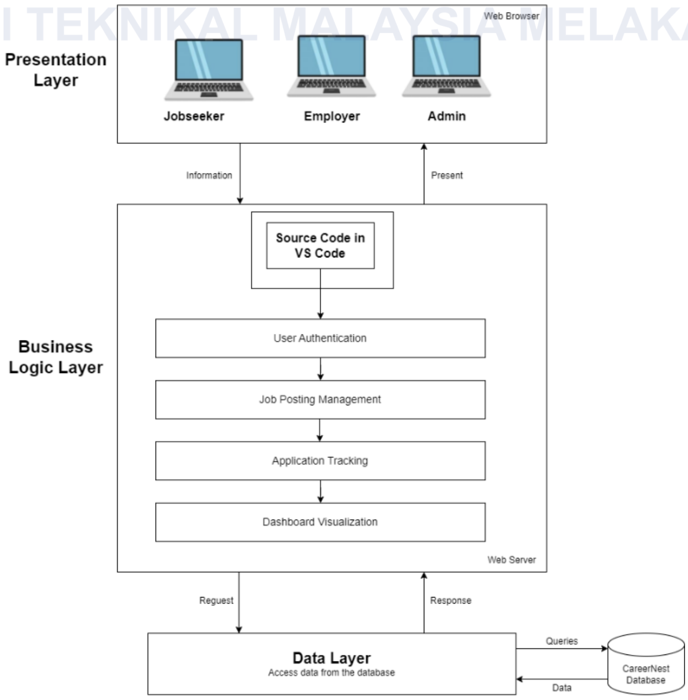


Figure 4.1: CareerNest System Architecture

4.3 Database Design

In the context of the CareerNest Management System, database design generally involves the following three stages namely conceptual design, logical design, and physical design. Conceptual design is the first step where I figure out what information the database needs to store and how different pieces of information are related. In the logical design stage, I turn the high-level concepts into a detailed plan. This means creating tables with specific columns and defining how these tables relate to each other. Finally, the physical design phase focuses on making the database work well in practice. This step ensures the database runs efficiently and can handle the actual data and queries it will encounter.

4.3.1 Conceptual Design

4.3.1.1 Normalization

Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity. Normalization aims to split large tables into smaller tables to ensure that each data unit is stored only once. The first normal form (1NF) requires that each table have a primary key and that each attribute in the table is atomic. The second normal form (2NF) requires that every attribute in the table that is not a primary key is functionally dependent on the entire primary key. This eliminates partial dependencies where only some of the attributes of the primary key are required to determine the value of a non-primary attribute. The third normal form (3NF) requires that each primary key attribute be dependent only on the primary key and not the other non-primary key attributes. This eliminates transitive dependencies where the value of a non-primary key attribute is dependent on another non-primary key attribute.

a. JOBSEEKER table

JobseekerID	Name	Username	Password	Email	Phone	Address	profile-picture

i. Step 1: UNF to 1NF

JobseekerID	Name	Username	Password	Email	Phone	Address	profile-picture

To be in 1NF, the table must satisfy the following conditions:

- Each column should contain atomic values (no repeating groups or arrays).
- Each row should have a unique identifier (Primary Key).

The jobseeker table is already in 1NF since:

- All columns contain atomic values (e.g., no multiple values in a single column).
- Each row has a unique identifier, JobseekerID, which is the Primary Key.

ii. Step 2: 1NF to 2NF

JobseekerID	Name	Username	Password	Email	Phone	Address	profile-picture

To be in 2NF, the table must:

- Be in 1NF.
- Have no partial dependencies (i.e., non-key attributes should depend on the entire primary key).

Since the JobseekerID is the Primary Key and all other attributes depend solely on it, there are no partial dependencies. Therefore, the table remains unchanged because it is already in 2NF.

iii. Step 3: 2NF to 3NF

JobseekerID	Name	Username	Password	Email	Phone	Address	profile-picture

To be in 3NF, the table must:

- Be in 2NF.
- Have no transitive dependencies (i.e., non-key attributes should not depend on other non-key attributes).

In this table, there are no transitive dependencies because:

- All non-key attributes (Name, Username, Password, Email, Phone, Address, profile_picture) depend directly on the Primary Key JobseekerID.
- None of the non-key attributes are dependent on other non-key attributes.

Thus, the table remains unchanged because it is already in 3NF.

b. COMPANY table

CompanyID	Name	Location	Sector

i. Step 1: UNF to 1NF

CompanyID	Name	Location	Sector

To be in 1NF, the table must satisfy the following conditions:

- Each column should contain atomic values (no repeating groups or arrays).
- Each row should have a unique identifier (Primary Key).

The Company table is already in 1NF since:

- All columns contain atomic values (e.g., no multiple values in a single column).
- Each row has a unique identifier, CompanyID, which is the primary key.

ii. Step 2: 1NF to 2NF

CompanyID	Name	Location	Sector

To be in 2NF, the table must:

- Be in 1NF.
- Have no partial dependencies (i.e., non-key attributes should depend on the entire primary key).

Since CompanyID is the primary key and all other attributes depend solely on it, there are no partial dependencies. Therefore, the table remains unchanged because it is already in 2NF.

iii. Step 3: 2NF to 3NF

CompanyID	Name	Location	Sector

To be in 3NF, the table must:

- Be in 2NF.
- Have no transitive dependencies (i.e., non-key attributes should not depend on other non-key attributes).

In this table, there are no transitive dependencies because:

- All non-key attributes (Name, Location, Sector) depend directly on the primary key CompanyID.
- None of the non-key attributes are dependent on other non-key attributes.

Thus, the table remains unchanged because it is already in 3NF.

c. EMPLOYER table

EmployerID	Name	Role	Username	Password	Email	Phone	CompanyID	profile_picture

- **Step 1: UNF to 1NF**

EmployerID	Name	Role	Username	Password	Email	Phone	CompanyID	profile_picture

To be in 1NF, the table must satisfy the following conditions:

- Each column should contain atomic values (no repeating groups or arrays).
- Each row should have a unique identifier (Primary Key).

The Employer table is already in 1NF since:

- All columns contain atomic values (e.g., no multiple values in a single column).
- Each row has a unique identifier, EmployerID, which is the Primary Key.

- **Step 2: 1NF to 2NF**

EmployerID	Name	Role	Username	Password	Email	Phone	CompanyID	profile_picture

To be in 2NF, the table must:

- Be in 1NF.
- Have no partial dependencies (i.e., non-key attributes should depend on the entire primary key).

Since EmployerID is the Primary Key and all other attributes depend solely on it, there are no partial dependencies. Therefore, the table remains unchanged because it is already in 2NF.

- **Step 3: 2NF to 3NF**

EmployerID	Name	Role	Username	Password	Email	Phone	CompanyID	profile_ picture

To be in 3NF, the table must:

- Be in 2NF.
- Have no transitive dependencies (i.e., non-key attributes should not depend on other non-key attributes).

Let's check for transitive dependencies:

- EmployerID → Name, Role, Username, Password, Email, Phone, CompanyID, profile_picture

All attributes depend directly on EmployerID, and none of the non-key attributes depend on other non-key attributes. Therefore, there are no transitive dependencies.

Thus, the table remains unchanged because it is already in 3NF.

d. ADVERTISEMENT table

AdvertisementID	Title	Description	Status	Published	ExpectedSalary	CompanyID

- **Step 1: UNF to 1NF**

AdvertisementID	Title	Description	Status	Published	ExpectedSalary	CompanyID

To be in 1NF, the table must satisfy the following conditions:

- Each column should contain atomic values (no repeating groups or arrays).
- Each row should have a unique identifier (Primary Key).

The Advertisement table is already in 1NF since:

- All columns contain atomic values (e.g., no multiple values in a single column).
- Each row has a unique identifier, AdvertisementID, which is the Primary Key.

- **Step 2: 1NF to 2NF**

AdvertisementID	Title	Description	Status	Published	ExpectedSalary	CompanyID

To be in 2NF, the table must:

- Be in 1NF.
- Have no partial dependencies (i.e., non-key attributes should depend on the entire primary key).

Since AdvertisementID is the Primary Key and all other attributes depend solely on it, there are no partial dependencies. Therefore, the table remains unchanged because it is already in 2NF.

- **Step 3: 2NF to 3NF**

AdvertisementID	Title	Description	Status	Published	ExpectedSalary	CompanyID

To be in 3NF, the table must:

- Be in 2NF.
- Have no transitive dependencies (i.e., non-key attributes should not depend on other non-key attributes).

Let's check for transitive dependencies:

- AdvertisementID → Title, Description, Status, Published, ExpectedSalary, CompanyID

All attributes depend directly on AdvertisementID, and none of the non-key attributes depend on other non-key attributes. Therefore, there are no transitive dependencies.

Thus, the table remains unchanged because it is already in 3NF.

e. APPLICATIONS table

ApplicationID	applied_at	Status	resume_path	Notification	approved_at	IVDate	IVTime	Note	JobseekerID	AdvertisementID

- **Step 1: UNF to 1NF**

ApplicationID	applied_at	Status	resume_path	Notification	approved_at	IVDate	IVTime	Note	JobseekerID	AdvertisementID

To be in 1NF, the table must:

- Have atomic values in each column (no repeating groups or arrays).
- Have a unique identifier (Primary Key) for each row.

The applications table is already in 1NF if:

- All columns contain atomic values (i.e., no multiple values in a single column).
- Each row has a unique identifier. In this case, ApplicationID serves as the Primary Key.

- **Step 2: 1NF to 2NF**

ApplicationID	applied_at	Status	resume_path	Notification	approved_at	IVDate	IVTime	Note	JobseekerID	AdvertisementID

To be in 2NF, the table must:

- Be in 1NF.
- Have no partial dependencies (i.e., non-key attributes should depend on the entire primary key).

Since ApplicationID is the Primary Key and all other attributes depend solely on it, there are no partial dependencies. Therefore, the table remains unchanged because it is already in 2NF.

- **Step 3: 2NF to 3NF**

ApplicationID	applied_at	Status	resume_path	Notification	approved_at	IVDate	IVTime	Note	JobseekerID	AdvertisementID

To be in 3NF, the table must:

- Be in 2NF.
- Have no transitive dependencies (i.e., non-key attributes should not depend on other non-key attributes).

Let's check for transitive dependencies:

- ApplicationID → applied_at, Status, resume_path, Notification, approved_at, IVDate, IVTime, Note, JobseekerID, AdvertisementID

All attributes depend directly on ApplicationID, and none of the non-key attributes depend on other non-key attributes. Therefore, there are no transitive dependencies.

Thus, the table remains unchanged because it is already in 3NF.

f. RESUME table

ResumeID	Skills	Qualifications	Experience	resume_path

- **Step 1: UNF to 1NF**

ResumeID	Skills	Qualifications	Experience	resume_path

To be in 1NF, the table must:

- Have atomic values (no repeating groups or arrays).
- Have a unique identifier (Primary Key) for each row.

Assuming that each column contains atomic values, and the table already has a unique identifier (ResumeID), the resume table should be in 1NF.

- **Step 2: 1NF to 2NF**

ResumeID	Skills	Qualifications	Experience	resume_path

To be in 2NF, the table must:

- Be in 1NF.
- Have no partial dependencies (i.e., non-key attributes should depend on the entire primary key).

Since ResumeID is the Primary Key and all other attributes depend solely on it, there are no partial dependencies. Therefore, the table remains unchanged because it is already in 2NF.

- **Step 3: 2NF to 3NF**

ResumeID	Skills	Qualifications	Experience	resume_path

To be in 3NF, the table must:

- Be in 2NF.
- Have no transitive dependencies (i.e., non-key attributes should not depend on other non-key attributes).

Let's check for transitive dependencies:

- ResumeID → Skills, Qualifications, Experience, resume_path

All attributes depend directly on ResumeID, and none of the non-key attributes depend on other non-key attributes. Therefore, there are no transitive dependencies.

Thus, the table remains unchanged because it is already in 3NF.

4.3.1.2 Entity Relationship Diagram (ERD)

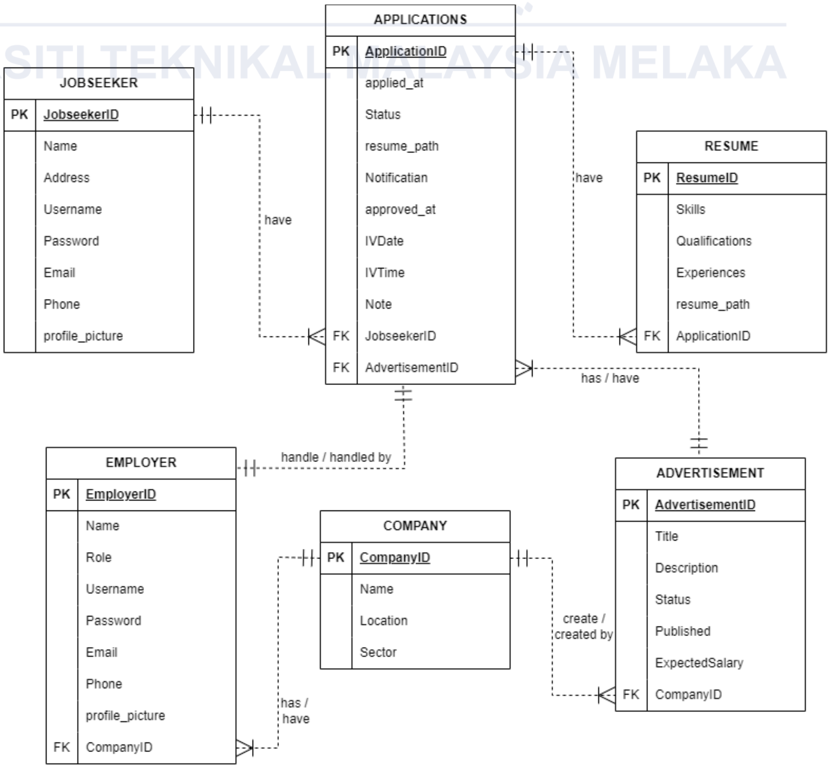


Figure 4.2: Entity Relationship Diagram

4.3.1.3 Business Rules

- A jobseeker can have one or many applications while an application must have one and only one jobseeker.

- An application can have one or many resume while a resume can have one and only one application.

- An employer can handle one and only one application while an application can handle by one and only employer.

- An employer has one and only one company while a company can have one or many employers.

- A company can create one or many advertisements while an advertisement had been created by one and only one company.

- An advertisement has one or many applications while an application can have one and only one advertisement.

4.3.2 Logical Design

In the logical design phase of the CareerNest Management System, the Entity Relationship Diagram (ERD) is converted into a relational model where entities become tables and their attributes turn into columns. Relationships between these tables are defined using foreign keys. This phase includes creating a data dictionary that details each table’s structure, columns, data types, and constraints for MySQL. Normalization is applied to reduce redundancy and ensure data integrity, while query design focuses on crafting efficient SQL queries for data retrieval and manipulation. These elements together ensure a well-structured and functional database.

4.3.2.1 Data Dictionary

JOBSEEKER table

Table 4.1: Jobseeker table

Attribute	Type	Null	Key	Description
JobseekerID	varchar(10)	NO	PK	Unique identifier for each jobseeker.
Name	varchar(200)	NO		Name of the jobseeker.
Username	varchar(200)	NO		Username for login.
Password	varchar(200)	NO		Password for login.
Email	varchar(200)	YES		Email address of the jobseeker.
Phone	varchar(200)	YES		Phone number of the jobseeker.
Address	varchar(200)	YES		Address of the jobseeker.
profile_picture	blob	YES		Profile picture of the jobseeker.

COMPANY table

Table 4.2: Company table

Attribute	Type	Null	Key	Description
CompanyID	varchar(10)	NO	PK	Unique identifier for each company.
Name	varchar(200)	NO		Name of the company.
Location	varchar(200)	NO		Location of the company.
Sector	varchar(200)	NO		Sector or industry of the company.

EMPLOYER table

Table 4.3: Employer table

Attribute	Type	Null	Key	Description
EmployerID	varchar(10)	NO	PK	Unique identifier for each employer.
Name	varchar(200)	NO		Name of the employer.
Role	varchar(200)	NO		Role of the employer.
Username	varchar(200)	NO		Username for login.
Password	varchar(200)	NO		Password for login.
Email	varchar(200)	YES		Email address of the

				employer.
Phone	varchar(200)	YES		Phone number of the employer.
CompanyID	varchar(10)	YES	FK	Identifier of the employer's company.
profile_picture	blob	YES		Profile picture of the employer.

ADVERTISEMENT table

Table 4.4: Advertisement table

Attribute	Type	Null	Key	Description
AdvertisementID	varchar(10)	NO	PK	Unique identifier for each advertisement.
Title	varchar(200)	NO		Title of the advertisement.
Description	varchar(200)	NO		Description of the advertisement.
Status	varchar(200)	NO		Current status of the advertisement.
Published	varchar(200)	NO		Publication status or date.
ExpectedSalary	varchar(200)	NO		Expected salary for the advertised job.
CompanyID	varchar(10)	NO	FK	Identifier of the company posting the advertisement.

APPLICATIONS table

Table 4.5: Applications table

Attribute	Type	Null	Key	Description
ApplicationID	varchar(10)	NO	PK	Unique identifier for each application.
applied_at	timestamp	YES		Timestamp when the application was made.
Status	varchar(200)	YES		Current status of the application.
resume_path	varchar(200)	NO		File path to the resume associated with the application.
Notification	varchar(250)	YES		Notification message related to the application status.
approved_at	datetime	YES		Date and time when the application was approved.
IVDate	date	YES		Interview date scheduled for the application.
IVTime	time	YES		Interview time scheduled for the application.

Note	varchar(255)	YES		Additional note or comments when application status change.
JobseekerID	varchar(10)	NO	FK	Identifier of the jobseeker who applied.
AdvertisementID	varchar(10)	NO	FK	Identifier of the related advertisement.

RESUME table

Table 4.6: Resume table

Attribute	Type	Null	Key	Description
ResumeID	varchar(10)	NO	PK	Unique identifier for each resume.
Skills	varchar(200)	NO		Skills listed on the resume.
Qualifications	varchar(200)	NO		Qualifications listed on the resume.
Experience	varchar(200)	NO		Work experience listed on the resume.
resume_path	varchar(200)	NO		File path to the resume document.

4.3.2.2 Query Design

Query design in CareerNest Management System involves crafting efficient SQL statements to fetch and manage data from the database. It aims to retrieve specific information like job listings or applicant details accurately and swiftly. This process optimizes query performance, ensuring quick execution even with large amounts of data. By using joins, filters, and aggregations effectively, the design ensures that users can access relevant data promptly, supporting smooth interactions and system responsiveness throughout the recruitment platform.

Table 4.7: Query table

Type of Query	Query	Explanation
Simple Query	SELECT * FROM jobseeker WHERE Username = '$username';	Retrieves all columns and rows from the jobseeker table where the Username column matches the value stored in the variable

		$username.

Join Table Query	SELECT a.ApplicationID, a.Status, a.applied_at, j.Name AS JobseekerName, j.Email, j.Phone, a.resume_path AS Resume, adv.Title AS JobTitle
	FROM applications a
	INNER JOIN jobseeker j ON a.JobseekerID = j.JobseekerID
	INNER JOIN advertisement adv ON a.AdvertisementID = adv.AdvertisementID
	WHERE adv.CompanyID = ?
	ORDER BY a.applied_at DESC;
		Fetch detailed information about job applications for a specific company, including the application status, date applied, job seeker details (name, email, phone), resume path, and job title. It joins data from the applications, jobseeker, and advertisement tables, and filters the results to only include applications for the specified company. The results are ordered by the date the applications were submitted, with the most recent first

Aggregate and Order by Query	SELECT COUNT(*) AS sectorCount FROM company WHERE Sector = 'SWASTA';
	
	SELECT COUNT(*) AS sectorCount FROM company WHERE Sector = 'KERAJAAN';
		The first query counts the number of companies in the private sector ('SWASTA'). The second query counts the number of companies in the government sector ('KERAJAAN').
		
		These queries are determining how many companies belong to the 'SWASTA' (private) sector and how many belong to the 'KERAJAAN' (government) sector. The results will show the counts as sectorCount.

4.3.3 Physical Design

The physical design refers to the actual implementation details of the database and system architecture, focusing on how data is stored, accessed, and managed for optimal performance and reliability. This involves defining the specific storage structures, such as tables, indexes, and data types, to ensure efficient data retrieval and manipulation. It also includes setting up physical storage configurations and optimizing database performance through indexing strategies and partitioning where necessary. Additionally, the physical design ensures data security by implementing encryption and access control mechanisms, as well as establishing backup and recovery procedures to safeguard against data loss. The goal is to create a robust, scalable, and secure environment that supports the seamless operation of the CareerNest Management System.

4.3.3.1 Usage of Trigger

Table 4.8: Trigger table

Trigger	Database Table	Query	Explanation

BEFORE INSERT	Jobseeker	CREATE TRIGGER `jobseeker_increment` BEFORE INSERT ON `jobseeker` FOR EACH ROW BEGIN
		INSERT INTO sequence_jobseeker VALUES (NULL);
		SET NEW.JobseekerID=CONCAT('J',LPAD (LAST_INSERT_ID (),2,'0'));
		END;
			Automatically generate a unique identifier (JobseekerID) for each new record inserted into the jobseeker table before the insertion occurs.

BEFORE INSERT	Employer	CREATE TRIGGER `emp_increment` BEFORE INSERT ON `employer` FOR EACH ROW BEGIN
		INSERT INTO sequence_employer VALUES (NULL);
			Automatically generate a unique identifier (EmployerID) for each new record

		SET
		NEW.EmployerID=CONCAT('E',LPAD
		(LAST_INSERT_ID (),2,'0'));
		END;
			inserted into the employer table before the insertion occurs.

BEFORE INSERT	Company	CREATE TRIGGER `company_increment`
		BEFORE INSERT ON `company` FOR
		EACH ROW BEGIN
		INSERT INTO sequence_company
		VALUES (NULL);
		SET
		NEW.CompanyID=CONCAT('C',LPAD
		(LAST_INSERT_ID (),2,'0'));
		END;
			Automatically generate a unique identifier (CompanyID) for each new record inserted into the company table before the insertion occurs.

4.3.3.2 Usage of Procedure

Table 4.9: Procedure table

Procedures	Query	Explanation

GetAdminData	DELIMITER $$
	CREATE DEFINER=`root`@`localhost`
	PROCEDURE `GetAdminData`(IN
	username VARCHAR(255))
	BEGIN
	SELECT * FROM employer WHERE
	Username = username;
	END$$
	DELIMITER ;
		Stored procedure GetAdminData simplifies the process of fetching data from the employer table based on a specified username parameter.

GetCompanyByID	DELIMITER $$ CREATE DEFINER=`root`@`localhost` PROCEDURE `GetCompanyByID`(IN p_id INT) BEGIN	Stored procedure GetCompanyByID provides a structured way to fetch specific company details from the database based on a given CompanyID.

	SELECT CompanyID, Name, Location,
	Sector FROM company WHERE
	CompanyID = p_id;
	END$$
	DELIMITER ;	

UpdateAdminProfile	DELIMITER $$
	CREATE DEFINER=`root`@`localhost`
	PROCEDURE `UpdateAdminProfile`(
	IN p_username VARCHAR(255),
	IN p_name VARCHAR(255),
	IN p_password VARCHAR(255),
	IN p_email VARCHAR(255),
	IN p_phone VARCHAR(255)
	)
	BEGIN
	UPDATE employer
	SET Name = p_name,
	Password = p_password,
	Email = p_email,
	Phone = p_phone
	WHERE Username = p_username;
	END$$
	DELIMITER ;
		Stored procedure UpdateAdminProfile provides a structured and secure way to update the profile details of an employer in the database, ensuring that changes to name, password, email, and phone number are applied correctly for the specified username.

4.3.3.3 Usage of Event

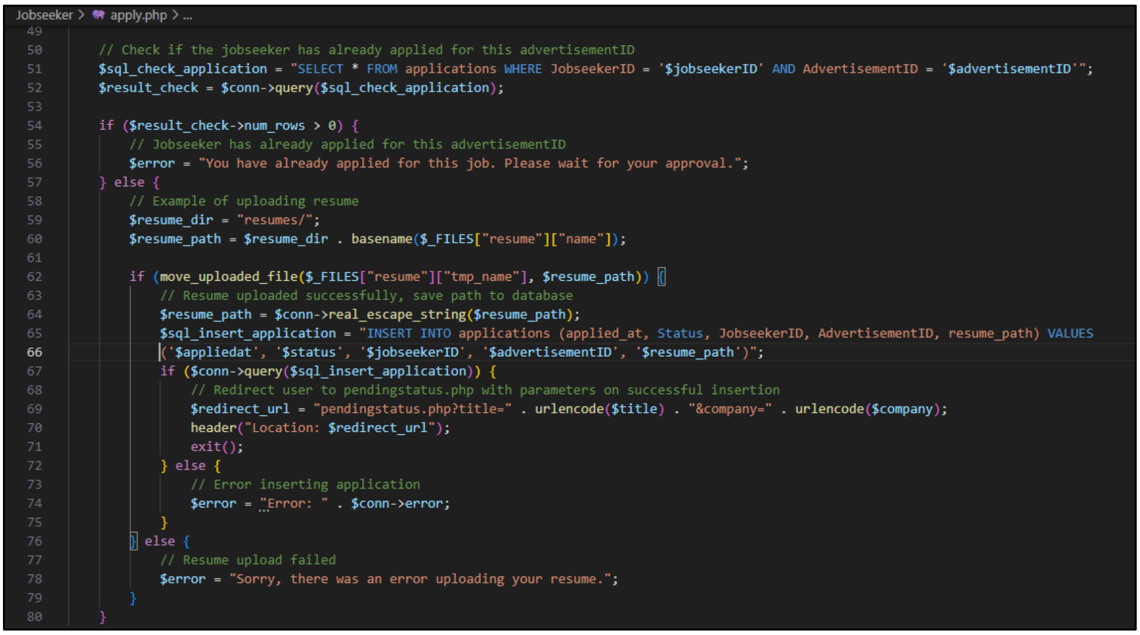
Table 4.10: Event table

Event	Query	Explanation
update_application_status	DELIMITER $$ CREATE DEFINER=`root`@`localhost` EVENT `update_application_status` ON SCHEDULE EVERY 1 DAY STARTS '2024-08-07 00:00:00' ON COMPLETION NOT PRESERVE ENABLE DO UPDATE applications	This query updates the applications table, changing the Status to 'Rejected' and the notification to 'The applications had been rejected.' for applications

	SET Status = 'Rejected', notification = 'The applications had been rejected.' WHERE Status = 'Approved' AND notification = 'No respond yet.' AND approved_at <= DATE_SUB(NOW(), INTERVAL 1 WEEK)$$ DELIMITER ;	that have the current Status as 'Approved' and notification as 'No respond yet.'. The update is triggered for applications where the approved_at date is older than one week from the current date and time (NOW()), meaning no response has been received within that time.

4.3.3.4 Security Mechanism

CareerNest Management System uses security to protect data and ensure reliable operation. It verifies users' identities with authentication, manages access with authorization controls, and encrypts sensitive data stored in the database. Access controls limit data access based on user roles, while regular audits monitor system activities for security breaches. These measures ensure data privacy and maintain the system's integrity, making CareerNest secure and dependable.



```php

    // Check if the jobseeker has already applied for this advertisementID
    $sql_check_application = "SELECT * FROM applications WHERE JobseekerID = '$jobseekerID' AND AdvertisementID = '$advertisementID'";
    $result_check = $conn->query($sql_check_application);

    if ($result_check->num_rows > 0) {
        // Jobseeker has already applied for this advertisementID
        $error = "You have already applied for this job. Please wait for your approval.";
    } else {
        // Example of uploading resume
        $resume_dir = "resumes/";
        $resume_path = $resume_dir . basename($_FILES["resume"]["name"]);

        if (move_uploaded_file($_FILES["resume"]["tmp_name"], $resume_path)) {
            // Resume uploaded successfully, save path to database
            $resume_path = $conn->real_escape_string($resume_path);
            $sql_insert_application = "INSERT INTO applications (applied_at, Status, JobseekerID, AdvertisementID, resume_path) VALUES
            ('$appliedat', '$status', '$jobseekerID', '$advertisementID', '$resume_path')";
            if ($conn->query($sql_insert_application)) {
                // Redirect user to pendingstatus.php with parameters on successful insertion
                $redirect_url = "pendingstatus.php?title=" . urlencode($title) . "&company=" . urlencode($company);
                header("Location: $redirect_url");
                exit();
            } else {
                // Error inserting application
                $error = "Error: " . $conn->error;
            }
        } else {
            // Resume upload failed
            $error = "Sorry, there was an error uploading your resume.";
        }
    }
```

Figure 4.3: Checking for duplicate applications

In Figure 4.3, the provided PHP code snippet manages the job application process for job seekers on a website. The script first checks whether the job seeker has already applied for the job by querying the database. This is done using a SQL query that searches the `applications` table for any records that match the current job seeker's ID (`$jobseekerID`) and the job advertisement ID (`$advertisementID`). If the query returns any results (`$result_check->num_rows > 0`), it indicates that the job seeker has already applied for the job, and an error message is set to inform the user that they have already applied for this position.

If the job seeker has not previously applied, the script proceeds to handle the upload of the job seeker's resume. The code designates a directory (`resumes/`) where all resumes are stored, and constructs the full path for the new file using the name of the uploaded file (`$_FILES["resume"]["name"]`). The `move_uploaded_file()` function is then used to transfer the resume from its temporary upload location to the specified directory. If the upload is successful, the file path is sanitized using `real_escape_string()` to prevent SQL injection attacks, ensuring that the file path is safe to insert into the database.

Following a successful upload, the script constructs a SQL `INSERT` statement to add the application details to the `applications` table in the database. The data inserted includes the time of application (`$appliedat`), the application status (`$status`), the job seeker's ID, the job advertisement ID, and the path to the uploaded resume. The script then attempts to execute this SQL query. If the insert operation is successful, the user is redirected to a status page (`pendingstatus.php`) that shows the status of their application, with the title and company information passed as URL parameters. This redirection confirms that the application has been successfully submitted.

4.4 Graphical User Interface (GUI) Design

The GUI (Graphical User Interface) of the CareerNest Management System is user-friendly and intuitive for job seekers, employers, and administrators. It has a clean layout with clear menus and buttons. Job seekers can easily search and apply for jobs, manage profiles, and track applications. Employers can post jobs, manage applications, and view candidate profiles. Administrators can oversee operations, manage user accounts, and monitor performance. The design works well on different devices and includes accessibility features. Overall, the GUI makes the platform easy to use and visually appealing.

4.4.1 Navigation Design

Navigation design refers to the process of creating the elements and structure that allow users to move through a website or application efficiently and intuitively. It involves designing menus, links, buttons, and other interface components that guide users to find the information they need or perform desired actions. Effective navigation design ensures that users can easily understand and use the system, enhancing their overall experience and satisfaction. Key aspects of navigation design include clarity, consistency, accessibility, and responsiveness, making it crucial for user-friendly interfaces in any digital platform.

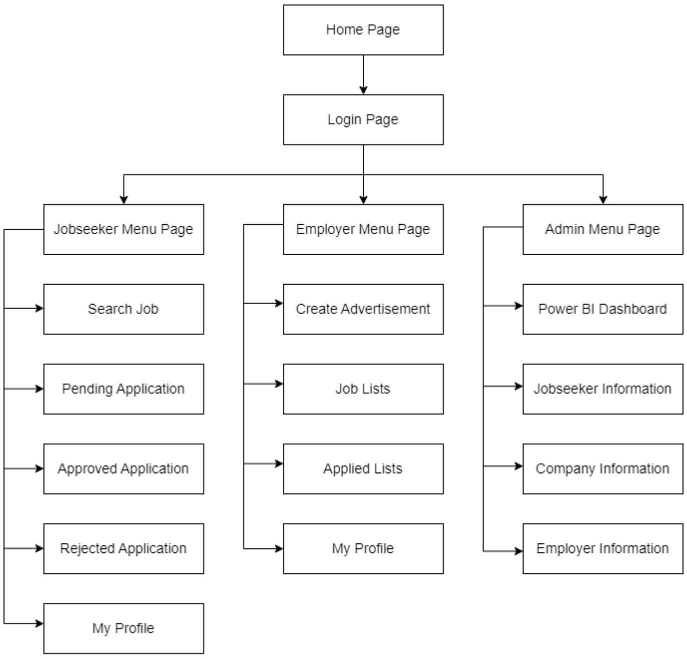
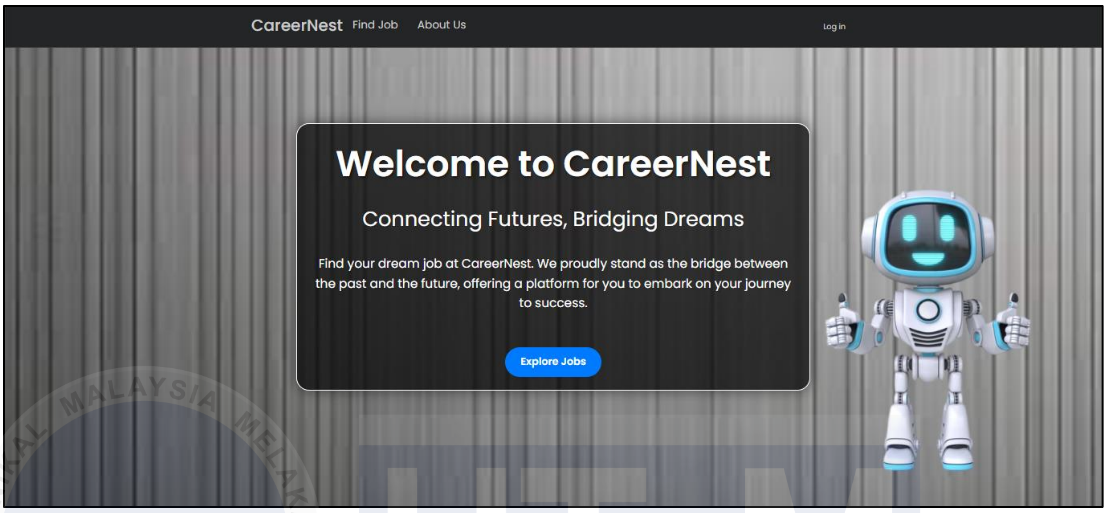


Figure 4.4: CareerNest Navigation Design

4.4.2 Input and Output Design

4.4.2.1 Display of Information



Welcome to CareerNest

Connecting Futures, Bridging Dreams

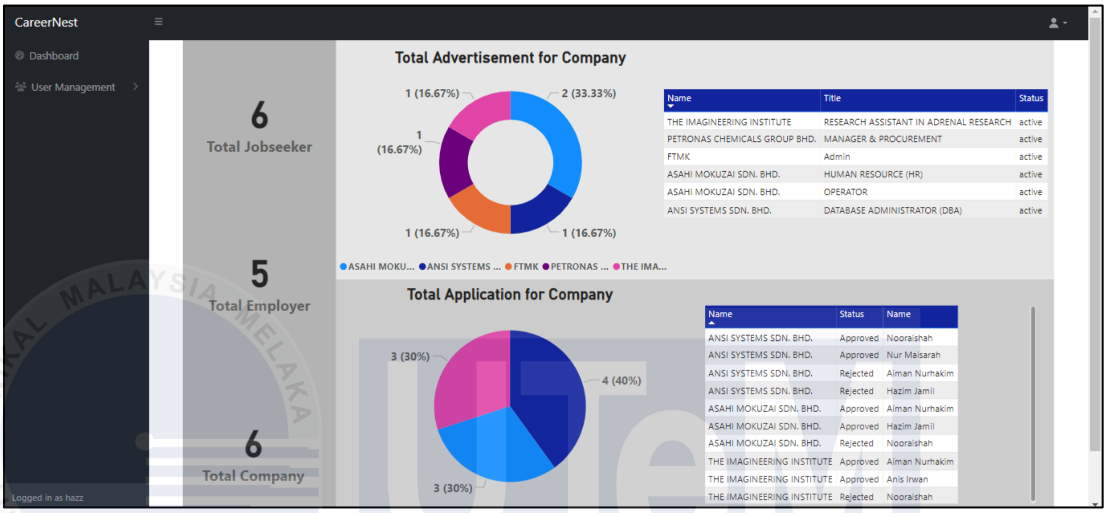
Find your dream job at CareerNest. We proudly stand as the bridge between the past and the future, offering a platform for you to embark on your journey to success.

Figure 4.5: CareerNest Dashboard

The CareerNest dashboard shows in Figure 4.5 presents a unified homepage for all users before they log in, ensuring a consistent and intuitive user experience. At the core of this homepage is a welcoming interface that introduces CareerNest and its primary benefits for job seekers and employers alike. This section sets the tone by highlighting the platform's mission and the value it provides, encouraging users to explore further.

Prominently featured is the "Find Job" section, which enables users to search for job opportunities efficiently. With search bar to make user easier to search for their desired job. This functionality is designed to streamline the job search process, making it easy and efficient for users. Moreover, the "About Us" page provides comprehensive information about CareerNest, including its mission, values, and the team behind the platform. This section builds credibility and trust by offering insight into the platform's goals and the people who drive its success.

Finally, a prominently displayed login button directs users to the login page, where they can enter their credentials to access their personalized dashboard. This button is strategically placed for easy and quick access, ensuring returning users can effortlessly log in and continue their job search or recruitment activities. By incorporating these essential elements, the CareerNest homepage offers a streamlined and user-friendly navigation experience, encouraging visitors to engage with the platform's features and take the next step in their job search or hiring process.



Name	Title	Status
THE IMAGINEERING INSTITUTE	RESEARCH ASSISTANT IN ADRENAL RESEARCH	active
PETRONAS CHEMICALS GROUP BHD.	MANAGER & PROCUREMENT	active
FTMK	Admin	active
ASAHI MOKUZAI SDN. BHD.	HUMAN RESOURCE (HR)	active
ASAHI MOKUZAI SDN. BHD.	OPERATOR	active
ANSI SYSTEMS SDN. BHD.	DATABASE ADMINISTRATOR (DBA)	active

Name	Status	Name
ANSI SYSTEMS SDN. BHD.	Approved	Nooraishah
ANSI SYSTEMS SDN. BHD.	Approved	Nur Maisarah
ANSI SYSTEMS SDN. BHD.	Rejected	Aiman Nurhakim
ANSI SYSTEMS SDN. BHD.	Rejected	Hazim Jamil
ASAHI MOKUZAI SDN. BHD.	Approved	Aiman Nurhakim
ASAHI MOKUZAI SDN. BHD.	Approved	Hazim Jamil
ASAHI MOKUZAI SDN. BHD.	Rejected	Nooraishah
THE IMAGINEERING INSTITUTE	Approved	Aiman Nurhakim
THE IMAGINEERING INSTITUTE	Approved	Anis Irwan
THE IMAGINEERING INSTITUTE	Rejected	Nooraishah

Figure 4.6: Admin Dashboard

The admin dashboard for the CareerNest Management System, as shown in Figure 4.6 is an essential tool for administrators to monitor and manage system activities effectively. Designed using Power BI, the dashboard offers a visually intuitive overview of key metrics, ensuring that admins have access to crucial information immediately.

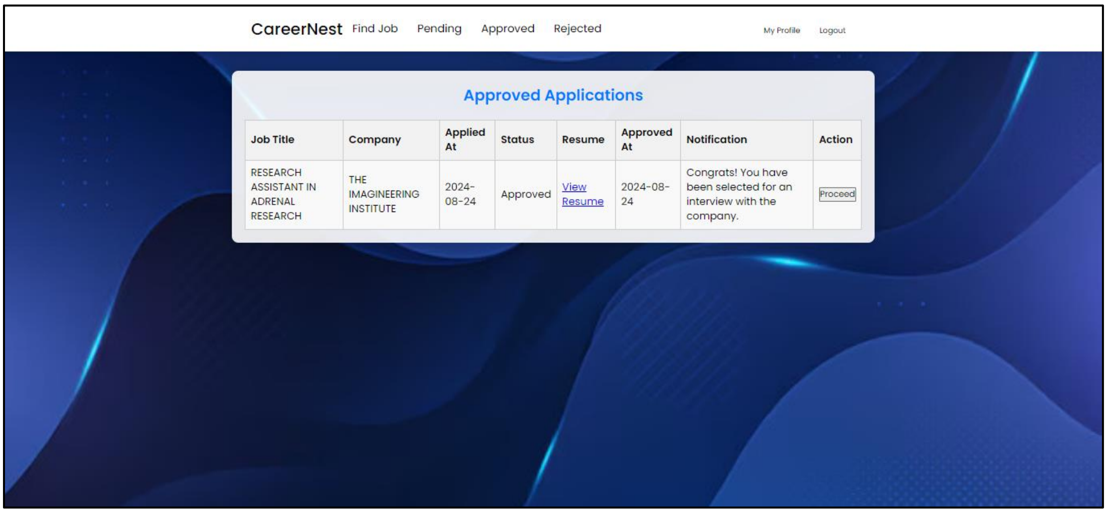
On the left side of the dashboard, summary tiles display the total number of job seekers, companies, and employers registered on the platform. These figures provide a quick snapshot of the user base, enabling administrators to gauge the level of engagement and activity within the system.

The right section of the dashboard is divided into two main parts. The first part, labeled "Advertisement by Status," features a pie chart that categorizes job advertisements based on their status, either active or inactive. This visual representation allows administrators to quickly assess the proportion of active versus inactive job postings. Accompanying the chart is a detailed list of advertisements, including their titles and statuses, which helps in monitoring the availability and activity of job postings on the platform.

The second part, "Application by Status," includes another pie chart that illustrates the status of job applications, whether they are approved, pending, or rejected. This chart provides insights into the distribution of application outcomes, helping administrators understand the flow and processing of job applications. Below this chart, a detailed table lists the names of job seekers, the titles of the jobs they have applied for, and the status of their applications. This detailed information enables administrators to track individual application statuses and manage the approval process efficiently.

Overall, the Power BI-driven admin dashboard is a powerful tool for the CareerNest Management System, offering a clear and organized interface for administrators. By providing insightful data visualization, it enhances decision-making and ensures that the platform operates smoothly and effectively.

4.4.2.2 Feedback and Notification



Approved Applications

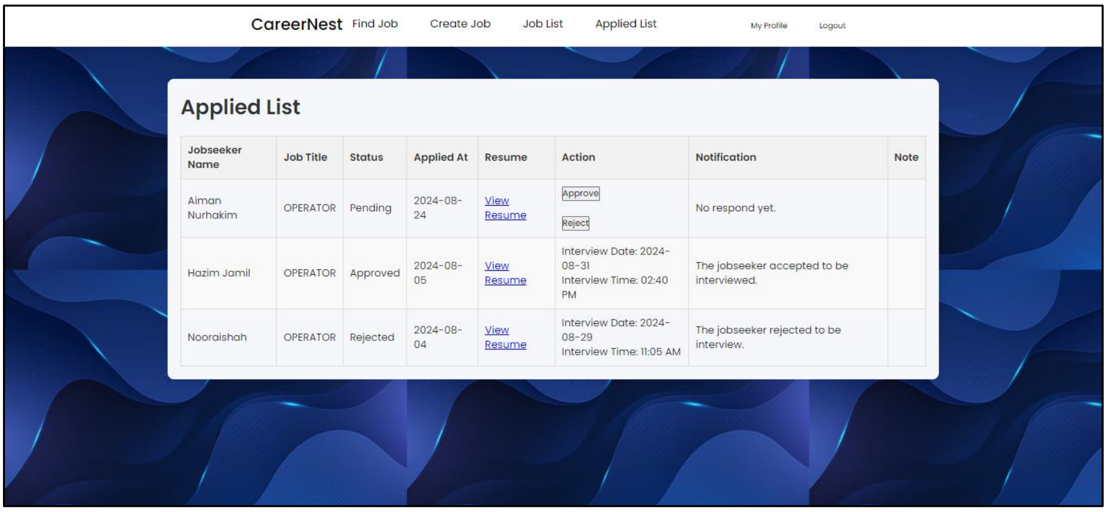
Job Title	Company	Applied At	Status	Resume	Approved At	Notification	Action
RESEARCH ASSISTANT IN ADRENAL RESEARCH	THE IMAGINEERING INSTITUTE	2024-08-24	Approved	View Resume	2024-08-24	Congrats! You have been selected for an interview with the company.	Proceed

Figure 4.7: Notification of Approval Application

The Figure 4.7 shows the approved applications page of the CareerNest Management System provides job seekers with clear feedback and notification regarding the status of their job applications. This interface is designed to enhance user experience by offering an organized and intuitive layout. At the top of the page, users can navigate between different statuses of their applications through the menu options which are Pending, Approved, and Rejected. This navigation bar ensures that job seekers can easily access the specific information they need.

In the main content area, the Figure 4.7 displays a table listing the details of approved job applications. The table includes columns for the job title, company name, application date and time, status, resume, and action proceed button. The status column clearly indicates that the application has been approved. The "Resume" column contains a link labeled "View Resume," allowing the user to easily access and review the resume submitted for the application. The "Action" column provides a "Proceed" button, enabling the user to go for further information which is the jobseeker will know the date and time for interview session with the company.

Overall, this page provides job seekers with immediate feedback on the status of their job applications, ensuring they are informed and can take appropriate actions based on the application's outcome. The clear and structured layout enhances the user experience by making it simple to navigate and manage their job search activities.



Applied List

Jobseeker Name	Job Title	Status	Applied At	Resume	Action	Notification	Note
Aiman Nurhakim	OPERATOR	Pending	2024-08-24	View Resume	Approve Reject	No respond yet.	
Hazim Jamil	OPERATOR	Approved	2024-08-05	View Resume	Interview Date: 2024-08-31 Interview Time: 02:40 PM	The jobseeker accepted to be interviewed.	
Nooraishah	OPERATOR	Rejected	2024-08-04	View Resume	Interview Date: 2024-08-29 Interview Time: 11:05 AM	The jobseeker rejected to be interview.	

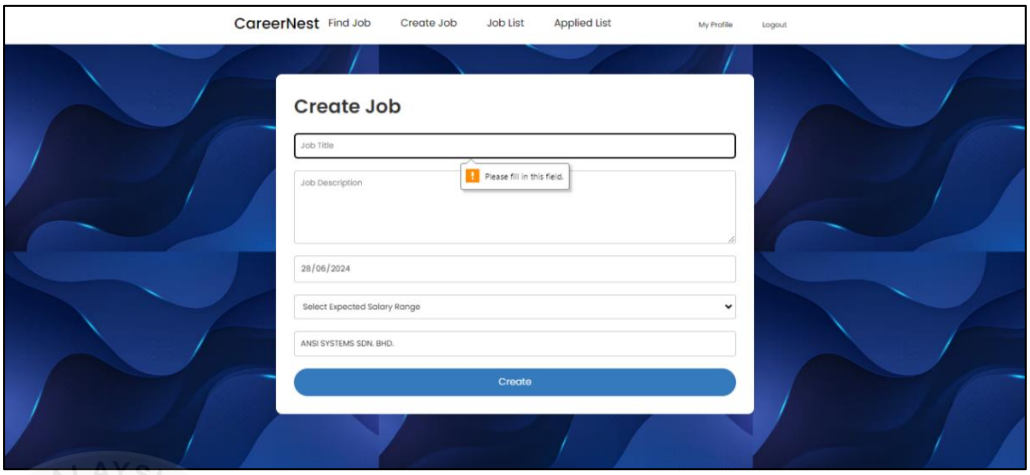
Figure 4.8 Notification of Jobseeker's Applied List

At the heart of the page is the "Applied List" table shows in Figure 4.8, which presents a comprehensive overview of all job applications in a clear and organized format. Each row in the table represents a distinct application submitted by a job seeker, with columns displaying relevant information. The "Jobseeker Name" column lists the names of the individuals who have applied for jobs, allowing employers to easily identify each applicant. Adjacent to this, the "Job Title" column specifies the position for which the job seeker has applied, providing context about the job role, such as the "OPERATOR" position shown for all applicants in the image.

The "Status" column indicates the current status of each application, such as "Pending," "Approved," or "Rejected," helping employers quickly assess the progress of each candidate. This is followed by the "Applied At" column, which shows the date each application was submitted, providing a chronological perspective on the influx of applications. Next is the "Resume" column, which includes links labelled "View Resume," enabling employers to directly access the resumes submitted by job seekers. This functionality allows for a streamlined review process where employers can quickly evaluate the qualifications of each candidate.

The "Action" column is particularly interactive, as it provides buttons for employers to either "Approve" or "Reject" applications, thereby updating the application status based on the review. For approved candidates, the column also displays scheduled interview dates and times, reflecting the next steps in the recruitment process. The "Notification" column offers further insights, such as updates on whether a job seeker has responded to an interview invitation or their acceptance of the interview, providing employers with feedback on the candidate’s engagement. Lastly, the "Note" column includes any additional remarks, such as the job seeker's decisions or comments from the employer, adding context to the application status.

4.4.2.3 Error Handling



Create Job

Job Title

Please fill in this field.

Job Description

28/06/2024

Select Expected Salary Range

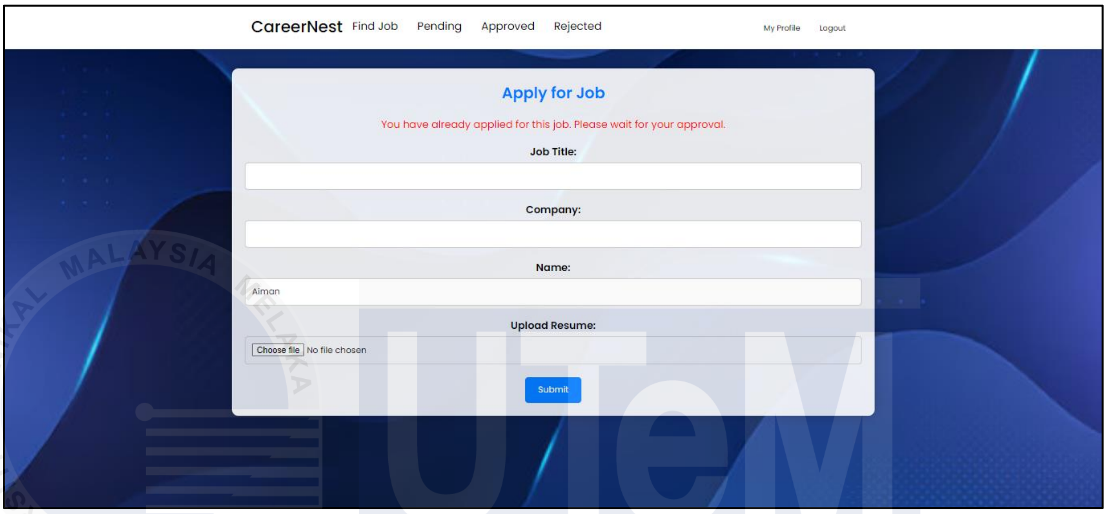
ANSI SYSTEMS SDN. BHD.

Figure 4.9: Error Handling at Create Job Form

Error handling is a crucial aspect of any web form, ensuring that users provide all necessary information before submission. In the "Create Job" form on the CareerNest Management System shown in Figure 4.9, certain fields are marked as mandatory. The job title field, for instance, is a required field that must be filled out to proceed with the job creation process. When an employer attempts to submit the form without entering a job title, the system immediately flags the omission and displays an error message: "Please fill in this field."

This real-time feedback mechanism is integral to the user experience, as it helps users understand exactly what is missing or incorrect in their submission. By highlighting the specific field that needs attention, the system reduces the likelihood of incomplete or erroneous job postings. This not only ensures that employers provide all the necessary details but also helps maintain the quality and completeness of the job listings on the platform. This approach allows the form to check for required fields and other validation rules before it is submitted to the server. If any validation rules are not met, the form prevents submission and displays relevant error messages. This method enhances user experience by providing immediate feedback and reducing the need for multiple form submissions.

Moreover, the form includes other fields such as job description, date, expected salary range, and company name, which might also have similar validation rules. Ensuring that all these fields are appropriately filled helps employers create detailed and informative job postings, which in turn attract well-suited candidates. The clear and concise error messages guide the user through the form completion process, making it more efficient and less prone to errors.



Apply for Job

You have already applied for this job. Please wait for your approval.

Job Title:

Company:

Name: Aiman

Upload Resume: Choose file No file chosen

Figure 4.10: Error Handling at Application Form

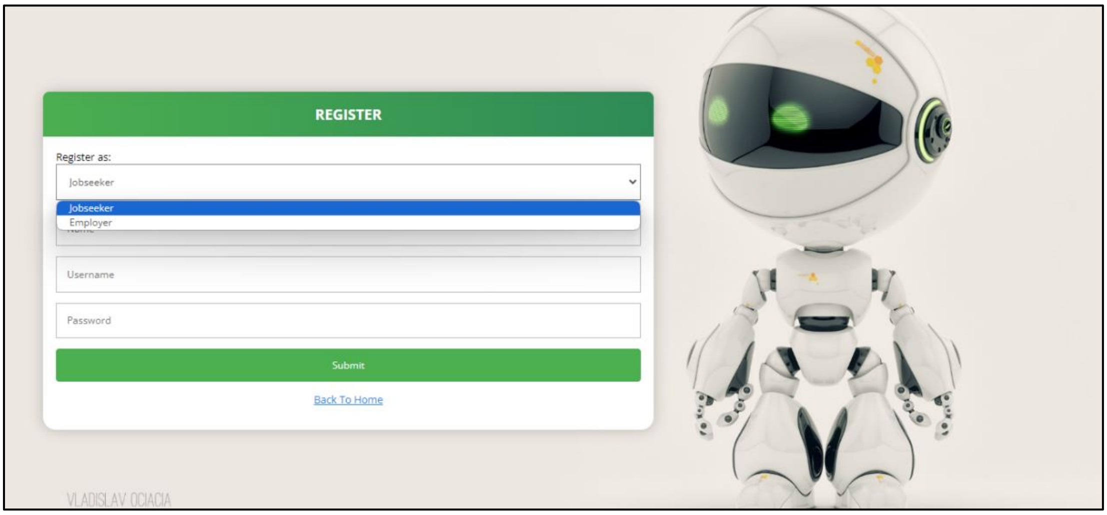
Error handling in the CareerNest Management System shown in Figure 4.10 plays a pivotal role in providing a seamless user experience and preserving the integrity of the job application process. One notable example is the handling of application duplication, which is effectively managed by the system. When a jobseeker attempts to apply for a job they have already applied before, the system performs a check to determine if an existing application is already in the database for that job and user. This is achieved through a database query that cross-references the jobseeker's profile with the job ID. If the system finds a matching record, it recognizes that the user has previously applied for this job.

Upon detecting a duplicate application attempt, the system prevents the new application from being submitted and promptly displays an informative message to the user: "You have already applied for this job. Please wait for your approval." This message serves multiple purposes. Firstly, it informs the jobseeker that their initial application is still under review and there is no need to reapply. Secondly, it prevents the user from wasting time and effort on a redundant application process.

This error handling mechanism ensures clarity for both jobseekers and employers. For jobseekers, it provides a clear indication of their application status, reducing confusion and frustration. For employers, it maintains a streamlined application process by preventing multiple submissions for the same position, which could otherwise clutter the system and complicate the review process. Moreover, this approach helps maintain the integrity of the application data within the system. By preventing duplicate entries, the system avoids potential inconsistencies and ensures that each job application is unique and accurately tracked. This is particularly important for generating reliable reports and analytics on application metrics.

In summary, the error handling mechanism for application duplication in the CareerNest Management System effectively prevents multiple applications for the same job. It enhances user experience by providing clear feedback, maintains the integrity of application data, and streamlines the process for both jobseekers and employers. This feature exemplifies the importance of robust error handling in creating a user-friendly and reliable job application system.

4.4.2.4 User Registration and Login Module



REGISTER

Register as: Jobseeker

Figure 4.11: Registration Form

The user registration and login module of the CareerNest Management System is a fundamental component designed to manage user access and roles within the system. The Figure 4.11 provided depicts the registration interface, where new users can sign up as either jobseekers or employers. This distinction is crucial because it tailors the user experience and access privileges according to their role.

Upon accessing the registration page, users are prompted to select their role from a dropdown menu, choosing between "Jobseeker" and "Employer." This selection determines the kind of profile the user will create. Jobseekers are individuals looking for employment opportunities, while employers are companies or recruiters posting job vacancies. This role-based registration ensures that the system can cater to the specific needs and functionalities required by each user type.

The registration form requires users to enter their name, create a username, and set a password. These credentials are essential for the system to authenticate users during subsequent logins. The simplicity of the form ensures that users can quickly register without unnecessary complexity, which enhances the user experience. Once the form is filled out and the user clicks the "Submit" button, the system processes the information, creating a new user profile in the database.

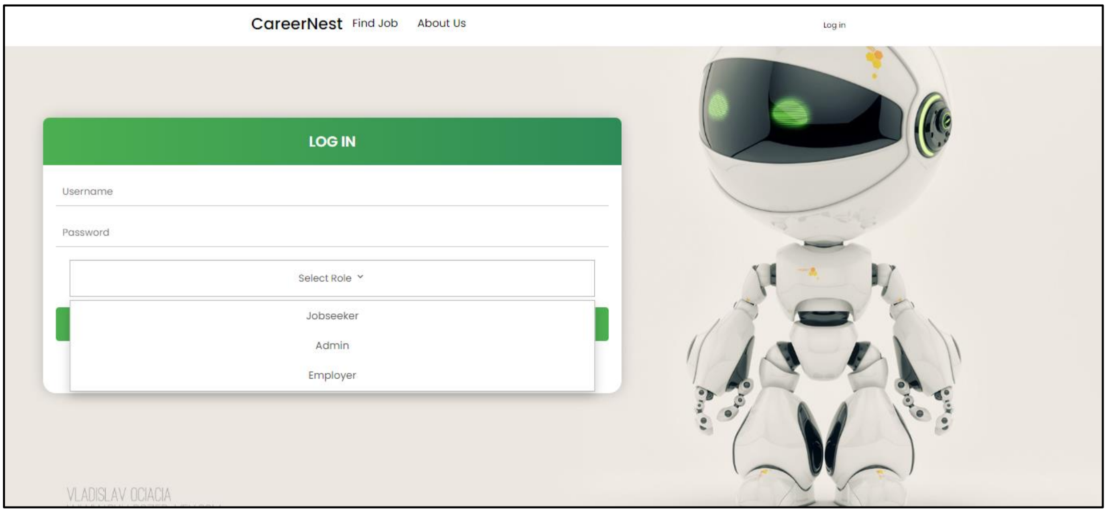


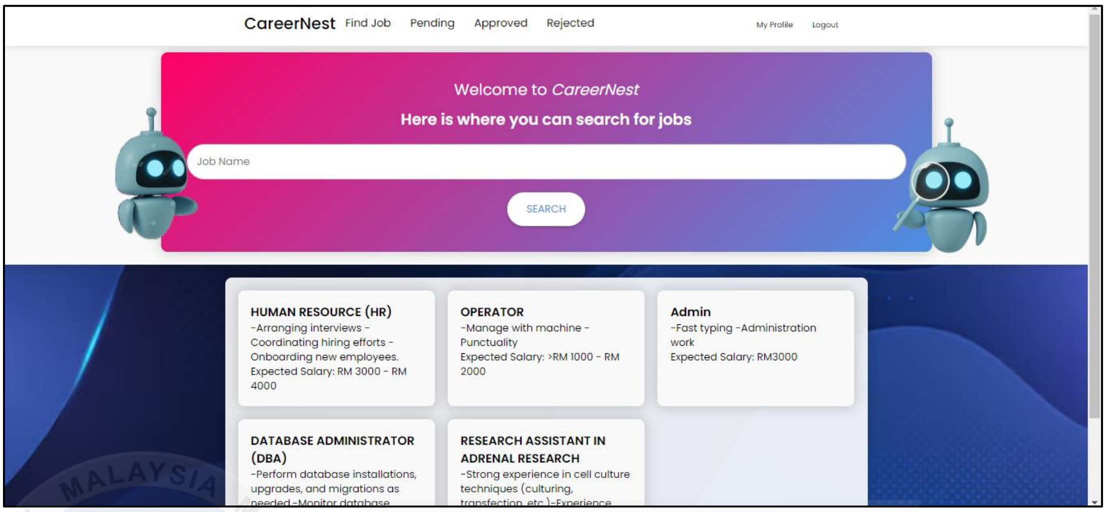
Figure 4.12: Login Form

The Figure 4.12 displays the login page for "CareerNest," a website that seems dedicated to job searching and career-related services. The page is part of the site's registration and login module, providing users a straightforward interface to access their accounts. The CareerNest Management System is prominently displayed at the top, and links like "Find Job" and "About Us" are available in the navigation bar, indicating that the site likely offers various job search functionalities and information about the platform.

The central element of the page is the login form, which consists of fields for entering a username and password, along with a dropdown menu for selecting the user's role. The dropdown menu offers three options: "Jobseeker," "Admin," and "Employer," suggesting that the platform supports different types of users, each with specific access levels and functionalities. This design indicates a role-based access control system, where each role may have different permissions and features available upon logging in. For example, jobseekers might have access to job listings and application tools, employers could post job openings and review applicants, and admins would manage the overall site operations.

Overall, this login module likely functions by validating the entered credentials against a database and granting access according to the selected role. Users can log in by entering their username, password, and selecting their role, which then directs them to their respective dashboards or interfaces based on their account type. The registration module on the site would mirror this process, allowing new users to create accounts by providing their details and choosing their role, thereby setting up their access level within the CareerNest platform.

4.4.2.5 Jobseeker Module



Welcome to *CareerNest*

Here is where you can search for jobs

HUMAN RESOURCE (HR)
-Arranging interviews -Coordinating hiring efforts -Onboarding new employees.
Expected Salary: RM 3000 - RM 4000

OPERATOR
-Manage with machine -Punctuality
Expected Salary: >RM 1000 - RM 2000

Admin
-Fast typing -Administration work
Expected Salary: RM3000

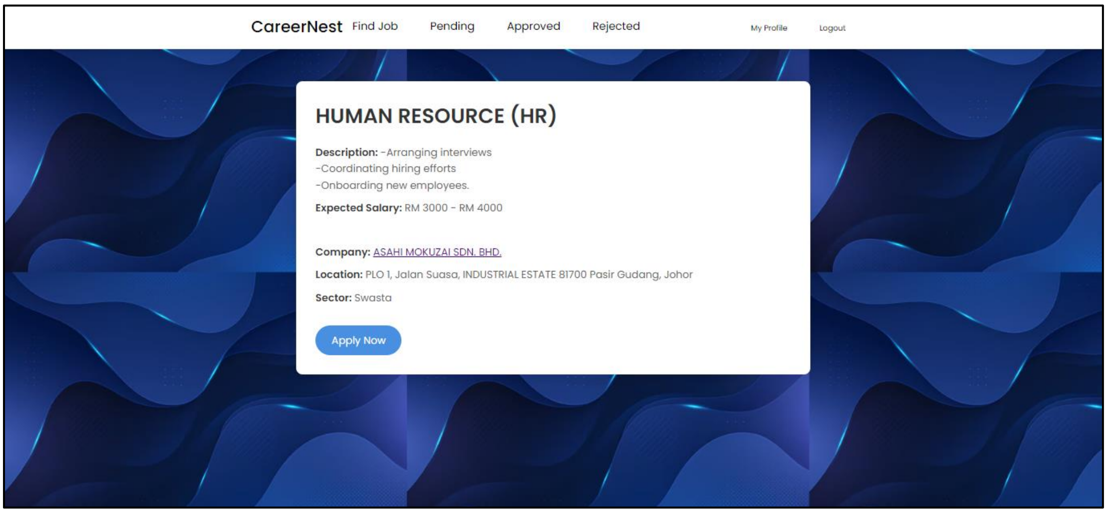
DATABASE ADMINISTRATOR (DBA)
-Perform database installations, upgrades, and migrations as needed -Monitor database

RESEARCH ASSISTANT IN ADRENAL RESEARCH
-Strong experience in cell culture techniques (culturing, transfection, etc.)-Experience

Figure 4.13: Find Job Page

The Figure 4.13 represents the find job page in the job seeker module of the "CareerNest" website, a platform designed to assist job seekers in finding and applying for job opportunities. At the top of the page, a welcoming message greets users with the statement, "Welcome to CareerNest. Here is where you can search for jobs," setting a positive and inviting tone. Central to the page is a prominent search bar labeled "Job Name," allowing users to input specific job titles or roles they are interested in. This search bar is complemented by a "SEARCH" button that initiates the job search, making it easy for users to quickly find relevant job listings.

The main content area is dedicated to displaying job listings. Each listing provides essential details, including the job title, a brief description of the role, and the expected salary range. For example, the listing for "Human Resource (HR)" outlines responsibilities such as arranging interviews, coordinating hiring efforts, and onboarding new employees, with an expected salary range of RM 3000 - RM 4000. Other job listings, like "Operator" and "Admin," similarly provide concise role descriptions and salary expectations, helping job seekers quickly assess which opportunities might suit their skills and salary requirements.



HUMAN RESOURCE (HR)

Description: -Arranging interviews
-Coordinating hiring efforts
-Onboarding new employees.

Expected Salary: RM 3000 - RM 4000

Company: ASAHI MOKUZAI SDN. BHD.

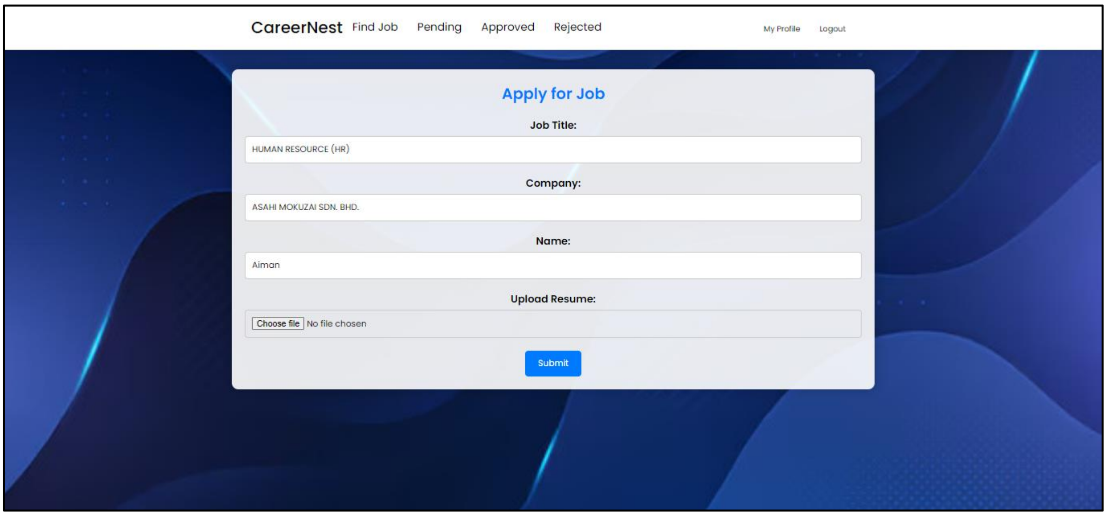
Location: PLO 1, Jalan Suasa, INDUSTRIAL ESTATE 81700 Pasir Gudang, Johor

Sector: Swasta

Figure 4.14: Job Details Page

The Figure 4.14 illustrates a detailed job listing page on the CareerNest Management System, which is tailored to provide job seekers with comprehensive information about specific job opportunities. At the top of the page, the job title "HUMAN RESOURCE (HR)" is prominently featured, immediately capturing the user's attention. Following the title, the job description outlines key responsibilities such as arranging interviews, coordinating hiring efforts, and onboarding new employees. This level of detail helps job seekers quickly understand the core duties and expectations associated with the position.

Beneath the job description, the expected salary range is clearly stated as RM 3000 - RM 4000. This transparency regarding compensation is essential, as it allows potential applicants to assess whether the offered salary aligns with their financial expectations and needs. By providing this information upfront, CareerNest helps streamline the decision-making process for job seekers, enabling them to focus on opportunities that meet their requirements. The company offering the job is identified as "ASAHI MOKUZAI SDN. BHD.," with a hyperlink included for further exploration. This link likely leads to the company's profile or official website, allowing candidates to research the company’s background, culture, and values. A prominently displayed "Apply Now" button serves as a call-to-action, inviting interested candidates to proceed with their application.



Apply for Job

Job Title: HUMAN RESOURCE (HR)

Company: ASAHI MOKUZAI SDN. BHD.

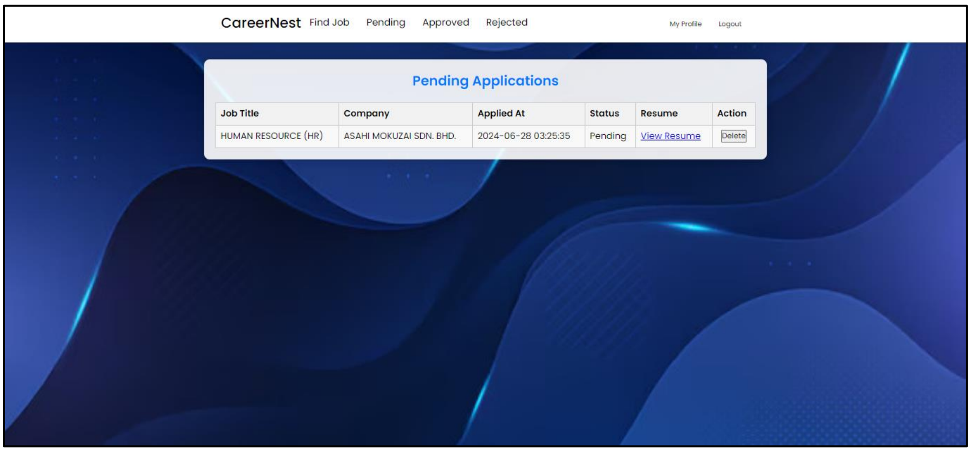
Name: Aiman

Upload Resume: Choose file No file chosen

Figure 4.15: Application Form Page

This application form shown in Figure 4.15 is a form where the jobseeker only needs to upload their resume since the job title, company and their name are already fetched from job details page and login user. It allows applicants to submit their information and resume for a specific position. Upon filling out the form, applicants typically need to upload their resume and click a submit button.

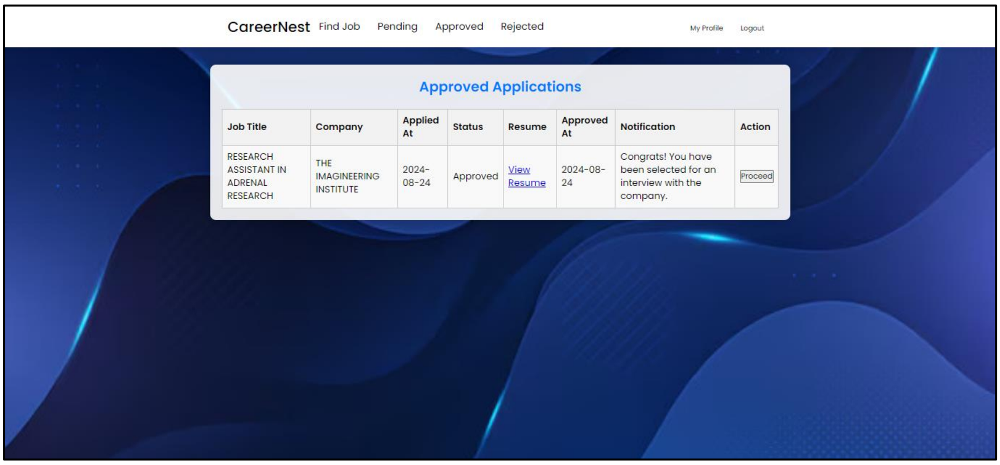
Once submitted, the application most likely transitions to a "pending" status where will appear in the Pending Application Page as shown in Figure 4.16 This status indicates the application has been received but not yet reviewed by the employer. During this pending period, the employer reviews applications and selects candidates for the next stage of the hiring process. If the employer had reviewed the resume and decided to approve the jobseeker's resume, it will appear in the Approved Application Page shown in Figure 4.17.



Pending Applications

Job Title	Company	Applied At	Status	Resume	Action
HUMAN RESOURCE (HR)	ASAHI MOKUZAI SDN. BHD.	2024-06-28 03:25:35	Pending	View Resume	Delete

Figure 4.16: Pending Application Page



Approved Applications

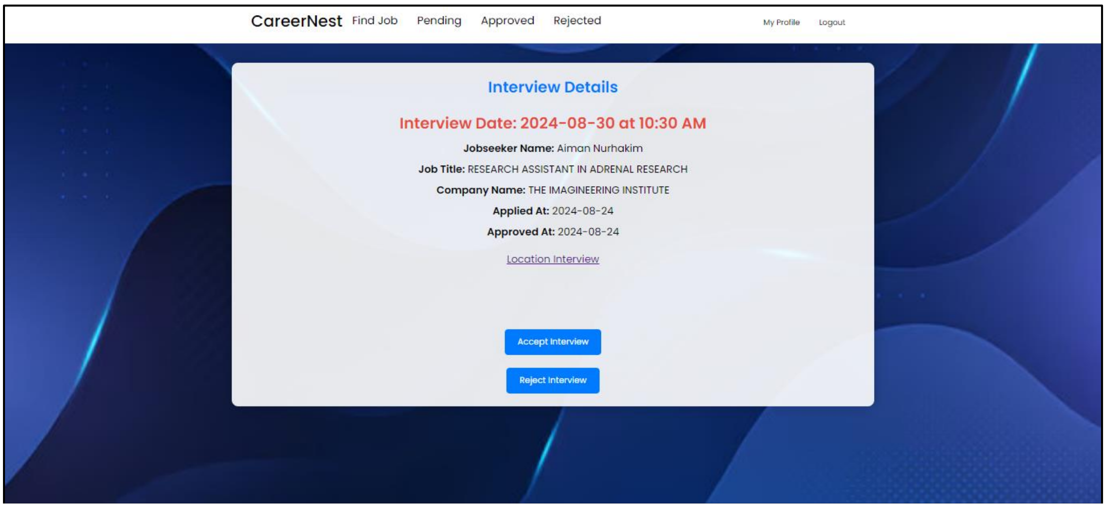
Job Title	Company	Applied At	Status	Resume	Approved At	Notification	Action
RESEARCH ASSISTANT IN ADRENAL RESEARCH	THE IMAGINEERING INSTITUTE	2024-08-24	Approved	View Resume	2024-08-24	Congrats! You have been selected for an interview with the company.	Proceed

Figure 4.17: Approved Application Page

Next, from the Figure 4.17, the jobseeker will do action proceed button then it will navigate to the Interview Details Page in Figure 4.18. In the Interview Details Page shows in Figure 4.18 jobseekers are presented with essential information about their upcoming interview. This page includes details such as the interview date and time, the jobseeker's name, the job title for which they applied, the company name, and key dates related to their application, including when they applied and when their application was approved.

At the bottom of the page, jobseekers are provided with two action buttons. The first button, labelled "Accept Interview," allows the jobseeker to confirm their attendance. Upon clicking this button, a message is displayed: “Thank you for responding. Please come to the interview punctually and wear neatly.” This message serves as a reminder to the jobseeker to be on time and present themselves appropriately.

The second button, "Reject Interview," enables the jobseeker to decline the interview offer. Clicking this button automatically updates the status of the application from "Approved" to "Rejected." This change reflects the jobseeker's decision not to proceed with the interview, ensuring that the application status is accurately maintained in the system.



CareerNest Find Job Pending Approved Rejected My Profile Logout

Interview Details

Interview Date: 2024-08-30 at 10:30 AM

Jobseeker Name: Aiman Nurhakim

Job Title: RESEARCH ASSISTANT IN ADRENAL RESEARCH

Company Name: THE IMAGINEERING INSTITUTE

Applied At: 2024-08-24

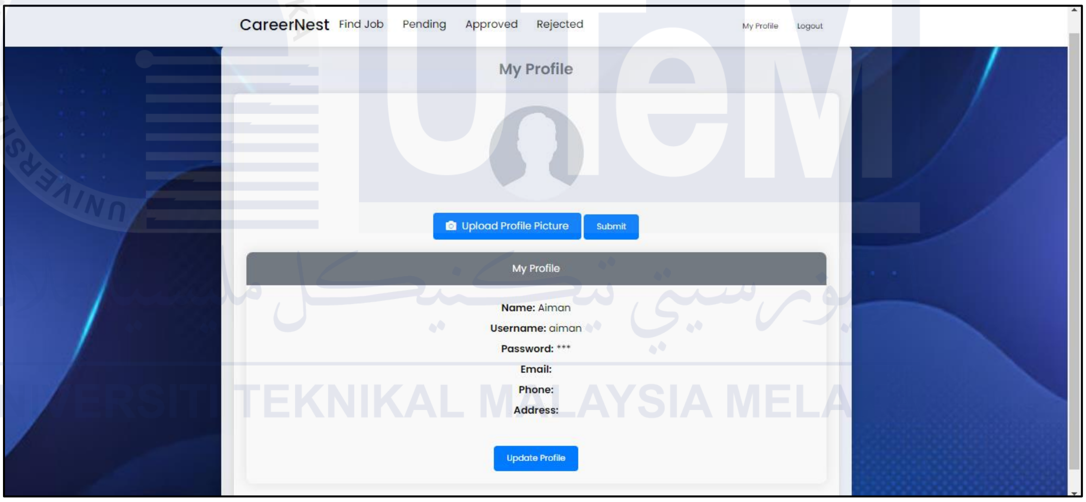
Approved At: 2024-08-24

Location Interview

Accept Interview

Reject Interview

Figure 4.18: Interview Details Page



CareerNest Find Job Pending Approved Rejected My Profile Logout

My Profile

Upload Profile Picture Submit

My Profile

Name: Aiman

Username: aiman

Password: ***

Email:

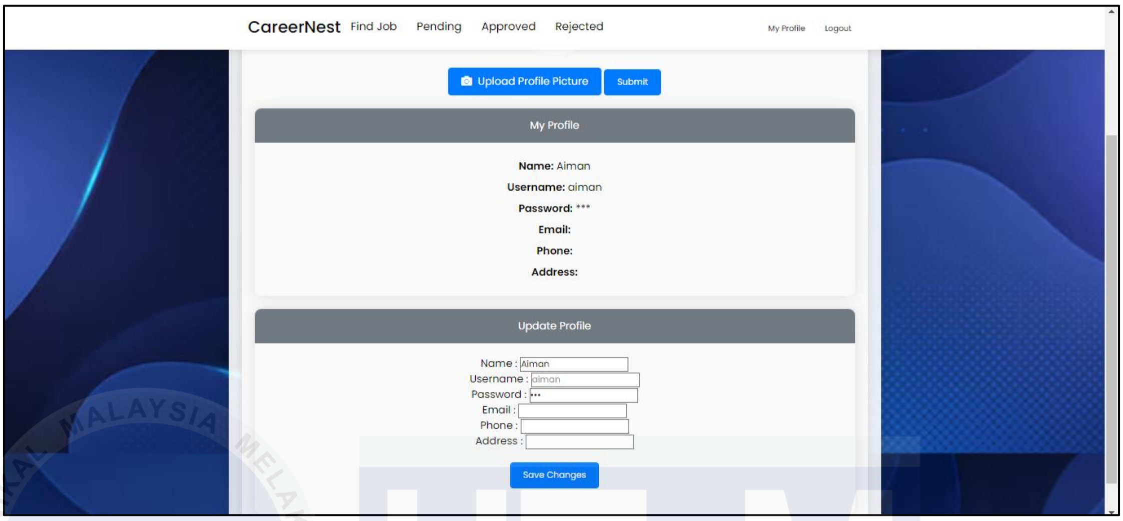
Phone:

Address:

Update Profile

Figure 4.19: Profile Page

This image shows in Figure 4.19 allows jobseekers to manage and update their personal information. The profile allows jobseeker to edit details like name, password, contact details and even upload a profile picture to create a more professional and attractive online presence. By keeping this profile updated, it will ensure employers can easily find details and learn more about jobseeker qualifications. The blank information such as email, phone and address refer that information is a null that have been set in the database. Thus, the name, username and password are required in the signup session. The username cannot update since it is a fix information, but the remains information can edit. If the username wants to be edit by the jobseeker, they need to call admin to edit their username. Once a user makes their edits, they can click the "Save Changes" button to update their profile information that illustrated in Figure 4.20.



My Profile

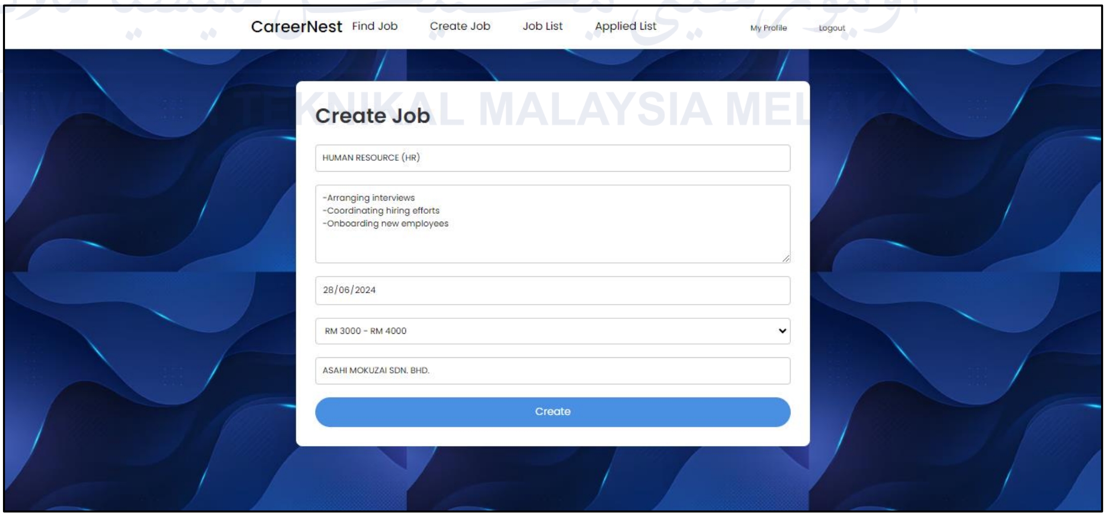
Name: Aiman
Username: aiman
Password: ***
Email:
Phone:
Address:

Update Profile

Name : Aiman
Username : aiman
Password : •••
Email :
Phone :
Address :

Figure 4.20: Update Profile Page

4.4.2.6 Employer Module



Create Job

HUMAN RESOURCE (HR)

-Arranging interviews
-Coordinating hiring efforts
-Onboarding new employees

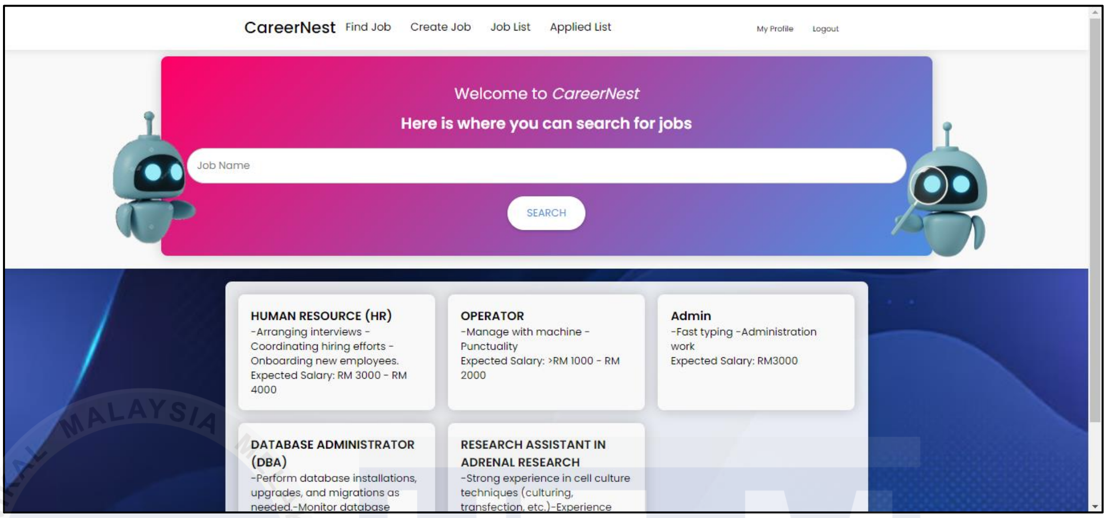
28/06/2024

RM 3000 - RM 4000

ASAHI MOKUZAI SDN. BHD.

Figure 4.21: Create Job Form Page

This specific page shown in Figure 4.21 is designed for employers to create new job listings. Employers can enter details about the open position, such as the job title and description. They can also specify the expected salary range and their company information. Once they have filled out this form, they can publish the job listing by clicking the "Create" button. Once it is published, the posting will appear at the find job page as the latest job that had been created shown in Figure 4.22.



HUMAN RESOURCE (HR)
-Arranging interviews -Coordinating hiring efforts -Onboarding new employees.
Expected Salary: RM 3000 - RM 4000

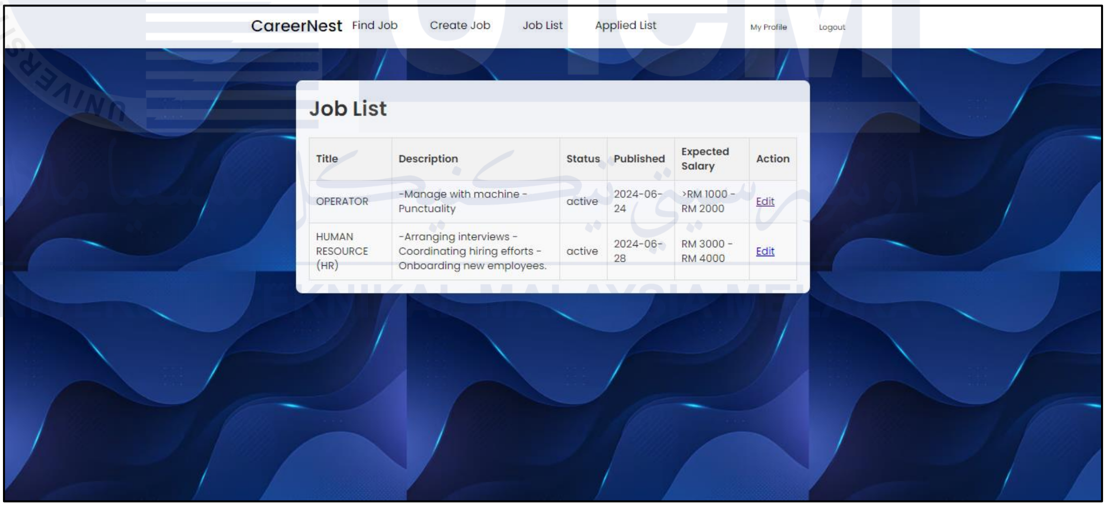
OPERATOR
-Manage with machine -Punctuality
Expected Salary: >RM 1000 - RM 2000

Admin
-Fast typing -Administration work
Expected Salary: RM3000

DATABASE ADMINISTRATOR (DBA)
-Perform database installations, upgrades, and migrations as needed.-Monitor database

RESEARCH ASSISTANT IN ADRENAL RESEARCH
-Strong experience in cell culture techniques (culturing, transfection, etc.)-Experience

Figure 4.22: Find Job Page

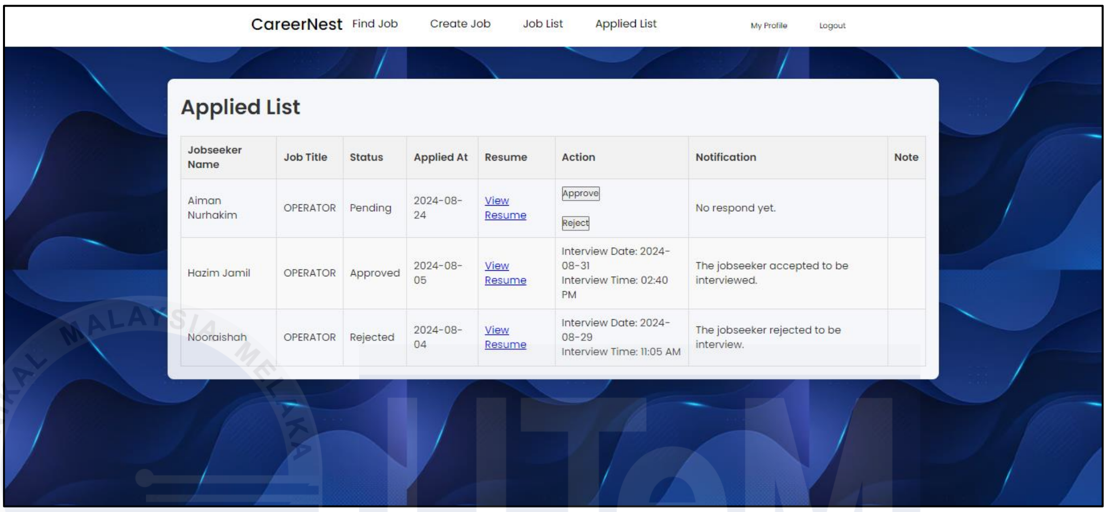


Job List

Title	Description	Status	Published	Expected Salary	Action
OPERATOR	-Manage with machine - Punctuality	active	2024-06-24	>RM 1000 - RM 2000	Edit
HUMAN RESOURCE (HR)	-Arranging interviews - Coordinating hiring efforts - Onboarding new employees.	active	2024-06-28	RM 3000 - RM 4000	Edit

Figure 4.23: Job List History Page

The job listings shown in Figure 4.23 refers to the history of job that employer had been created before which include details such as the job title, a brief description of the responsibilities, the status of the application (active in this case), the date it was published, the expected salary range and finally some action buttons. These buttons likely allow the user to edit the job listing or see a list of applicants. The status can be change into active or inactive where it will perform at the find job page. If the employer set the status of the job is active, means the job is still open for jobseeker to apply the positions. While if employer set the status as inactive, so the job will not be performed at the find job page meaning that the employer has closed the job positions.

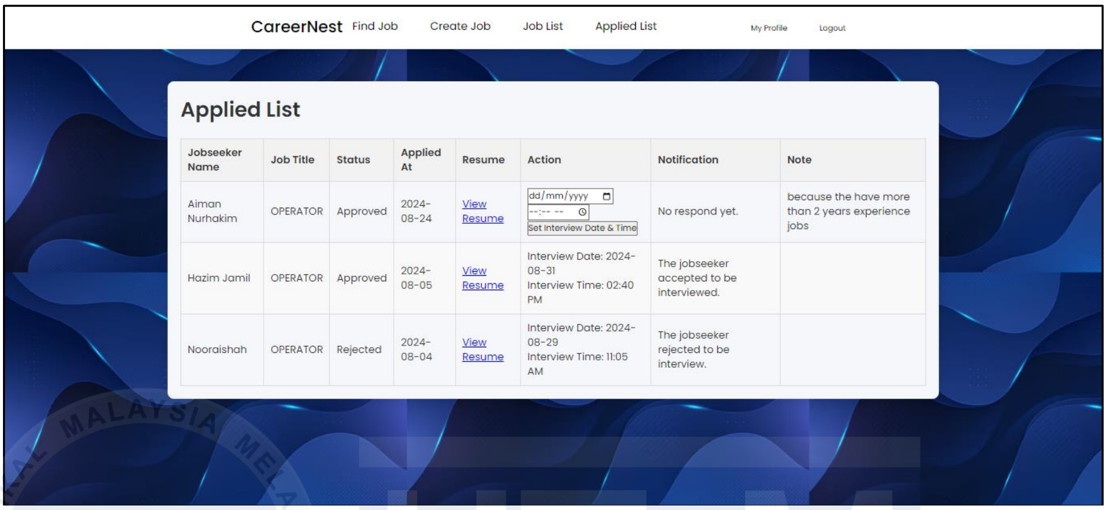


Applied List

Jobseeker Name	Job Title	Status	Applied At	Resume	Action	Notification	Note
Aiman Nurhakim	OPERATOR	Pending	2024-08-24	View Resume	Approve Reject	No respond yet.	
Hazim Jamil	OPERATOR	Approved	2024-08-05	View Resume	Interview Date: 2024-08-31 Interview Time: 02:40 PM	The jobseeker accepted to be interviewed.	
Nooraishah	OPERATOR	Rejected	2024-08-04	View Resume	Interview Date: 2024-08-29 Interview Time: 11:05 AM	The jobseeker rejected to be interview.	

Figure 4.24: Jobseeker Applied List Page

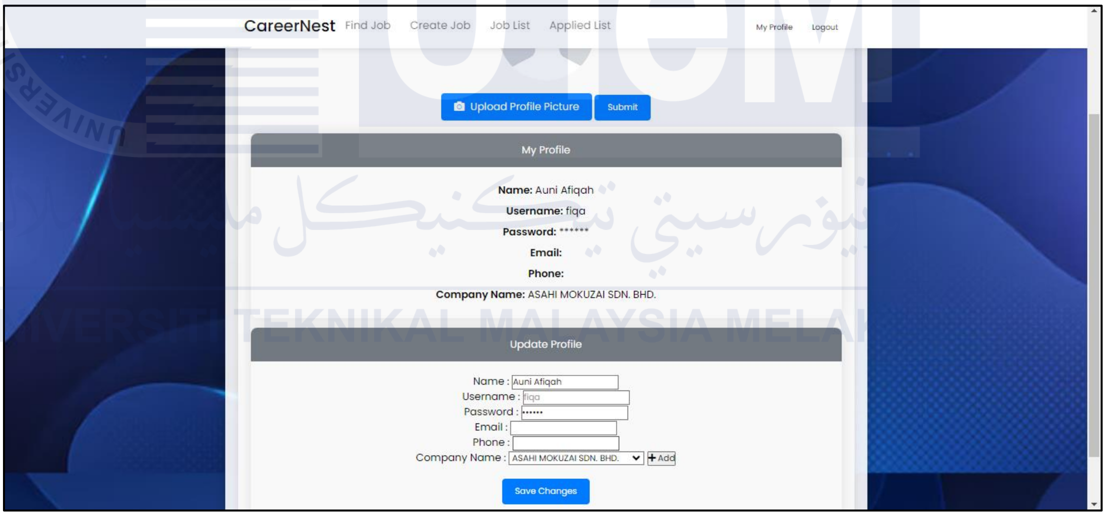
The Figure 4.24 illustrated the jobseeker applied list on CareerNest Management System. This page shows a list of applications that a jobseeker has submitted for various job openings. Each entry on the list provides details such as the jobseeker's name, the job title they applied for, the application status, the date it was submitted, the option to view the applicant's resume, the action where the employer needs to choose either want to approve or reject the application, notification and note. In this example, the applicant, Aiman, has applied for an operator position on August 24, 2024. The application status is currently listed as "Pending", which means it has been received but not yet reviewed by the employer. So, the employer is responsible to review the applied list and make an action to change the status of application either approve or reject. If employer done reviewed and decided to make the status change on each jobseeker applications, the employer must have a reason of approval and rejected as a note of history then the jobseeker will get notified in their page. Thus, if the employer decided to approve the application, the employer need to enter the date and time for interview session with the jobseekers illustrated in Figure 4.25 and it will appear in the interview details page as shown in Figure 4.18. Overall, this applied list page helps jobseekers to track the progress of their applications. They can easily see which positions they have applied for and the current status of each application.



Applied List

Jobseeker Name	Job Title	Status	Applied At	Resume	Action	Notification	Note
Aiman Nurhakim	OPERATOR	Approved	2024-08-24	View Resume	dd/mm/yyyy --:-- -- Set Interview Date & Time	No respond yet.	because the have more than 2 years experience jobs
Hazim Jamil	OPERATOR	Approved	2024-08-05	View Resume	Interview Date: 2024-08-31 Interview Time: 02:40 PM	The jobseeker accepted to be interviewed.	
Nooraishah	OPERATOR	Rejected	2024-08-04	View Resume	Interview Date: 2024-08-29 Interview Time: 11:05 AM	The jobseeker rejected to be interview.	

Figure 4.25: Jobseeker Applied List Page



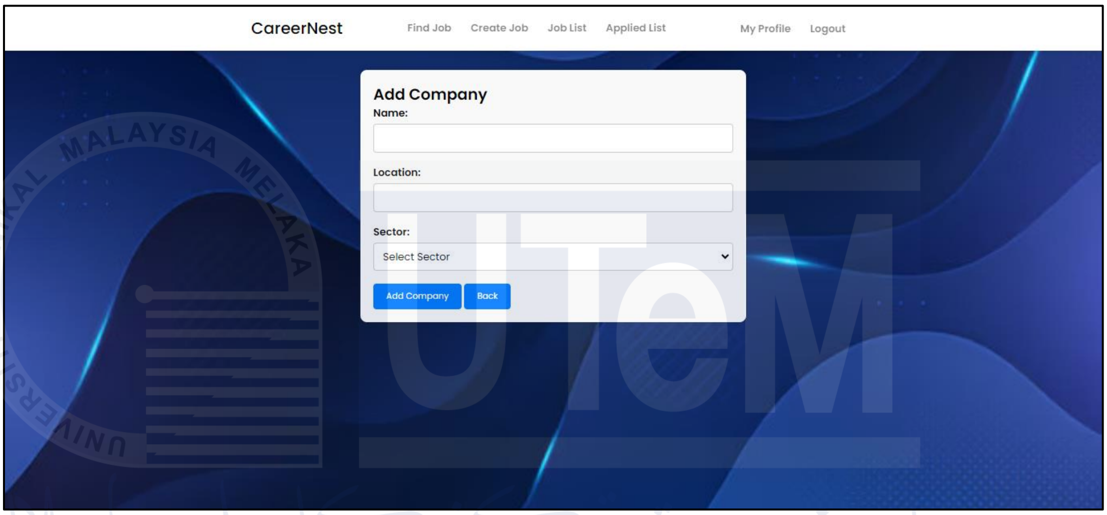
Name: Auni Afiqah
Username: fiqa
Password: ******
Email:
Phone:
Company Name: ASAHI MOKUZAI SDN. BHD.

Name : Auni Afiqah
Username : fiqa
Password : ••••••
Email :
Phone :
Company Name : ASAHI MOKUZAI SDN. BHD.

Figure 4.26: Update Profile Page

This page illustrated in Figure 4.26 allows employers to manage and update their personal information. The profile allows employer to edit details like name, password, email, phone, company name and even upload a profile picture to create a more professional and attractive online presence. The blank information such as email, and phone refer that information is a null that have been set in the database. In this example, the profile belongs to Auni Afiqah with the username "fiqa". They have also listed their current employer as ASAHI MOKUZAI SDN BHD. Thus, the name, username, password, and company name are required in the signup session.

The username cannot update since it is a fix information, but the remains information can edit. For the company name, they can choose the company they belong to in a drop-down menu. If their company does not exist yet in the company name list, they can add their company by clicking add button and the add company page will appear shows in Figure 4.27. Thus, if the username wants to be edit by the employer, they need to call admin to edit their username. Once an employer makes their edits, they can click the "Save Changes" button to update their profile information.



Add Company

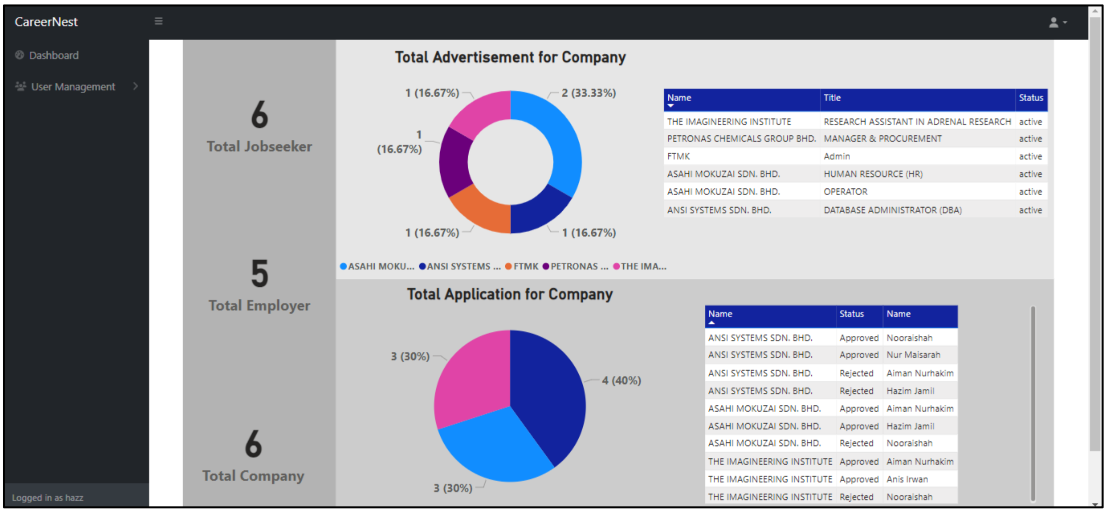
Name:

Location:

Sector: Select Sector

Figure 4.27: Add Company Page

4.4.2.7 Admin Module

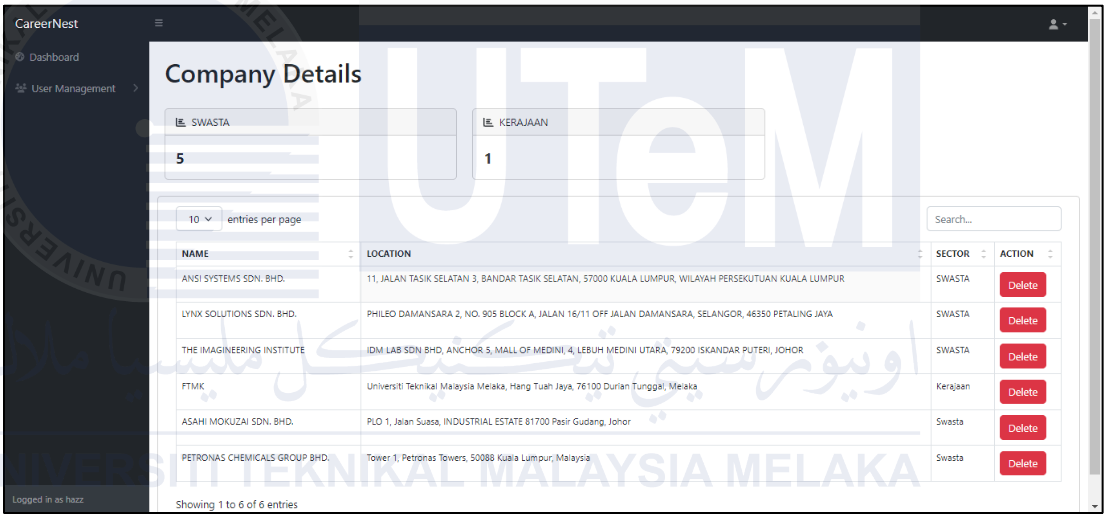


Name	Title	Status
THE IMAGINEERING INSTITUTE	RESEARCH ASSISTANT IN ADRENAL RESEARCH	active
PETRONAS CHEMICALS GROUP BHD.	MANAGER & PROCUREMENT	active
FTMK	Admin	active
ASAHI MOKUZAI SDN. BHD.	HUMAN RESOURCE (HR)	active
ASAHI MOKUZAI SDN. BHD.	OPERATOR	active
ANSI SYSTEMS SDN. BHD.	DATABASE ADMINISTRATOR (DBA)	active

Name	Status	Name
ANSI SYSTEMS SDN. BHD.	Approved	Nooraishah
ANSI SYSTEMS SDN. BHD.	Approved	Nur Maisarah
ANSI SYSTEMS SDN. BHD.	Rejected	Aiman Nurhakim
ANSI SYSTEMS SDN. BHD.	Rejected	Hazim Jamil
ASAHI MOKUZAI SDN. BHD.	Approved	Aiman Nurhakim
ASAHI MOKUZAI SDN. BHD.	Approved	Hazim Jamil
ASAHI MOKUZAI SDN. BHD.	Rejected	Nooraishah
THE IMAGINEERING INSTITUTE	Approved	Aiman Nurhakim
THE IMAGINEERING INSTITUTE	Approved	Anis Irwan
THE IMAGINEERING INSTITUTE	Rejected	Nooraishah

Figure 4.28: Admin Dashboard Page

The CareerNest Management System dashboard shown as Figure 4.28 is a vital tool for administrators to manage activity on the platform. This dashboard, designed using Power BI, provides a user-friendly visual interface to display key metrics. On the left side, administrators can see a quick snapshot of the user base with tiles showing the total number of job seekers, companies, and employers registered. The right section focuses on job postings. There is a pie chart labelled "Advertisement by Status" gives a quick overview of the proportion of active versus inactive postings. A detailed list of job titles and their corresponding statuses accompanies the chart, allowing administrators to monitor the availability of job postings on the platform.



Company Details

SWASTA 5

KERAJAAN 1

10 entries per page

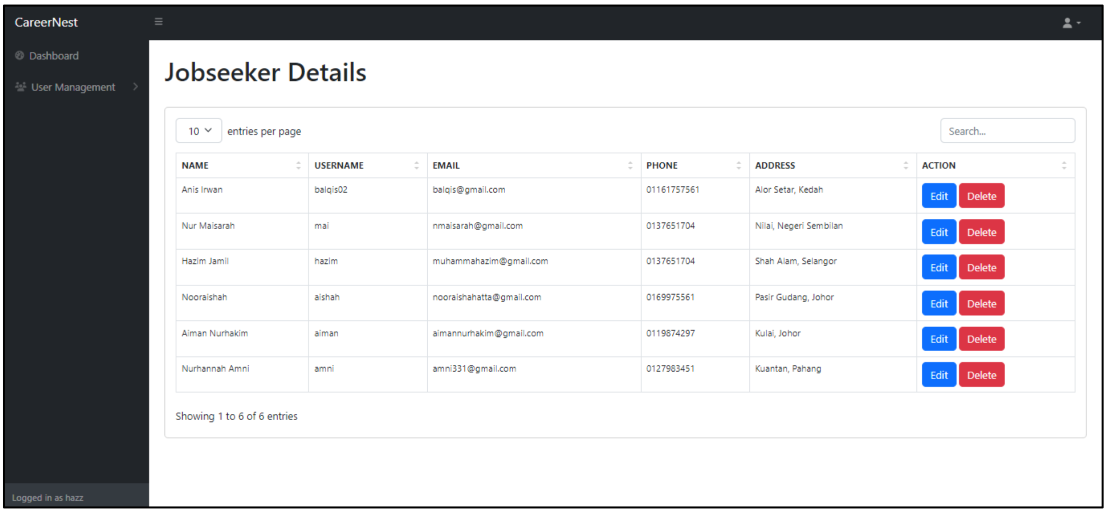
Search...

NAME	LOCATION	SECTOR	ACTION
ANSI SYSTEMS SDN. BHD.	11, JALAN TASIK SELATAN 3, BANDAR TASIK SELATAN, 57000 KUALA LUMPUR, WILAYAH PERSEKUTUAN KUALA LUMPUR	SWASTA	Delete
LYNX SOLUTIONS SDN. BHD.	PHILEO DAMANSARA 2, NO. 905 BLOCK A, JALAN 16/11 OFF JALAN DAMANSARA, SELANGOR, 46350 PETALING JAYA	SWASTA	Delete
THE IMAGINEERING INSTITUTE	IDM LAB SDN BHD, ANCHOR 5, MALL OF MEDINI, 4, LEBUH MEDINI UTARA, 79200 ISKANDAR PUTERI, JOHOR	SWASTA	Delete
FTMK	Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka	Kerajaan	Delete
ASAHI MOKUZAI SDN. BHD.	PLO 1, Jalan Suasa, INDUSTRIAL ESTATE 81700 Pasir Gudang, Johor	Swasta	Delete
PETRONAS CHEMICALS GROUP BHD.	Tower 1, Petronas Towers, 50088 Kuala Lumpur, Malaysia	Swasta	Delete

Showing 1 to 6 of 6 entries

Figure 4.29: Company Details Page

The company details page shows in Figure 4.29 appears to be a list view of company which displays key details about a company such as name, location, sector, and action. The delete button will give a permission for admin to delete the company name by using company id and all the information about that company will be delete permanently from database. The sector of the company available for “swasta” and “kerajaan” only. The “swasta” sector will be count from the list and shows the total of swasta sector in the top of the list. While the total of kerajaan will be count through the column sector of company details and it will appear as total count at the box above the detail list.



Jobseeker Details

10 entries per page

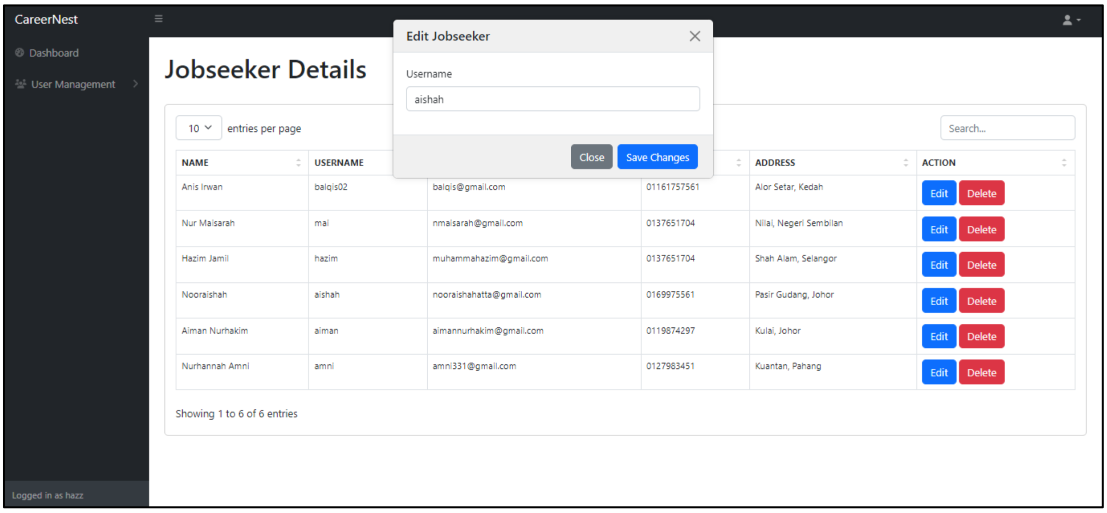
Search...

NAME	USERNAME	EMAIL	PHONE	ADDRESS	ACTION
Anis Irwan	balqis02	balqis@gmail.com	01161757561	Alor Setar, Kedah	Edit Delete
Nur Maisarah	mai	nmaisarah@gmail.com	0137651704	Nilai, Negeri Sembilan	Edit Delete
Hazim Jamil	hazim	muhammahazim@gmail.com	0137651704	Shah Alam, Selangor	Edit Delete
Nooraishah	aishah	nooraishahatta@gmail.com	0169975561	Pasir Gudang, Johor	Edit Delete
Aiman Nurhakim	aiman	aimannurhakim@gmail.com	0119874297	Kulai, Johor	Edit Delete
Nurhannah Amni	amni	amni331@gmail.com	0127983451	Kuantan, Pahang	Edit Delete

Showing 1 to 6 of 6 entries

Figure 4.30: Jobseeker Details Page

The jobseeker details page shows in Figure 4.30 appears to be a list view of jobseeker information which displays key details about a jobseeker such as name, username, email, phone, address, and action. The delete button will give a permission for admin to delete the jobseeker's name by using jobseeker id and all the information about that jobseeker will be delete permanently from database. Thus, by clicking on edit button, it will show in the Figure 4.31 which admin have responsible to edit the jobseeker's username the jobseeker has not any permission to edit their username unless they need to contact the admin to change their username.



Edit Jobseeker

Username: aishah

Jobseeker Details

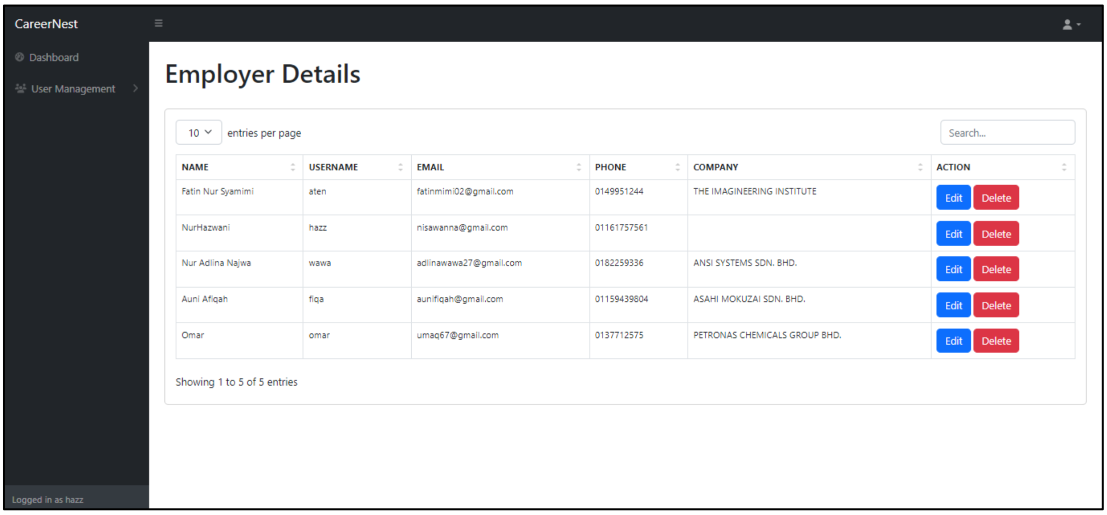
10 entries per page

Search...

NAME	USERNAME			ADDRESS	ACTION
Anis Irwan	balqis02	balqis@gmail.com	01161757561	Alor Setar, Kedah	Edit Delete
Nur Maisarah	mai	nmaisarah@gmail.com	0137651704	Nilai, Negeri Sembilan	Edit Delete
Hazim Jamil	hazim	muhammahazim@gmail.com	0137651704	Shah Alam, Selangor	Edit Delete
Nooraishah	aishah	nooraishahatta@gmail.com	0169975561	Pasir Gudang, Johor	Edit Delete
Aiman Nurhakim	aiman	aimannurhakim@gmail.com	0119874297	Kulai, Johor	Edit Delete
Nurhannah Amni	amni	amni331@gmail.com	0127983451	Kuantan, Pahang	Edit Delete

Showing 1 to 6 of 6 entries

Figure 4.31: Edit Jobseeker Page



Employer Details

10 entries per page

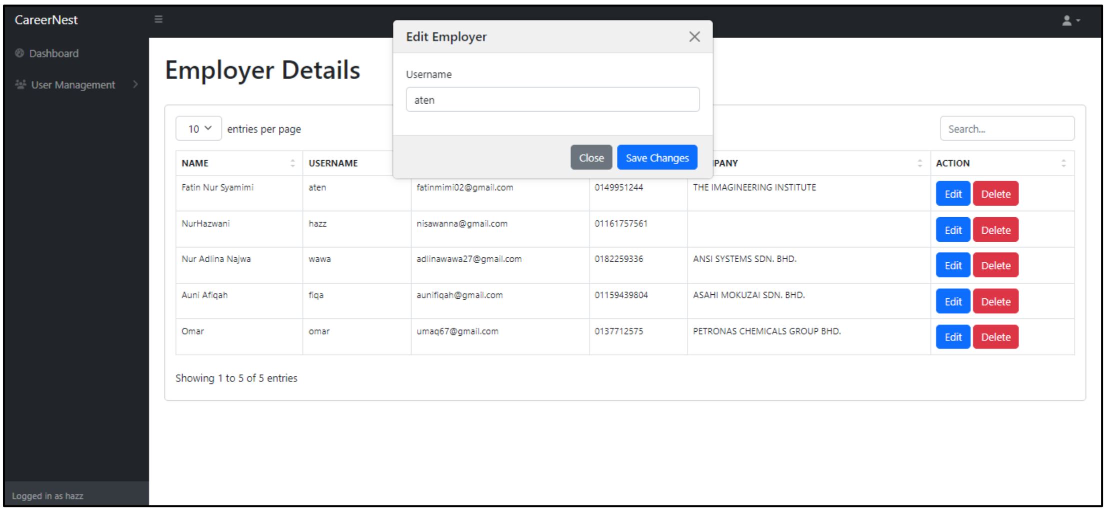
Search...

NAME	USERNAME	EMAIL	PHONE	COMPANY	ACTION
Fatin Nur Syamimi	aten	fatinmimi02@gmail.com	0149951244	THE IMAGINEERING INSTITUTE	Edit Delete
NurHazwani	hazz	nisawanna@gmail.com	01161757561		Edit Delete
Nur Adlina Najwa	wawa	adlinawawa27@gmail.com	0182259336	ANSI SYSTEMS SDN. BHD.	Edit Delete
Auni Afiqah	fiqa	aunifiqah@gmail.com	01159439804	ASAHI MOKUZAI SDN. BHD.	Edit Delete
Omar	omar	umaq67@gmail.com	0137712575	PETRONAS CHEMICALS GROUP BHD.	Edit Delete

Showing 1 to 5 of 5 entries

Figure 4.32: Employer Details Page

The employer details page shows in Figure 4.32 appears to be a list view of employer information which displays key details about an employer such as name, username, email, phone, address, and action. The delete button will give a permission for admin to delete the employer's name by using employer id and all the information about that employer will be delete permanently from database. Thus, by clicking on edit button, it will show in the Figure 4.33 which admin have responsible to edit the employer's username since an employer have not any permission to edit their username unless they need to contact the admin to change their username.



Employer Details

Edit Employer

Username: aten

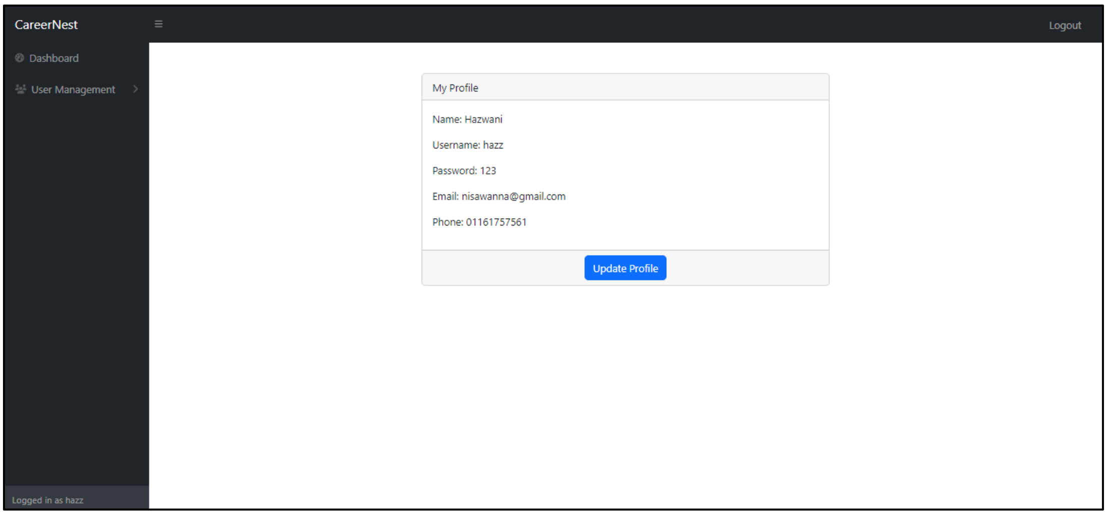
10 entries per page

Search...

NAME	USERNAME			PANY	ACTION
Fatin Nur Syamimi	aten	fatinmimi02@gmail.com	0149951244	THE IMAGINEERING INSTITUTE	Edit Delete
NurHazwani	hazz	nisawanna@gmail.com	01161757561		Edit Delete
Nur Adlina Najwa	wawa	adlinawawa27@gmail.com	0182259336	ANSI SYSTEMS SDN. BHD.	Edit Delete
Auni Afiqah	fiqa	aunifiqah@gmail.com	01159439804	ASAHI MOKUZAI SDN. BHD.	Edit Delete
Omar	omar	umaq67@gmail.com	0137712575	PETRONAS CHEMICALS GROUP BHD.	Edit Delete

Showing 1 to 5 of 5 entries

Figure 4.33: Edit Employer Page



My Profile

Name: Hazwani

Username: hazz

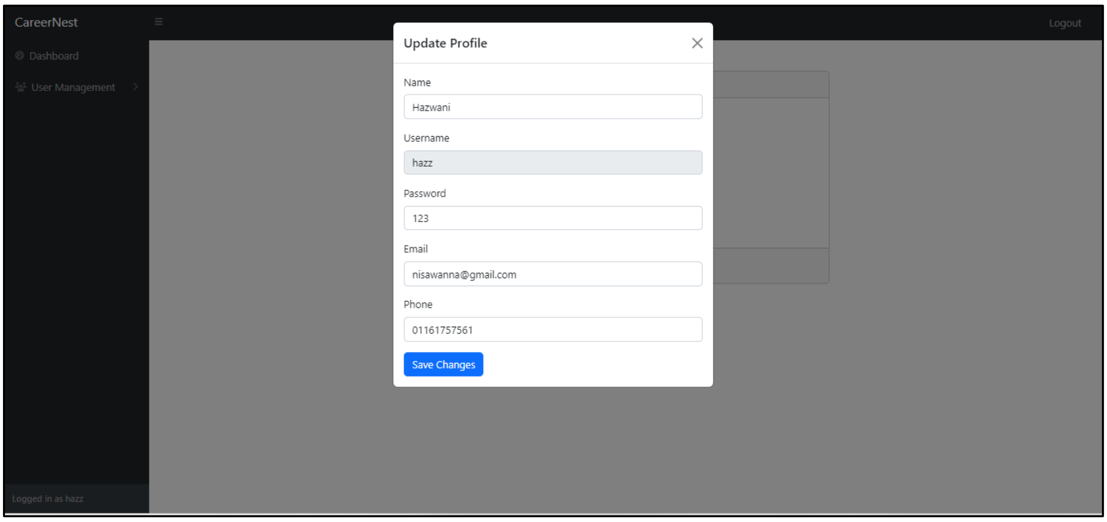
Password: 123

Email: nisawanna@gmail.com

Phone: 01161757561

Figure 4.34: Admin Profile Page

The admin profile page as illustrated in Figure 4.34 allows admin to manage and update their personal information. Once they click on button update profile, it will appear page update admin profile as shown in Figure 4.35 where the profile allows admin to edit details like name, password, email, and phone. The username cannot update since it is a fix information, but the remains information can be edit. Once admin makes their edits, they can click the "Save Changes" button to update their profile information.



Update Profile

Name: Hazwani

Username: hazz

Password: 123

Email: nisawanna@gmail.com

Phone: 01161757561

Figure 4.35: Update Profile Page

4.5 Conclusion

In Chapter 4, the transition from the analysis phase to the design phase was made, laying the groundwork for the CareerNest Management System's architecture. The chapter begins with a comprehensive exploration of the database design, covering the conceptual, logical, and physical aspects. This ensured that data storage and retrieval mechanisms were robust, efficient, and scalable. The conceptual design established the foundational data model and relationships, the logical design translated these into a structured schema, and the physical design focused on practical implementation details to optimize performance.

Following the database design, the focus shifted to GUI design. The system emphasized creating user-friendly interfaces that facilitate seamless interaction for job seekers and employers. Through iterative prototyping and usability testing, the system aimed to balance functionality with user experience, adhering to industry standards and best practices.

The design considerations outlined in this chapter are pivotal for the successful implementation of the CareerNest Management System. By thoughtfully designing both the backend infrastructure and frontend interfaces, the foundation is set for developing a cohesive, efficient, and user-centric recruitment platform.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

The implementation phase of the CareerNest Management System marks a critical step in bringing the project from concept to reality. This phase involves translating the system's design into a functional application that meets the needs of administrators, jobseekers, and employers. By leveraging modern development tools such as Laragon, PHP, HTML, phpMyAdmin, and Visual Studio Code, the system is built to deliver a seamless and efficient user experience. Throughout the implementation process, key functionalities are integrated, including job postings, application tracking, and profile management, ensuring that the system operates smoothly and effectively. The use of Microsoft Power BI for data visualization further enhances the system's capability to provide valuable insights and reports. The implementation is carefully executed to align with the project's objectives, ensuring that the CareerNest Management System is robust, scalable, and ready to support the recruitment and hiring process.

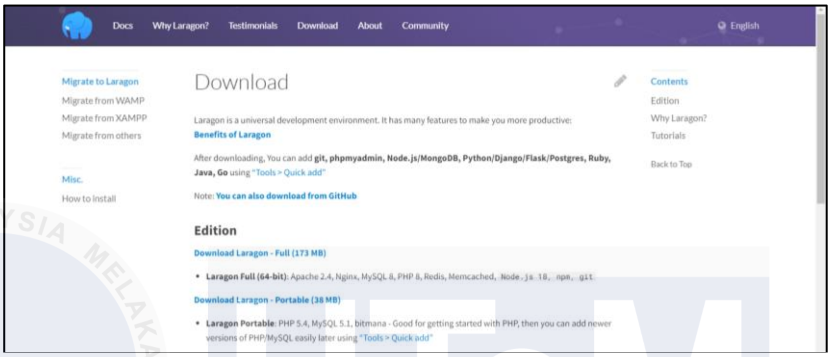
5.2 Software Development Environment Setup

Establishing the software development environment is a vital step in ensuring the smooth and efficient creation of the CareerNest Management System. This phase involves setting up the key tools and platforms necessary for designing, implementing, and managing the job portal application. Laragon is used for local server and database management, providing a strong backend foundation for the system. Visual Studio Code facilitates code development and debugging, offering a versatile environment that boosts coding efficiency. Additionally, Microsoft Power BI is integrated for data visualization and reporting, delivering valuable insights into system performance and user activity. Collectively, these tools form a robust environment that enables seamless development, testing, and deployment, laying a strong groundwork for the successful implementation of the CareerNest Management System.

5.2.1 System and Database Installation Setup

i. Laragon

- Step 1: Download the Laragon installer from the official website. Go to https://laragon.org/download/ and choose the full edition of Laragon.



Download

Laragon is a universal development environment. It has many features to make you more productive:
Benefits of Laragon

After downloading, You can add **git, phpmyadmin, Node.js/MongoDB, Python/Django/Flask/Postgres, Ruby, Java, Go** using "Tools > Quick add"

Note: **You can also download from GitHub**

Edition

Download Laragon - Full (173 MB)

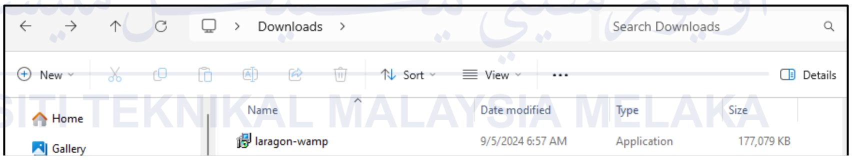
- **Laragon Full (64-bit)**: Apache 2.4, Nginx, MySQL 8, PHP 8, Redis, Memcached, `Node.js 18, npm, git`

Download Laragon - Portable (38 MB)

- **Laragon Portable**: PHP 5.4, MySQL 5.1, bitmana - Good for getting started with PHP, then you can add newer versions of PHP/MySQL easily later using "Tools > Quick add"

Figure 5.1 Download Page

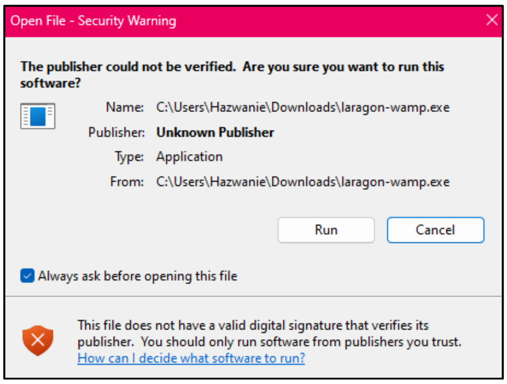
- Step 2: Locate the download destination folder in your laptop.



Name	Date modified	Type	Size
laragon-wamp	9/5/2024 6:57 AM	Application	177,079 KB

Figure 5.2: Location of Laragon application after download

- Step 3: Click the installation package until the window as shown in Figure 5.3 appears and click Run.



The publisher could not be verified. Are you sure you want to run this software?

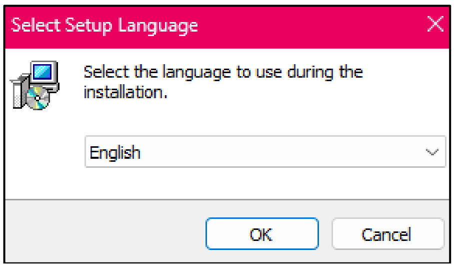
Name: C:\Users\Hazwanie\Downloads\laragon-wamp.exe
Publisher: **Unknown Publisher**
Type: Application
From: C:\Users\Hazwanie\Downloads\laragon-wamp.exe

☑ Always ask before opening this file

This file does not have a valid digital signature that verifies its publisher. You should only run software from publishers you trust.
How can I decide what software to run?

Figure 5.3: Setup Page

- Step 4: Choose preferred language and click OK.



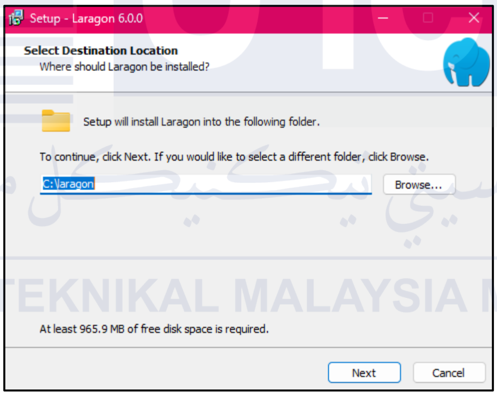
Select Setup Language

Select the language to use during the installation.

English

Figure 5.4: Setup Language Page

- Step 5: Choose the file destination folder where you want to install Laragon. You can choose the default location or select a different location by clicking on the "Browse" button. Then, click Next.



Setup - Laragon 6.0.0

Select Destination Location

Where should Laragon be installed?

Setup will install Laragon into the following folder.

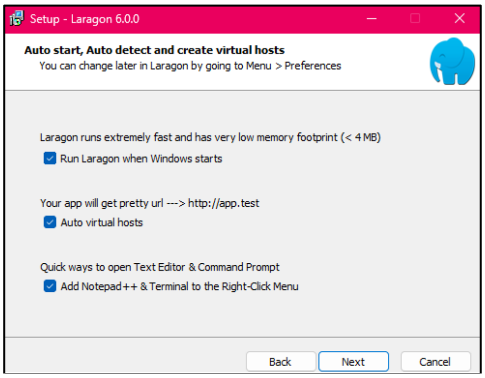
To continue, click Next. If you would like to select a different folder, click Browse.

C:\laragon

At least 965.9 MB of free disk space is required.

Figure 5.5: Installation Folder Page

- Step 6: Select all the checkboxes shown in Figure 5.6 and click Next.



Setup - Laragon 6.0.0

Auto start, Auto detect and create virtual hosts

You can change later in Laragon by going to Menu > Preferences

Laragon runs extremely fast and has very low memory footprint (< 4 MB)

☑ Run Laragon when Windows starts

Your app will get pretty url ---> http://app.test

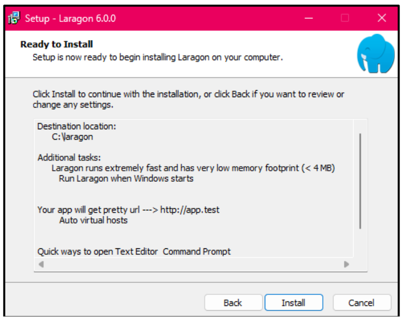
☑ Auto virtual hosts

Quick ways to open Text Editor & Command Prompt

☑ Add Notepad++ & Terminal to the Right-Click Menu

Figure 5.6: Select Component Page

- Step 7: Once you have completed all the steps above, the screen will show you all the previous selections and click on the "**Install**" button to begin the installation process. Laragon will now be installed on your computer.



Setup - Laragon 6.0.0

Ready to Install

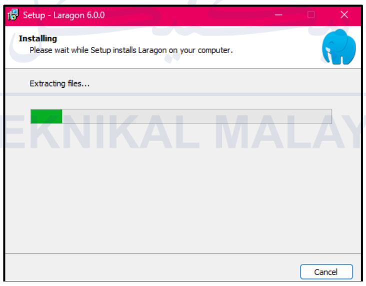
Setup is now ready to begin installing Laragon on your computer.

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

```
Destination location:
    C:\laragon

Additional tasks:
    Laragon runs extremely fast and has very low memory footprint (< 4 MB)
      Run Laragon when Windows starts


Your app will get pretty url ---> http://app.test
      Auto virtual hosts


Quick ways to open Text Editor  Command Prompt
```

Figure 5.7: Installation Page

- Step 8: The download progress bar shown in Figure 5.8 will appear indicating the installation process has begun. Once the installation is complete, click "Finish" button.



Setup - Laragon 6.0.0

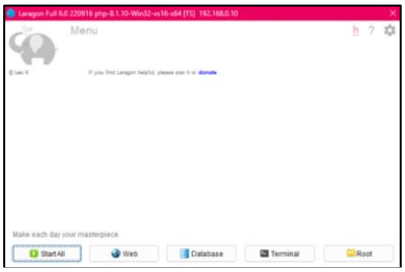
Installing

Please wait while Setup installs Laragon on your computer.

Extracting files...

Figure 5.8: Installation in Progress Page

- Step 9: Open Laragon application and click on the "Start All" button to start the services.



Laragon Full 6.0 220916 php-8.1.10-Win32-vs16-x64 [TS] 192.168.0.10

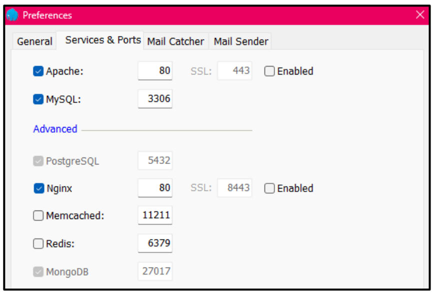
© Leo K

If you find Laragon helpful, please star it or donate

Make each day your masterpiece.

Figure 5.9: Control Panel Page

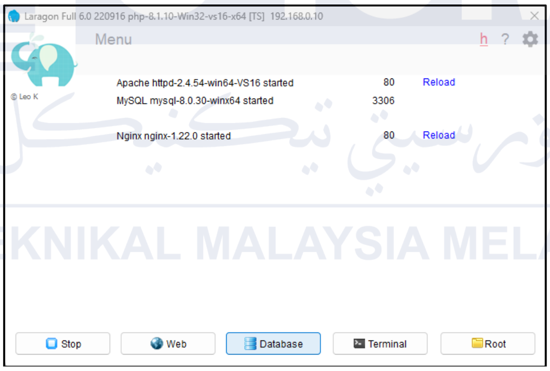
- Step 10: Click on setting icon at the top right of control panel page and choose Services & Ports tab to enable necessary options and ports like Apache, MySQL, and Nginx.



☑ Apache: 80 SSL: 443 □ Enabled
☑ MySQL: 3306
☑ PostgreSQL 5432
☑ Nginx 80 SSL: 8443 □ Enabled
□ Memcached: 11211
□ Redis: 6379
☑ MongoDB 27017

Figure 5.10: Services & Ports Page

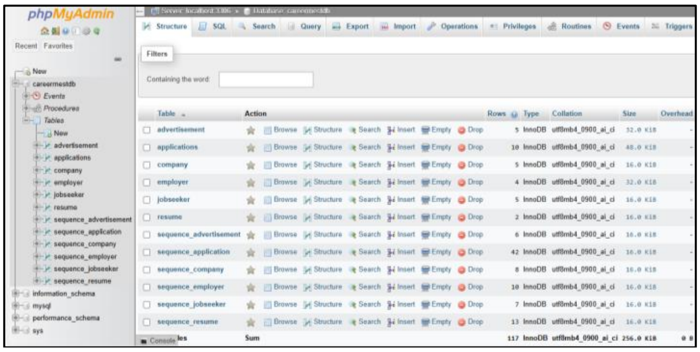
- Step 11: Click on the Database button at the bottom of the page to go direct to phpMyAdmin page.


Laragon Full 6.0 220916 php-8.1.10-Win32-vs16-x64 [TS] 192.168.0.10
Menu
h
?
© Leo K
Reload
Reload
Stop
Web
Database
Terminal
Root

```
Apache httpd-2.4.54-win64-VS16 started    80
MySQL mysql-8.0.30-winx64 started       3306

Nginx nginx-1.22.0 started                80
```

Figure 5.11: Database Page

- Step 12: The localhost will appear on browser as shown in Figure 5.12.



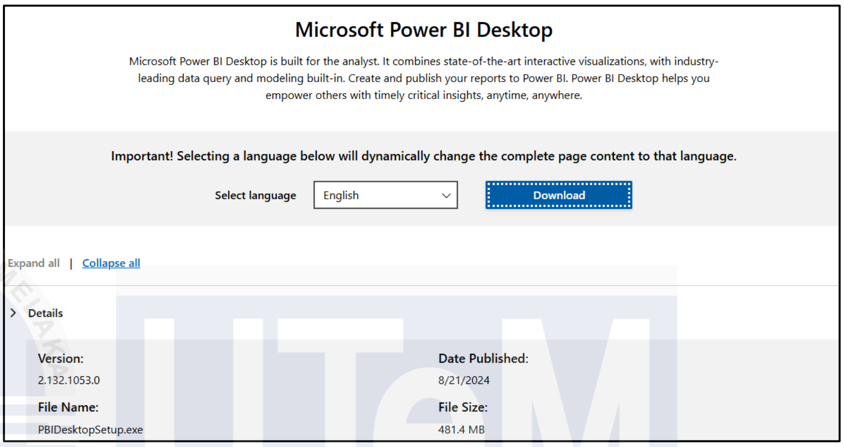
Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
advertisement	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_0900_ai_ci	32.0 KiB	-
applications	Browse Structure Search Insert Empty Drop	10	InnoDB	utf8mb4_0900_ai_ci	48.0 KiB	-
company	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
employer	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_0900_ai_ci	32.0 KiB	-
jobseeker	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
resume	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
sequence_advertisement	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
sequence_application	Browse Structure Search Insert Empty Drop	42	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
sequence_company	Browse Structure Search Insert Empty Drop	8	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
sequence_employer	Browse Structure Search Insert Empty Drop	10	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
sequence_jobseeker	Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
sequence_resume	Browse Structure Search Insert Empty Drop	13	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
les	Sum	117	InnoDB	utf8mb4_0900_ai_ci	256.0 KiB	0 B

Figure 5.12: phpMyAdmin Page

ii. Microsoft Power BI Desktop

- Step 1: Download the Power BI Desktop from the official website. Go to link below and click "Download" button.
  Link - https://www.microsoft.com/en-us/download/details.aspx?id=58494



Microsoft Power BI Desktop

Microsoft Power BI Desktop is built for the analyst. It combines state-of-the-art interactive visualizations, with industry-leading data query and modeling built-in. Create and publish your reports to Power BI. Power BI Desktop helps you empower others with timely critical insights, anytime, anywhere.

Important! Selecting a language below will dynamically change the complete page content to that language.

Select language: English

Version: 2.132.1053.0

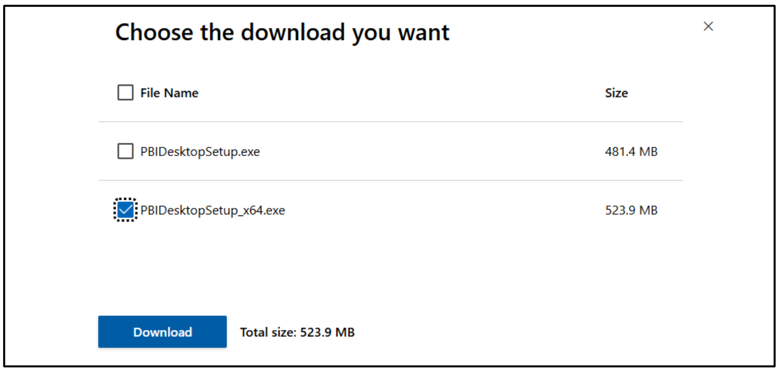
Date Published: 8/21/2024

File Name: PBIDesktopSetup.exe

File Size: 481.4 MB

Figure 5.13: Download Page

- Step 2: Choose the version of (64 bit) of Power BI Desktop and click "Download" button.



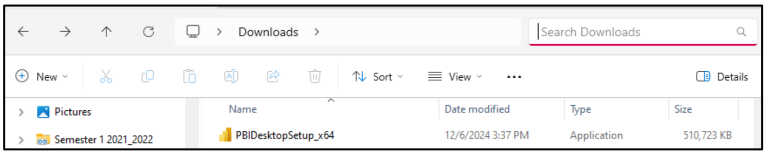
Choose the download you want

□ File Name	Size
□ PBIDesktopSetup.exe	481.4 MB
☑ PBIDesktopSetup_x64.exe	523.9 MB

Total size: 523.9 MB

Figure 5.14: File Page

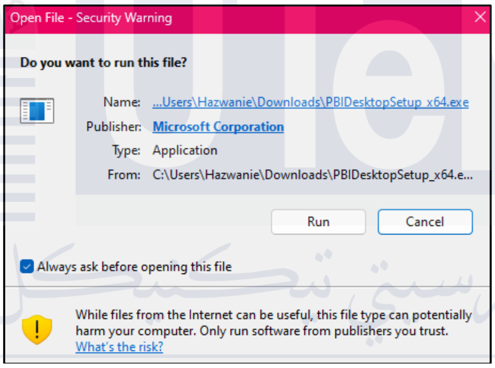
- Step 3: Locate the download destination folder in your laptop.



Name	Date modified	Type	Size
PBIDesktopSetup_x64	12/6/2024 3:37 PM	Application	510,723 KB

Figure 5.15: Location of Laragon application after download

- Step 4: Click the folder until the window as shown in Figure 5.16 appears and click Run.



Do you want to run this file?

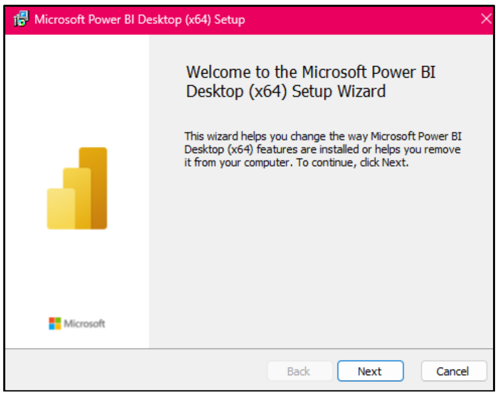
Name: ...Users\Hazwanie\Downloads\PBIDesktopSetup_x64.exe
Publisher: Microsoft Corporation
Type: Application
From: C:\Users\Hazwanie\Downloads\PBIDesktopSetup_x64.e...

☑ Always ask before opening this file

While files from the Internet can be useful, this file type can potentially harm your computer. Only run software from publishers you trust. What's the risk?

Figure 5.16: Setup Page

- Step 5: Click Next.

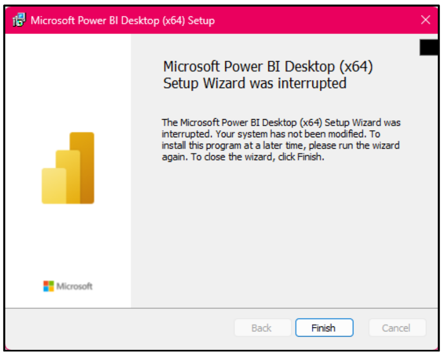


Welcome to the Microsoft Power BI Desktop (x64) Setup Wizard

This wizard helps you change the way Microsoft Power BI Desktop (x64) features are installed or helps you remove it from your computer. To continue, click Next.

Figure 5.17: Installation Page

- Step 6: Once the installation is completed, then click Finish.



Microsoft Power BI Desktop (x64) Setup Wizard was interrupted

The Microsoft Power BI Desktop (x64) Setup Wizard was interrupted. Your system has not been modified. To install this program at a later time, please run the wizard again. To close the wizard, click Finish.

Figure 5.18: Completing PBI Desktop Setup Page

- Step 7: Click blank report to start creating the dashboard.

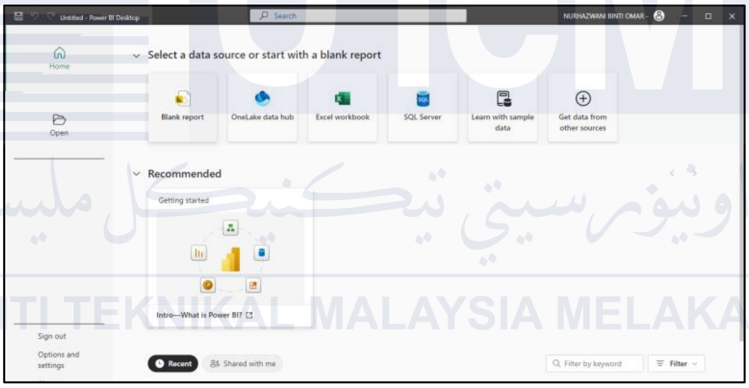
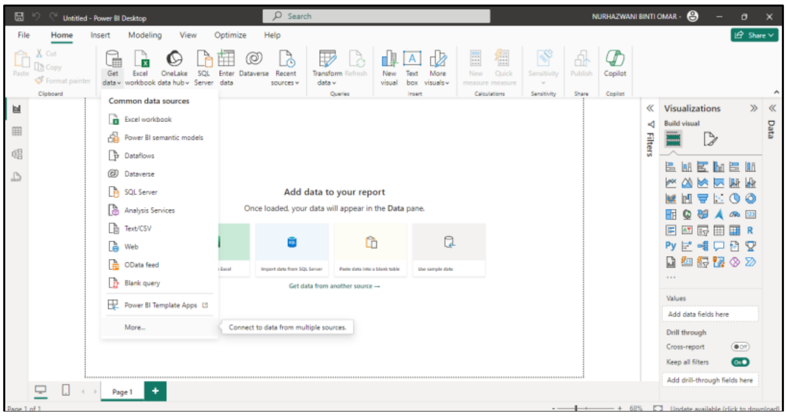


Figure 5.19: PBI Desktop Page

- Step 8: Choose “Get data” and click more.



Add data to your report

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

Figure 5.20: Get Data Page

- Step 9: Go to Database then click on MySQL database. Click “**Connect**” button.

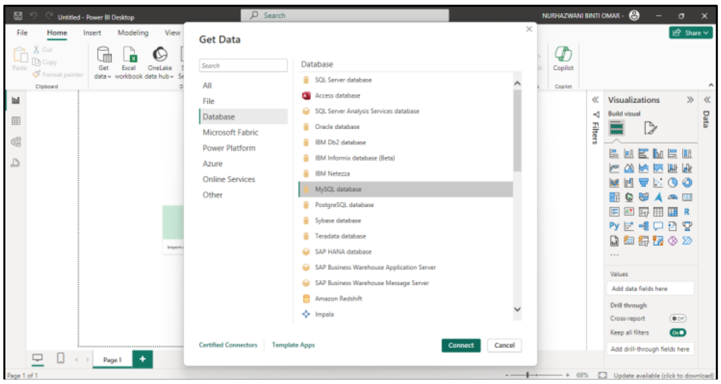
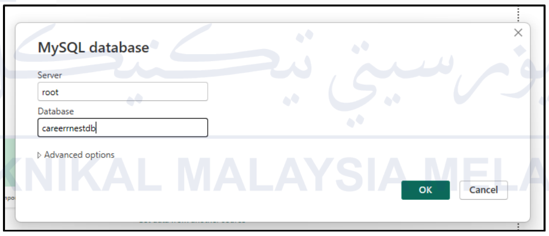


Figure 5.21: Import Data from MySQL database

- Step 10: Insert server name and database name that you want to import from MySQL database. Then, click OK.



MySQL database

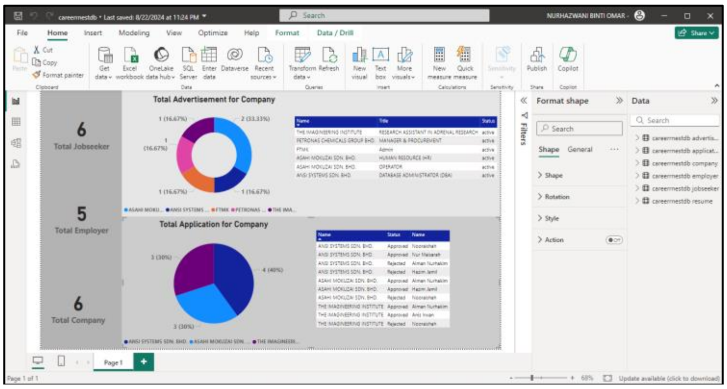
Server: root

Database: careerrnestdb

Figure 5.22: MySQL information

- Step 11: Select all the table from the database and click Load.

- Step 12: You can see all the data in database appear there.



careernestdb • Last saved: 8/22/2024 at 11:24 PM

Name	Title	Status
THE IMAGINEERING INSTITUTE	RESEARCH ASSISTANT IN ADRENAL RESEARCH	active
PETRONAS CHEMICALS GROUP BHD.	MANAGER & PROCUREMENT	active
FTMK	Admin	active
ASAHI MOKUZAI SDN. BHD.	HUMAN RESOURCE (HR)	active
ASAHI MOKUZAI SDN. BHD.	OPERATOR	active
ANSI SYSTEMS SDN. BHD.	DATABASE ADMINISTRATOR (DBA)	active

Name	Status	Name
ANSI SYSTEMS SDN. BHD.	Approved	Nooraishah
ANSI SYSTEMS SDN. BHD.	Approved	Nur Maisarah
ANSI SYSTEMS SDN. BHD.	Rejected	Aiman Nurhakim
ANSI SYSTEMS SDN. BHD.	Rejected	Hazim Jamil
ASAHI MOKUZAI SDN. BHD.	Approved	Aiman Nurhakim
ASAHI MOKUZAI SDN. BHD.	Approved	Hazim Jamil
ASAHI MOKUZAI SDN. BHD.	Rejected	Nooraishah
THE IMAGINEERING INSTITUTE	Approved	Aiman Nurhakim
THE IMAGINEERING INSTITUTE	Approved	Anis Irwan
THE IMAGINEERING INSTITUTE	Rejected	Nooraishah

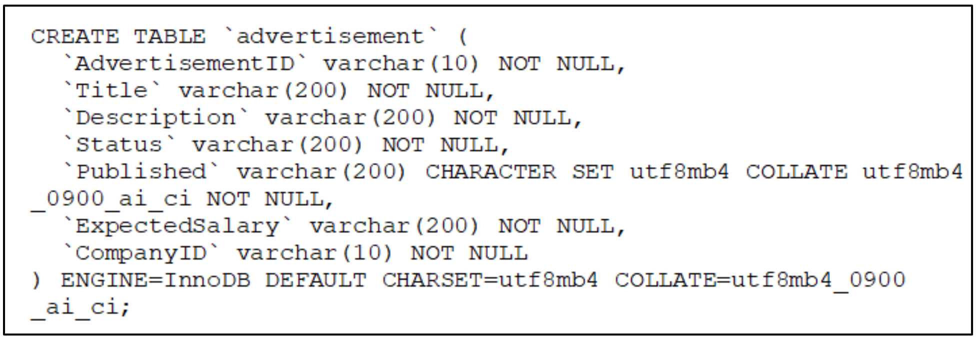
Figure 5.23: Data Load from MySQL

5.3 Database Implementation

The data implementation phase of the CareerNest Management System focuses on structuring, organizing, and populating the database to support the system's functionality. This phase is critical as it involves translating the system's design into a functional data structure that meets the needs of administrators, jobseekers, and employers. During this phase, tables are created for storing essential information, such as user profiles, job postings, applications, and notifications. The data is carefully organized to ensure efficiency, accuracy, and scalability, allowing for smooth data retrieval and manipulation throughout the system. Additionally, the implementation includes setting up relationships between different data entities, ensuring that the system can handle complex queries and operations with ease. By aligning the data implementation with the system's objectives, the CareerNest Management System is equipped to manage and process large volumes of information, providing a reliable and efficient platform for all users.

5.3.1 Data Definition Language (DDL)

1. Create ADVERTISEMENT table



```
CREATE TABLE `advertisement` (
  `AdvertisementID` varchar(10) NOT NULL,
  `Title` varchar(200) NOT NULL,
  `Description` varchar(200) NOT NULL,
  `Status` varchar(200) NOT NULL,
  `Published` varchar(200) CHARACTER SET utf8mb4 COLLATE utf8mb4
_0900_ai_ci NOT NULL,
  `ExpectedSalary` varchar(200) NOT NULL,
  `CompanyID` varchar(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900
_ai_ci;
```

Figure 5.24: DDL for ADVERTISEMENT

2. Create APPLICATIONS table

```
CREATE TABLE `applications` (
  `ApplicationID` varchar(10) NOT NULL,
  `applied_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `Status` varchar(200) CHARACTER SET utf8mb4 COLLATE utf8mb4_
0900_ai_ci DEFAULT 'pending',
  `resume_path` varchar(200) NOT NULL,
  `Notification` varchar(250) CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci DEFAULT NULL,
  `approved_at` datetime DEFAULT NULL,
  `IVDate` date DEFAULT NULL,
  `IVTime` time DEFAULT NULL,
  `Note` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900
_ai_ci DEFAULT NULL,
  `JobseekerID` varchar(10) NOT NULL,
  `AdvertisementID` varchar(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900
_ai_ci;
```

Figure 5.25: DDL for APPLICATIONS

3. Create COMPANY table

```
CREATE TABLE `company` (
  `CompanyID` varchar(10) NOT NULL,
  `Name` varchar(200) NOT NULL,
  `Location` varchar(200) NOT NULL,
  `Sector` varchar(200) CHARACTER SET utf8mb4 COLLATE utf8mb4_
0900_ai_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900
_ai_ci;
```

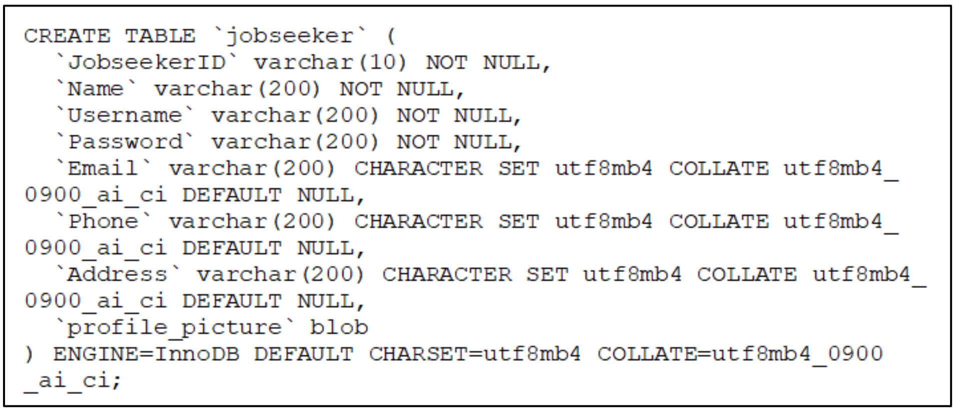
Figure 5.26: DDL for COMPANY

4. Create EMPLOYER table

```
CREATE TABLE `employer` (
  `EmployerID` varchar(10) NOT NULL,
  `Name` varchar(200) NOT NULL,
  `Role` varchar(200) NOT NULL,
  `Username` varchar(200) NOT NULL,
  `Password` varchar(200) NOT NULL,
  `Email` varchar(200) CHARACTER SET utf8mb4 COLLATE utf8mb4_
0900_ai_ci DEFAULT NULL,
  `Phone` varchar(200) CHARACTER SET utf8mb4 COLLATE utf8mb4_
0900_ai_ci DEFAULT NULL,
  `CompanyID` varchar(10) CHARACTER SET utf8mb4 COLLATE utf8mb4_
0900_ai_ci DEFAULT NULL,
  `profile_picture` blob
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900
_ai_ci;
```

Figure 5.27: DDL for EMPLOYER

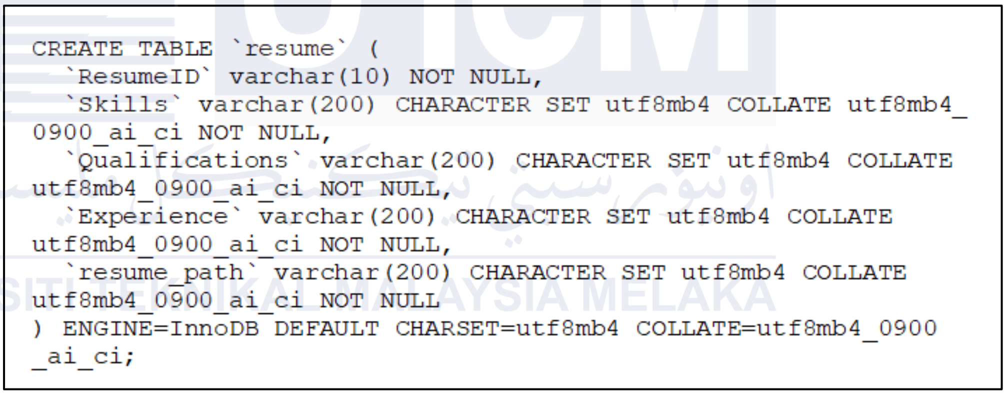
5. Create JOBSEEKER table



```
CREATE TABLE `jobseeker` (
  `JobseekerID` varchar(10) NOT NULL,
  `Name` varchar(200) NOT NULL,
  `Username` varchar(200) NOT NULL,
  `Password` varchar(200) NOT NULL,
  `Email` varchar(200) CHARACTER SET utf8mb4 COLLATE utf8mb4_
0900_ai_ci DEFAULT NULL,
  `Phone` varchar(200) CHARACTER SET utf8mb4 COLLATE utf8mb4_
0900_ai_ci DEFAULT NULL,
  `Address` varchar(200) CHARACTER SET utf8mb4 COLLATE utf8mb4_
0900_ai_ci DEFAULT NULL,
  `profile_picture` blob
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900
_ai_ci;
```

Figure 5.28: DDL for JOBSEEKER

6. Create RESUME table



```
CREATE TABLE `resume` (
  `ResumeID` varchar(10) NOT NULL,
  `Skills` varchar(200) CHARACTER SET utf8mb4 COLLATE utf8mb4_
0900_ai_ci NOT NULL,
  `Qualifications` varchar(200) CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci NOT NULL,
  `Experience` varchar(200) CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci NOT NULL,
  `resume_path` varchar(200) CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900
_ai_ci;
```

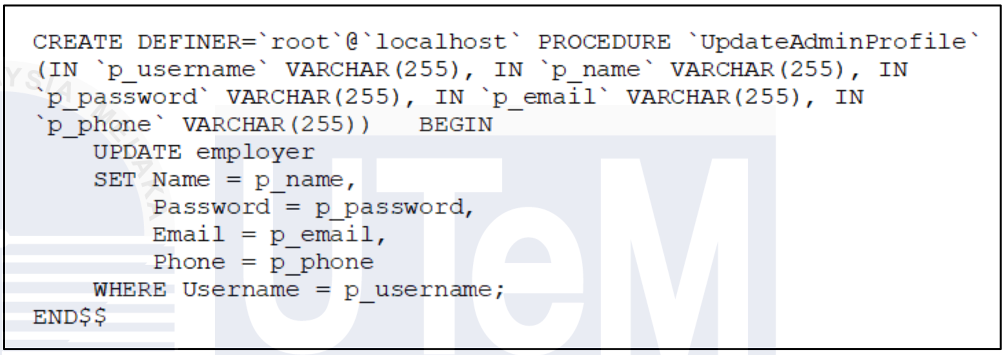
Figure 5.29: DDL for RESUME

5.3.2 Implementation of Main Processes

1. Stored Procedure

The UpdateAdminProfile stored procedure in the CareerNest Management System is designed to update the profile information of an employer in the database. The procedure takes in several parameters: p_username, p_name, p_password, p_email, and p_phone. These parameters represent the new values that will replace the existing information for an employer.

When the procedure is executed, it performs an UPDATE operation on the employer table, setting the Name, Password, Email, and Phone fields to the values provided in the procedure's parameters. The update is applied specifically to the record where the Username matches the provided p_username. This ensures that only the intended employer's profile is updated with the new details.



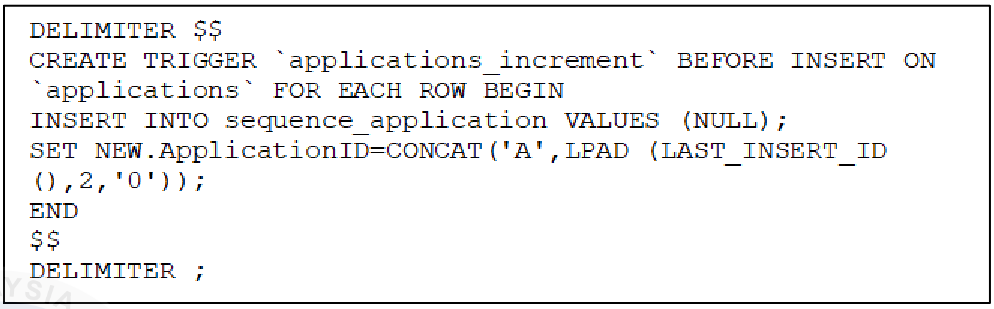
```
CREATE DEFINER=`root`@`localhost` PROCEDURE `UpdateAdminProfile`
(IN `p_username` VARCHAR(255), IN `p_name` VARCHAR(255), IN
`p_password` VARCHAR(255), IN `p_email` VARCHAR(255), IN
`p_phone` VARCHAR(255))   BEGIN
    UPDATE employer
    SET Name = p_name,
        Password = p_password,
        Email = p_email,
        Phone = p_phone
    WHERE Username = p_username;
END$$
```

Figure 5.30: SQL Procedure for UpdateAdminProfile

2. Trigger

The `applications_increment` trigger is a critical component in the CareerNest Management System, designed to ensure that every application record inserted into the `applications` table is assigned a unique and consistent `ApplicationID`. This process is essential for maintaining data integrity, facilitating easy data retrieval, and ensuring that each application is distinctly identifiable within the system.

When a new application is inserted, the trigger first interacts with the `sequence_application` table by inserting a null value. This action generates a new sequence number, which serves as the basis for creating the unique `ApplicationID`. The trigger then constructs this ID by concatenating the letter 'A' with the newly generated sequence number. To maintain a uniform format, the sequence number is padded with leading zeros to ensure it is at least two digits long. For example, if the sequence number generated is 1, the resulting `ApplicationID` would be 'A01'. This consistent and automated generation of `ApplicationID` simplifies database management, making it easier to track and manage application records within the system.

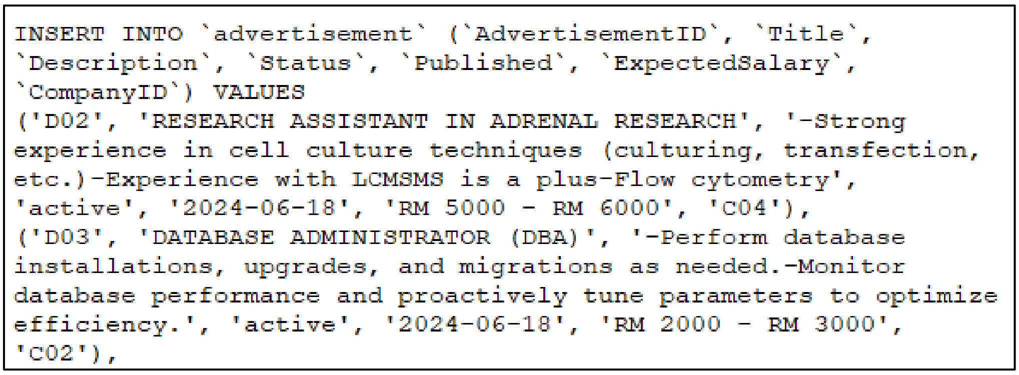


```
DELIMITER $$
CREATE TRIGGER `applications_increment` BEFORE INSERT ON
`applications` FOR EACH ROW BEGIN
INSERT INTO sequence_application VALUES (NULL);
SET NEW.ApplicationID=CONCAT('A',LPAD (LAST_INSERT_ID
(),2,'0'));
END
$$
DELIMITER ;
```

Figure 5.31: SQL Trigger for ApplicationID

5.3.3 Data Loading Process

The data loading process for the CareerNest Management System's database involves populating it with initial test data to verify the system's functionality and performance. This process is carried out using SQL INSERT statements, which insert predefined values into the database tables to create sample records. For instance, the advertisement table was populated with test data using the following SQL script:



```
INSERT INTO `advertisement` (`AdvertisementID`, `Title`,
`Description`, `Status`, `Published`, `ExpectedSalary`,
`CompanyID`) VALUES
('D02', 'RESEARCH ASSISTANT IN ADRENAL RESEARCH', '-Strong
experience in cell culture techniques (culturing, transfection,
etc.)-Experience with LCMSMS is a plus-Flow cytometry',
'active', '2024-06-18', 'RM 5000 - RM 6000', 'C04'),
('D03', 'DATABASE ADMINISTRATOR (DBA)', '-Perform database
installations, upgrades, and migrations as needed.-Monitor
database performance and proactively tune parameters to optimize
efficiency.', 'active', '2024-06-18', 'RM 2000 - RM 3000',
'C02'),
```

Figure 5.32: SQL script INSERT statement table 'advertisement'

In this context, the script provided is used to insert new records into the advertisement table of the CareerNest Management System's database. Each INSERT statement adds a unique job advertisement, identified by the

AdvertisementID, along with other key details like Title, Description, Status, Published date, ExpectedSalary, and associated CompanyID. This initial test data is crucial for verifying that the database schema is correctly structured and that the system can interact with the database as intended. It also allows for thorough testing of database queries, application logic, and user interface components before the system is deployed with real data, ensuring that everything operates smoothly and efficiently.

5.4 Conclusion

The implementation phase of the CareerNest Management System is critical in turning design concepts into a fully functional job portal application, involving key activities such as coding, configuration, and deployment. By setting up a strong software development environment using tools like Laragon, Visual Studio Code, Microsoft Power BI, and phpMyAdmin, the project lays a solid groundwork for development and testing. The database implementation, which includes the use of DDL (Data Definition Language) and Data Control Language statements, as well as the creation of stored procedures and triggers, plays a vital role in ensuring data integrity and efficient system operations. The process of data loading, carried out through SQL INSERT statements, populates the database with initial test data, validating the system's functionality and performance. This comprehensive implementation approach guarantees that the CareerNest Management System meets its requirements and is ready for effective real-world deployment.

CHAPTER 6: TESTING

6.1 Introduction

The testing phase is a crucial step in ensuring that the CareerNest Management System works as expected. During this phase, the system is thoroughly checked to make sure all features, performance, and security measures are functioning correctly. The goal is to identify and fix any issues or bugs before the system is fully deployed. This phase involves different types of tests, including unit testing, integration testing, system testing, and user acceptance testing (UAT). Each test focuses on specific parts of the system to ensure everything works together smoothly and provides a good user experience. By carefully testing the system, it confirms its reliability and prepare it for real-world use. This helps ensure that the CareerNest Management System is ready to meet the needs of jobseekers, employers, and administrators effectively.

6.2 Test Plan

A test plan is a detailed document that outlines the strategy, scope, resources, and schedule for testing a software application. It defines the objectives of testing, the items to be tested, the testing tasks to be performed, who will do the testing, the environment in which testing will take place, and the criteria for passing or failing each test. The purpose of a test plan is to ensure that all aspects of the software are tested thoroughly, and any defects are identified and addressed before the software is released.

6.2.1 Test Organization

In the test organization, different roles are assigned to manage and execute the testing process as shown in Table 6.1. The Test Manager oversees the process, while Test Engineers and QA Analysts handle the execution of test cases and reporting of defects. Developers focus on fixing issues, and stakeholders may be involved in reviewing the system during the final testing phase.

Table 6.1: Test Organization

Role	Responsibilities
Test Manager	Oversees the testing process, assigns tasks, and ensures timelines are met.
Test Engineers/QA Analysts	Execute test cases, log defects, and retest after fixes.
Developers	Fix defects identified during testing and ensure code quality.
Stakeholders	Review the system during the final testing phase and provide feedback.

6.2.2 Test Environment

The test environment is set up to replicate the production environment as closely as possible. It includes the necessary hardware, software, and tools required for testing. The environment is designed to ensure that the system performs well under various conditions and scenarios.

Table 6.2: Test Environment

Component	Specification	Description
Hardware	Operating System	Windows 11 Home Single Language
	Processor	Intel(R) Core(TM) i3-8130U CPU @ 2.20GHz
	RAM	4.00 GB (3.3 GB usable)
	System Type	64-bit operating system, x64-based processor
Software	Database	MySQL or relevant database used
	Web Server	Apache or relevant server for PHP
	Chrome, MicrosoftEdge, and other supported browsers	Chrome, MicrosoftEdge, and other supported browsers

Remarks	Test Data	Use realistic data mimicking production
	Test Environment Setup	Ensure environment closely matches production setup for accurate testing

6.2.3 Test Schedule

The test schedule outlines the timeline for the testing process, starting with the preparation phase, followed by the execution phase, which includes different levels of testing, and ending with the reporting phase. Each phase has specific tasks that must be completed within set timeframes to ensure the system is thoroughly tested.

Table 6.3: Test Schedule

Modules	Types	Duration
Registration	✓ Unit Testing ✓ Integration Testing ✓ User Acceptance Testing	5 days
Login	✓ Integration Testing ✓ User Acceptance Testing	3 days
Job Info	✓ Integration Testing ✓ User Acceptance Testing ✓ System Testing	5 days
Application Info	✓ Integration Testing ✓ User Acceptance Testing ✓ System Testing	12 days
Notification	✓ Unit Testing ✓ Integration Testing	7 days

6.3 Test Strategy

A test strategy is a plan that outlines how software testing will be conducted. It includes the goals of testing, what will be tested, and how testing will be done. It also specifies the resources needed, the schedule for testing, and the criteria for determining if the software is acceptable. Essentially, it provides a clear direction for testing to ensure the software meets its requirements and works properly before it is released. In this case, white box approach and black box approach has been selected for testing CareerNest Management System.

White box testing is a technique where testers have access to the internal code and logic of the software. This approach involves examining the code, algorithms, and internal structures to ensure that each part of the system functions correctly. For the CareerNest Management System, white box testing is crucial because it allows to validate the internal logic of features such as registration, login, and job applications. It helps identify coding errors that may not be visible through external testing and improves the overall quality of the code. Additionally, it ensures that all parts of the code are thoroughly tested, which helps in uncovering issues that could lead to system failures or inefficiencies. By using white box testing, it can enhance the reliability and performance of each application, making sure that it operates smoothly and meets all its functional requirements.

Black box testing focuses on evaluating a software application’s functionality without needing to understand its internal code or structure. For the CareerNest Management System, this means testing features like job applications, user registration, and admin functions from the user's perspective. It ensures that the system performs as expected and meets user requirements. Black box testing helps identify issues that might affect the user experience, such as incorrect outputs or usability problems. By focusing on how the system behaves in real-world scenarios, black box testing ensures that the software functions correctly and meets user needs.

6.3.1 Classes of tests

Classes of tests are categories of testing that focus on different aspects of the software's functionality, performance, and quality. They help in systematically evaluating the software to ensure it meets various requirements and performs as expected. Table 6.4 describes in detail about classes of tests.

Table 6.4: Description of Classes of tests

Class of Test	Explanation
Unit Testing	Tests individual components or functions of the system to ensure they work correctly on their own. For example, verifying that the registration process correctly handles user input.
Integration Testing	Examines the interaction between different components or modules to ensure they work together properly. For instance, checking that the job application module correctly interacts with the user profile module.
System Testing	Evaluates the complete, integrated system to ensure it meets the specified requirements and functions as expected. This includes testing the entire CareerNest system to confirm all features work together seamlessly.
User Acceptance Testing	Focuses on validating that the CareerNest Management System functions correctly in real-world scenarios and aligns with the user requirements. For example, users would interact with the system to confirm that features like job application management, automated notifications, and communication tools work as intended and provide a satisfactory user experience.

6.4 Test Design

Test design is the process of planning and creating detailed test cases and scenarios to evaluate a software application's functionality, performance, and overall quality. It involves defining test objectives, specifying inputs, and expected outcomes, and determining the conditions and environments for testing. The purpose of test design is to ensure comprehensive coverage of the software's requirements and to identify potential issues early in the development process. By developing structured and systematic tests, test design helps ensure that the software meets quality standards, functions correctly, and fulfills user needs effectively.

6.4.1 Test Description

The Test Description for the registration, login, and job modules involves validating that each module operates correctly and integrates smoothly. This includes verifying user account creation, authentication processes, and job advertisement functionality to ensure a seamless and effective user experience across the CareerNest Management System.

6.4.1.1 Registration Module

The Registration Module of the CareerNest Management System enables new users to create accounts by providing essential details such as their name, role, username, and password. It validates input to ensure that usernames are unique and securely stores user information in the database. The module includes options for selecting the user type, such as jobseeker or employer. To ensure a smooth and secure experience, comprehensive testing is performed to verify functionality, security, and usability.

Table 6.5: Description of Registration Module

Test Case ID	Description	Action	Expected Output

CN A/01	Name = blank
	Role = blank
		No input provided	ERROR

	Username = blank
	Password = blank		

CN A/02	Name = Aiman Nurhakim
	Role = blank
	Username = blank
	Password = blank
		Role, Username and Password left blank	ERROR

CN A/03	Name = blank
	Role = ‘jobseeker’
	Username = blank
	Password = blank
		Name, Username and Password left blank	ERROR

CN A/04	Name = blank
	Role = blank
	Username = aiman
	Password = blank
		Name, Role and Password left blank	ERROR

CN A/05	Name = blank
	Role = blank
	Username = blank
	Password = ***
		Name, Role and Username left blank	ERROR

CN A/06	Name = Aiman Nurhakim
	Role = ‘jobseeker’
	Username = aiman
	Password = ***
		All necessary input is inserted	OK

6.4.1.2 Login Module

The Login Module of the CareerNest Management System allows users to access their accounts by entering their username, password, and role (jobseeker, employer, or admin). It validates the provided credentials against the stored data to authenticate users. The module ensures that usernames and passwords match the records in the database. After successful login, users are directed to the appropriate dashboard or interface based on their role. Comprehensive testing is performed to verify the accuracy and security of the login process, ensuring a reliable access control mechanism for all users.

Table 6.6: Description of Login Module

Test Case ID	**Description**	**Action**	**Expected Output**

CN B/01	Username = blank
	Password = blank
	Role = blank
		No input provided	ERROR

CN B/02	Username = aiman
	Password = blank
	Role = blank
		Password and Role left blank	ERROR

CN B/03	Username = blank
	Password = ***
	Role = blank
		Username and Role left blank	ERROR

CN B/04	Username = blank
	Password = blank
	Role = ‘jobseeker’
		Username and Password left blank	ERROR

CN B/05	Username = aiman
	Password = ***
	Role = ‘jobseeker’
		All necessary input is inserted	OK

6.4.1.3 Job Module

The Job Module of the CareerNest Management System enables employers to add job advertisements. Users can create a new job posting by providing essential details such as the job title, description, salary, and company name. The module allows employers to input these details into the system, which are then securely stored in the database and made available for jobseekers to view. The module includes validation to ensure that all required fields are completed. Comprehensive testing ensures that the job postings are accurately created, stored, and displayed, providing a functional and user-friendly interface for employers to manage job advertisements.

Table 6.7: Description of Job Module

Test Case ID	Description	Action	Expected Output

CN C/01	JobTitle = blank
	Description = blank
	Salary = blank
	CompanyName = blank
		No input provided	ERROR

CN C/02	JobTitle = HR
	Description = blank
	Salary = blank
	CompanyName = blank
		Description, Salary and CompanyName left blank	ERROR

CN C/03	JobTitle = blank
	Description = Manage financial activities
	Salary = blank
	CompanyName = blank
		JobTitle, Salary and CompanyName left blank	ERROR

CN C/04	JobTitle = blank
	Description = blank
	Salary = RM3000-RM5000
	CompanyName = blank
		JobTitle, Description and CompanyName left blank	ERROR

CN C/05	JobTitle = blank
	Description = blank
	Salary = blank
	CompanyName = MMHE SDN. BHD.
		JobTitle, Description and Salary left blank	ERROR

CN C/06	JobTitle = HR
	Description = Manage financial activities
	Salary = RM3000-RM5000
	CompanyName = MMHE SDN. BHD.
		All necessary input is inserted	OK

6.4.2 Test Data

Test data is the information used during testing to check if a software system works correctly. It includes different types of data, like correct and incorrect values, to see how the software handles various situations. Test data helps find problems and ensures the software meets its requirements. Table 6.8 shows the examples of test data.

Table 6.8: Description of Login Test Data

Test No	Attribute	Data
TEST/01	Admin	
	Username	hazz
	Password	***
TEST/02	Jobseeker	
	Username	aiman
	Password	***
TEST/03	Employer	
	Username	wawa
	Password	******

6.5 Test Results and Analysis

Test results are the outcomes from running test cases on a software application, showing whether the software functions as expected or if there are any issues. Test analysis involves reviewing these results to understand the performance and quality of the software, including identifying the causes of any failures and assessing their impact. This process helps determine if the software meets its requirements and guides decisions on whether to fix issues, retest, or proceed with deployment.

System: CareerNest Management System

Module: Registration Module

Table 6.9: Test Result and Analysis for Registration Module

Test Number	**Action**	**Result**	**Pass Initials (OK / Fail)**
CN_1000	Valid input: Based on each input type. Condition: User enters personal details.	System will prompt if the username has already existed or not. If exists, user need to change to unique username.	OK

System: CareerNest Management System

Module: Login Module

Table 6.10: Test Result and Analysis for Login Module

Test Number	**Action**	**Result**	**Pass Initials (OK / Fail)**
CN_1001	Valid input: Username: aiman Password: *** Condition: Username and Password already in the database.	Able to access the system as jobseeker.	OK
CN_1002	Valid input: Username: testing Password: ******* Condition: Username and Password does not exist in the database.	Display error message	OK

System: CareerNest Management System

Module: Job Module

Table 6.11: Test Result and Analysis for Job Module

Test Number	Action	Result	Pass Initials (OK / Fail)
CN_1003	Valid input: Status of advertisement is active. Condition: User edit the status of advertisement.	System will appear advertisement which have active status only.	OK
CN_1004	Valid input: Status of advertisement is inactive. Condition: User edit the status of advertisement.	System will not appear advertisement which have inactive status.	OK

6.6 User Acceptance Testing (UAT)

In the context of CareerNest Management System, User Acceptance Testing (UAT) serves as a vital step to ensure that the system meets the end users' expectations and requirements. UAT is the stage where actual users, such as jobseekers and employers, interact with the system to verify that all functionalities work as intended in a real-world environment. The goal is to confirm that the system effectively enhances job application efficiency, improves communication between jobseekers and employers, and integrates user-friendly features like automated notifications and application management.

UAT is being implemented by using a Google Form survey to collect direct feedback from users. This method allowed to gather insights into how users perceive the system’s performance. The feedback is essential in identifying any gaps between user expectations and the actual system functionality. Since UAT focuses on validating the system's external behavior without delving into the internal code structure, it falls under the category of black box testing. In black box testing, the internal workings of the software are not examined; instead, the emphasis is on ensuring that the system behaves as expected from the user's perspective. Through the UAT process, I am ensured that the CareerNest Management System is ready for deployment by addressing any issues or concerns raised by the users, ultimately leading to a more polished and user-friendly system.

In Appendix XX, a Google Form survey was conducted with 31 respondents to assess their experience with the CareerNest Management System. The survey aimed to gather feedback on the system’s interface, functionality, and overall performance. The results show that most respondents are under 25 years old (74.2%), indicating a younger user base. The remaining respondents are between 25-39 years old (16.1%) and 40-59 years old (9.7%), with no participants aged 60 or above. Gender distribution revealed that 77.4% of the respondents are female, while 22.6% are male, reflecting a predominantly female user base.

In terms of education, most respondents have a Diploma (35.5%) or a bachelor’s degree (25.8%). Additionally, 29% have completed SPM, 6.5% hold STPM qualifications, and 3.2% have master’s Degrees or PhDs. This diverse educational background provides insight into the users' varying levels of education and expectations.

- Interface Feedback

The survey results indicate a generally positive reception of the interface of the CareerNest Management System. Navigation through the various sections is notably user-friendly, with 67.7% of respondents rating it a 5 out of 5. Similarly, the colors and fonts used in the system are seen as enhancing the user experience, with 58.1% giving them the highest rating. The clarity of buttons, menus, and links is also well-regarded, as 74.2% found them clearly labeled and easy to understand. The organization of information on each page received high praise, with 77.4% rating it as very well-organized. Consistency in the interface design across different sections is another strong point, with 74.2% expressing satisfaction.

- System Feedback

In terms of system performance and reliability, users are generally satisfied. A substantial 64.5% of respondents found it easy to find and access specific features or tools, giving this aspect a rating of 5. The system's reliability in performing tasks without errors or crashes was rated highly by 54.8% of users. The effectiveness of the system in notifying users about important updates, such as application status changes and new job postings, was positively rated by 71%. Similarly, the speed and efficiency of the system in processing requests, like job searches and application submissions, were praised by 71% of respondents. Despite the overall positive feedback, there were mentions of confusion regarding specific parts of the system, such as “Find Job” and “Update/Delete Applications,” though 83.9% did not report any difficulties.

- Knowledge Feedback

Regarding user understanding and guidance, the majority of respondents felt confident in their knowledge of the system. A significant 61.3% rated their understanding of the system's features and functionalities as excellent. Instructions and guidance provided by the system were also well-received, with 54.8% rating them as very clear. Most users (54.8%) felt familiar with navigating through the system to find information and tools, and 58.1% were confident in using the system for tasks such as job applications and managing job postings. Feedback on navigating advanced features is not yet fully completed, but the responses so far suggest a positive experience.

Overall, while the CareerNest Management System has been well-received, with high ratings for its interface, system reliability, and user understanding, there are areas for potential improvement, particularly in specific features and parts of the system where some users have reported confusion.

6.7 Conclusion

The testing chapter for the CareerNest Management System provides a comprehensive evaluation of the system's functionality, security, and user experience. It includes detailed test plans for each module, such as registration, login, and job posting, ensuring that each component performs correctly and integrates seamlessly. the chapter outlines various classes of tests, including unit, integration, and functional tests, to cover different aspects of the system. By employing both white box and black box testing approaches, the chapter ensures thorough validation of internal logic and external functionality. The results and analysis highlight the system's adherence to requirements, the effectiveness of its features, and areas for improvement. Overall, the testing chapter demonstrates that the CareerNest Management System is robust, reliable, and ready to deliver a smooth and efficient experience for its users.

CHAPTER 7: CONCLUSION

7.1 Introduction

This chapter provides a comprehensive conclusion to the development and analysis of the CareerNest Management System. It synthesizes the findings from the project, evaluating the system's effectiveness in enhancing job application processes, facilitating communication between jobseekers and employers, and improving overall recruitment efficiency. This conclusion will assess the system's strengths and weaknesses, propose potential areas for future improvements, and highlight the key contributions made by this project. By reflecting on the development journey and its impact, this chapter aims to offer a holistic understanding of the system's current performance and explore avenues for its further evolution.

7.2 Observation on Weaknesses and Strengths

1. Strengths

The CareerNest Management System exhibits several notable strengths that contribute to its overall effectiveness. Firstly, it streamlines the job application process, allowing jobseekers to efficiently search for, apply to, and manage their applications. This simplification enhances user experience and operational efficiency. For employers, the system provides a user-friendly interface to post job advertisements, review applications, and communicate with candidates, thereby improving the recruitment workflow.

Another significant strength is the system's automated notifications. These notifications keep jobseekers informed about changes in their application status, reducing the need for manual updates and fostering better communication. The system also offers customizable application management features, such as allowing employers to provide reasons for their decisions, which adds a layer of personalization and transparency to the application process.

Clear status updates are another advantage, as they provide jobseekers with immediate feedback regarding their applications and interview outcomes. This clarity reduces ambiguity and enhances user satisfaction. The user-friendly design of the system ensures that both jobseekers and employers, regardless of their technical expertise, can navigate the platform with ease.

Moreover, the system includes detailed tracking for application statuses, interview dates, and notifications, which helps maintain comprehensive records and improve management efficiency. Lastly, the system is designed with future enhancements in mind, offering flexibility to incorporate upgrades based on user feedback and evolving needs. These strengths collectively highlight the system's effectiveness in meeting the needs of its users and establishing a solid foundation for ongoing improvement.

2. Weaknesses

Despite its strengths, the CareerNest Management System has some weaknesses that could impact its overall performance. One notable area for improvement is the lack of a backup functionality. Without an effective backup system, there is a risk of data loss in the event of a system failure or other unforeseen issues, which could affect both jobseekers and employers.

Another weakness is the reliance on a single table for notifications within the application process. While this approach simplifies the system, it might lead to challenges in managing and scaling notifications as the number of users and transactions grows. A more robust notification management system could help address these potential scalability issues.

The system’s current design does not include advanced features for data analysis and reporting beyond basic visualization. Although it uses Microsoft Power BI for data visualization, it may benefit from more integrated analytics capabilities to provide deeper insights and better support decision-making processes.

Additionally, the user interface, while functional, may not fully cater to diverse user needs and preferences. There may be opportunities to enhance the interface to improve accessibility and usability for all users, including those with disabilities.

Finally, the system's approach to handling interview scheduling and responses could be further refined. For instance, jobseekers currently have limited options for interacting with interview details, which might restrict their flexibility and engagement with the process. Expanding these options could enhance the overall user experience.

7.3 Propositions for Improvement

CareerNest Management System can implement a robust backup functionality which is essential to enhance the system. A reliable backup system will safeguard against data loss due to system failures, human errors, or other unforeseen issues. Regular backups and effective recovery options will ensure the integrity of jobseeker and employer data, maintaining operational continuity and preventing disruptions that could impact user trust and system reliability.

Improving the notification management system is another crucial step. Currently, notifications are handled within a single table, which may limit scalability as the user base grows. By creating a dedicated notifications table or integrating a specialized notification service, the system can better manage and scale notifications. This enhancement will ensure that communication remains efficient and reliable, even as the volume of notifications increases.

Expanding the system's analytics capabilities will also provide significant benefits. While basic visualization tools like Microsoft Power BI are useful, integrating advanced analytics and reporting features will offer deeper insights into application trends, user behavior, and overall system performance. These enhancements will support more informed decision-making and strategic planning, allowing the system to adapt and evolve based on comprehensive data insights.

The user interface design needs improvement to enhance accessibility and usability. Conducting user experience research can help identify areas where the interface could be more inclusive and visually appealing. By making the design more user-friendly, the system can cater to a broader range of users, including those with disabilities, and ensure a more positive and efficient user experience.

Finally, enhancing interview scheduling features, strengthening data security, and providing comprehensive user support are important for overall system effectiveness. Allowing jobseekers to propose alternative interview times will improve flexibility and engagement. Strengthening data security through encryption and secure authentication will protect sensitive information. Additionally, offering detailed user guides, FAQs, and responsive support will help users navigate the system effectively and resolve issues promptly. Ensuring mobile compatibility and optimizing system performance will further enhance accessibility and efficiency, meeting user expectations and operational needs.

7.4 Project Contribution

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

From an academic perspective, the CareerNest Management System demonstrates the practical application of database management and software development methodologies in addressing real-world recruitment challenges. It serves as a case study and reference for future projects focused on job application systems, offering insights into the development process, the challenges encountered, and the solutions implemented. The system provides a tangible example of how theoretical concepts can be applied in practice, making it a valuable teaching tool for courses related to database management and software engineering.

In the context of the recruitment industry, the system offers a modern solution to many operational challenges faced by both jobseekers and employers. By enhancing operational efficiency, communication, and user engagement, the CareerNest Management System has the potential to set new standards in job application management. Its comprehensive feature set—including automated notifications, flexible application management, and advanced tracking—provides significant improvements over traditional methods, helping employers streamline their hiring processes and jobseekers navigate the application journey more effectively.

The user manual, detailed in Appendix XX, serves as a comprehensive guide for new users and administrators. It provides clear instructions on system setup, usage, and troubleshooting, ensuring that users can quickly and effectively adopt the system. This manual is an essential resource for facilitating a smooth transition to the CareerNest Management System and maximizing its benefits, ultimately supporting users in achieving their recruitment and job application goals.

Overall, the CareerNest Management System represents a significant advancement in the recruitment field, offering both practical benefits to users and valuable insights for academic study. Its innovative features and thoughtful design contribute to improved efficiency and effectiveness in job application management, setting a high standard for future systems in this domain.

7.5 Conclusion

To encapsulate, the CareerNest Management System has proven to be a valuable and effective tool in modernizing the job application and recruitment processes. By addressing key challenges faced by both jobseekers and employers, the system enhances operational efficiency, improves communication, and offers a user-centric experience. Its innovative features, such as automated notifications, flexible application management, and comprehensive tracking, provide significant improvements over traditional methods, setting a new benchmark for recruitment systems.

The project has made substantial contributions to both academic research and practical application. Academically, it serves as a case study in the application of database and software development methodologies, offering insights and a tangible example for future projects and educational purposes. Practically, it provides a modern solution that elevates industry standards and supports more effective recruitment practices.

Looking ahead, there are clear opportunities for further enhancement, including improvements in data security, user interface design, and system performance. Addressing these areas will ensure that the CareerNest Management System continues to meet evolving user needs and industry demands.

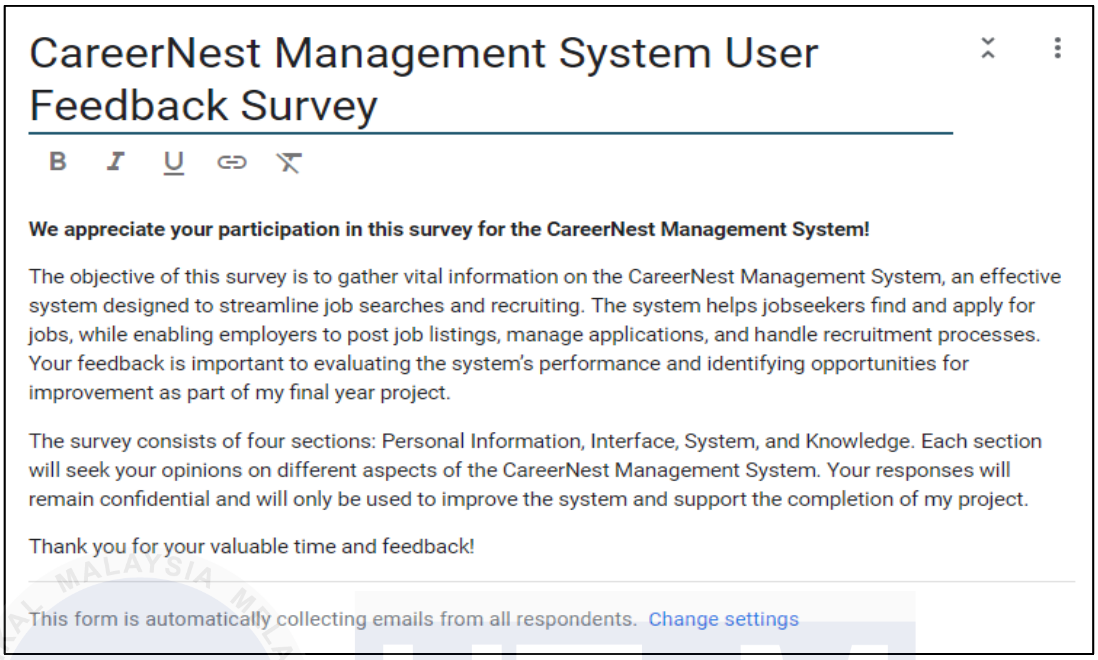
Overall, this project not only achieves its goal of improving job application efficiency but also sets the stage for continued innovation and development in the recruitment sector. The CareerNest Management System stands as a testament to the successful integration of technology and user-focused design, offering a robust and scalable solution for the challenges of modern recruitment.

REFERENCES

Ahmed, A. (2023), "A Review on the Impact of Online Job Portals on Employee Performance." *International Journal of Research, 10(8),* 370-377. https://doi.org/10.5281/zenodo.8311575

Amanova, C., Bulak, S., Msilu, F., & Demir, F. (2021). "Demystifying the User Experience: A Case Study on Online Job Search Engines." *American Journal of Qualitative Research, 5(2),* 233-250. https://doi.org/10.29333/ajqr/1141

Messalina, P., & Sieweke, J. (2021). "Recruiting digital talent: The strategic role of recruitment in organisations' digital transformation." *German Journal of Human Resource Management, 3(1),* 53-82. https://doi.org/10.1177/2397002220952734



CareerNest Management System User Feedback Survey

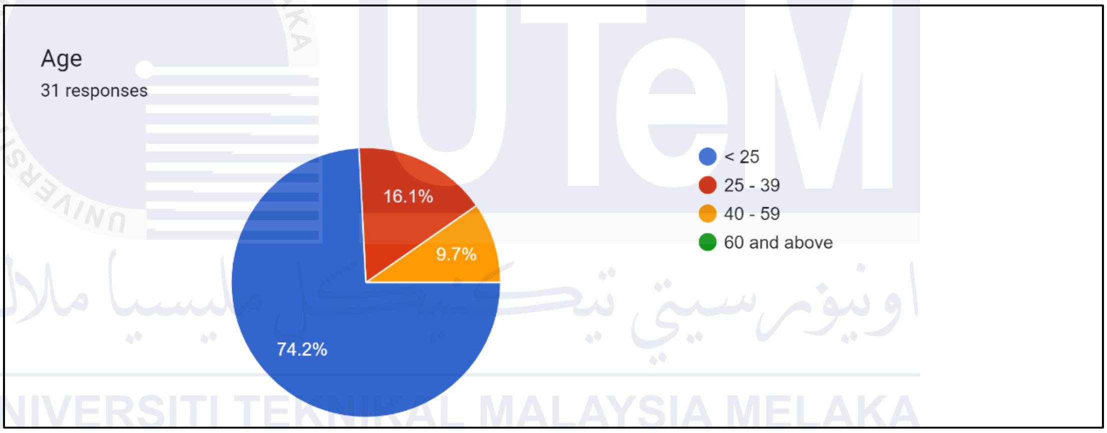
We appreciate your participation in this survey for the CareerNest Management System!

The objective of this survey is to gather vital information on the CareerNest Management System, an effective system designed to streamline job searches and recruiting. The system helps jobseekers find and apply for jobs, while enabling employers to post job listings, manage applications, and handle recruitment processes. Your feedback is important to evaluating the system's performance and identifying opportunities for improvement as part of my final year project.

The survey consists of four sections: Personal Information, Interface, System, and Knowledge. Each section will seek your opinions on different aspects of the CareerNest Management System. Your responses will remain confidential and will only be used to improve the system and support the completion of my project.

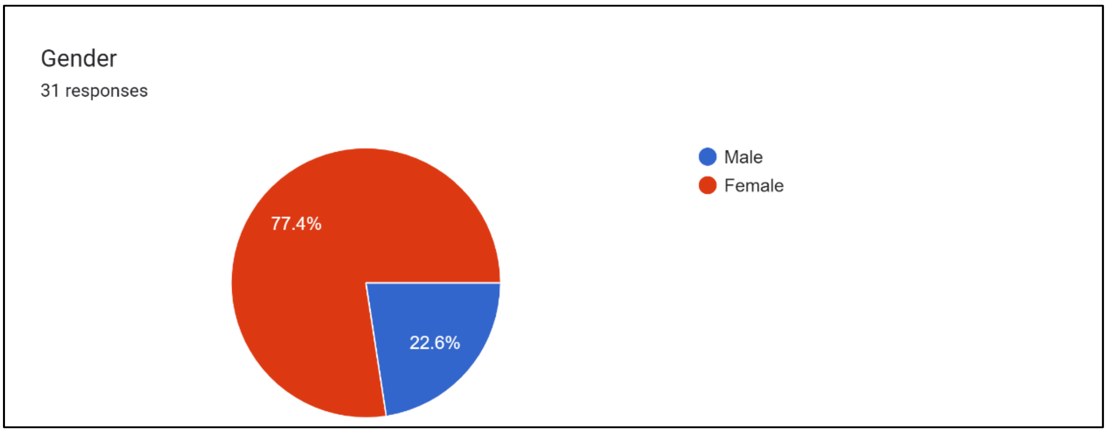
Thank you for your valuable time and feedback!

This form is automatically collecting emails from all respondents. Change settings



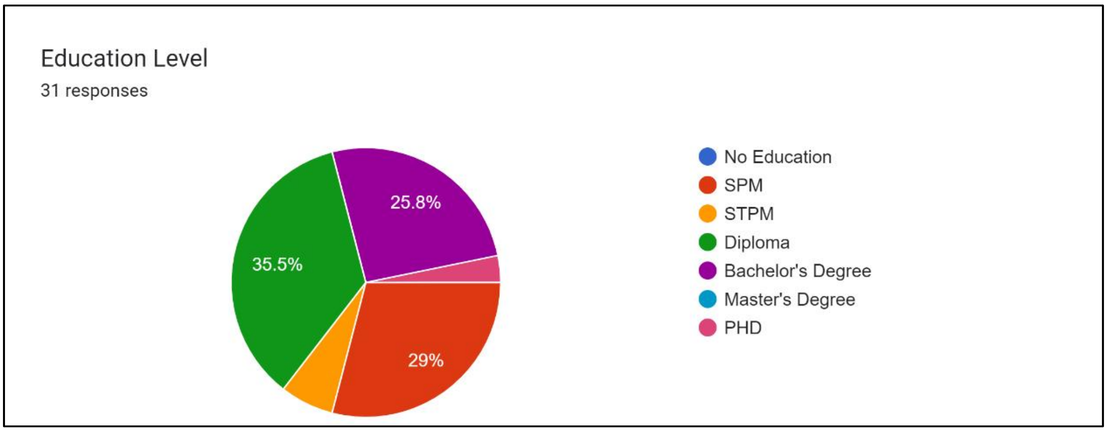
Age

31 responses



Gender

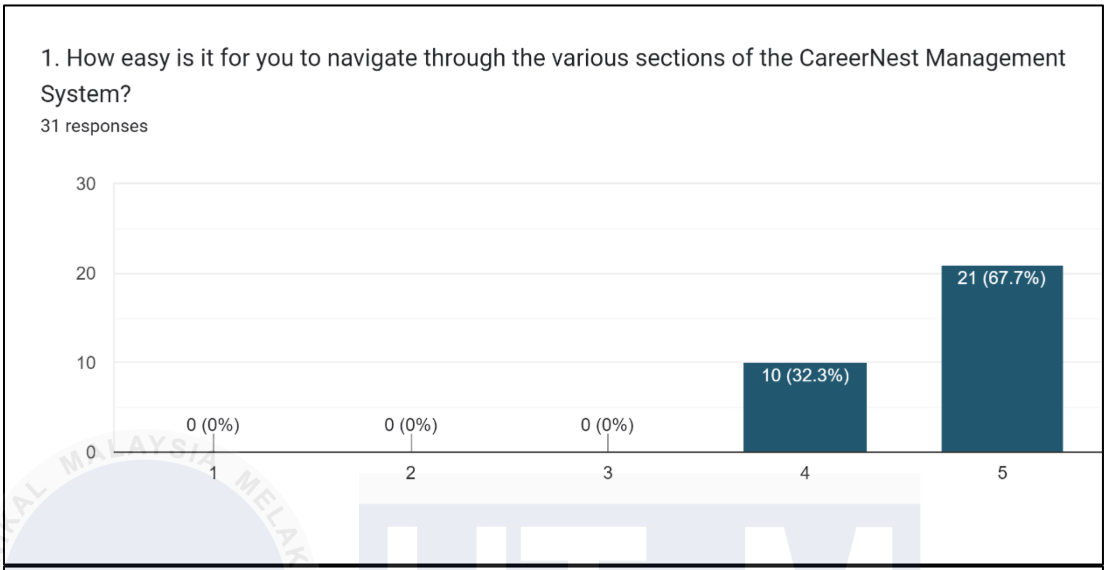
31 responses



Education Level

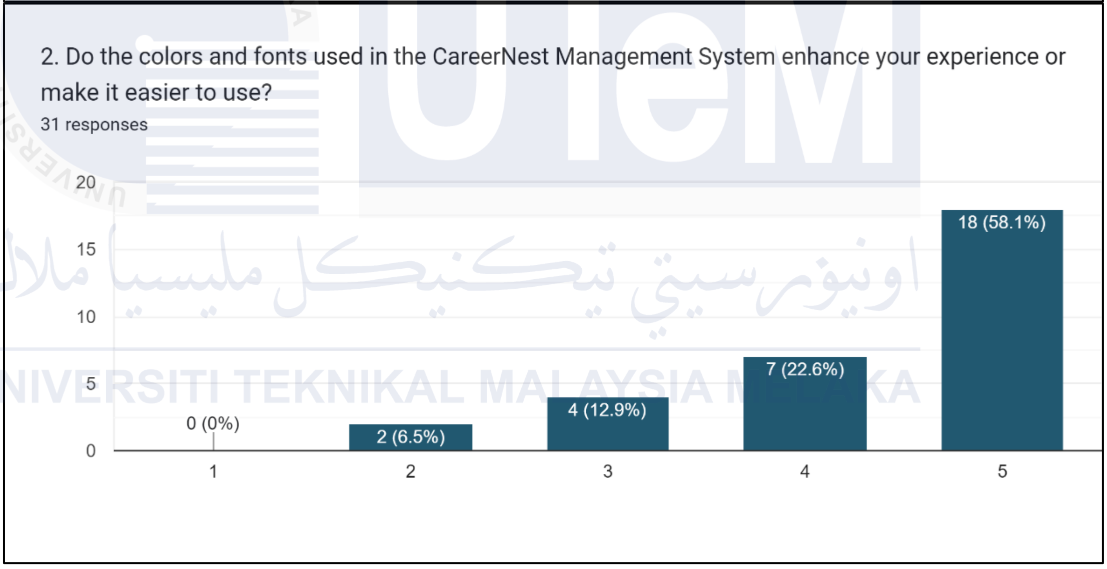
31 responses

i. Interface Feedback



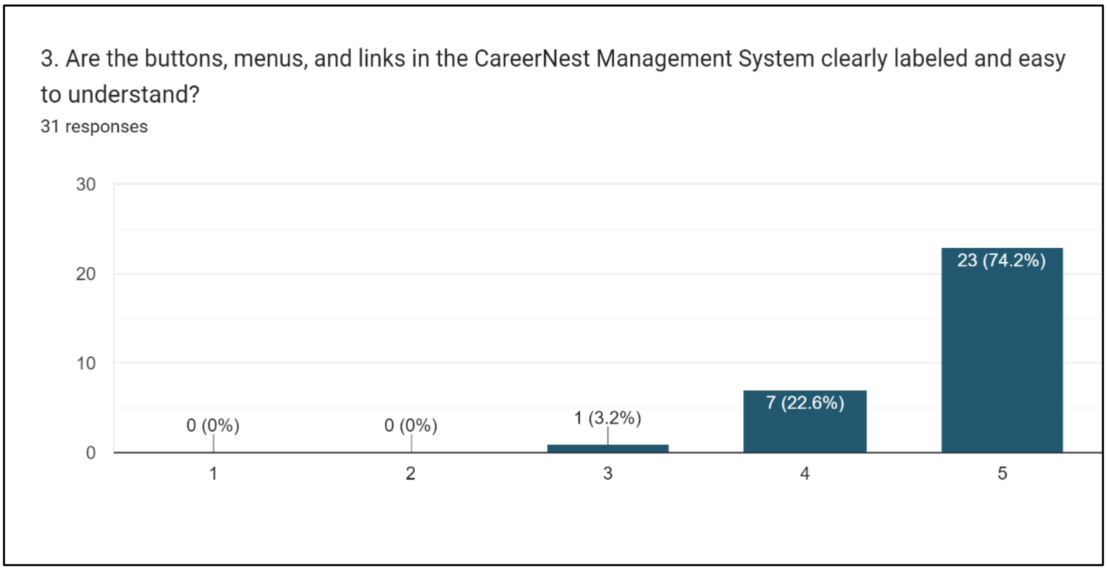
1. How easy is it for you to navigate through the various sections of the CareerNest Management System?

31 responses



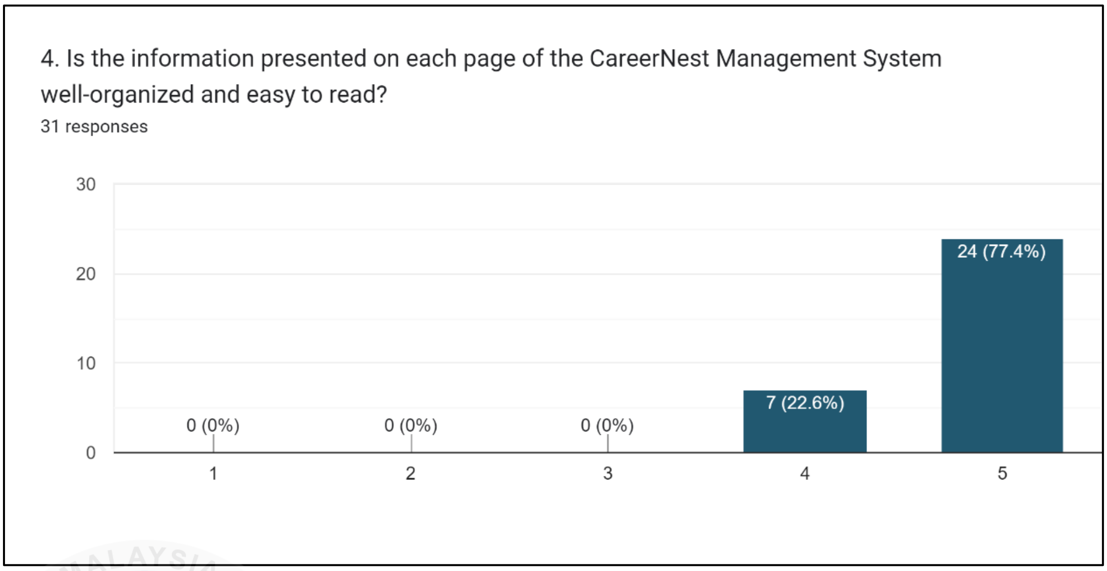
2. Do the colors and fonts used in the CareerNest Management System enhance your experience or make it easier to use?

31 responses



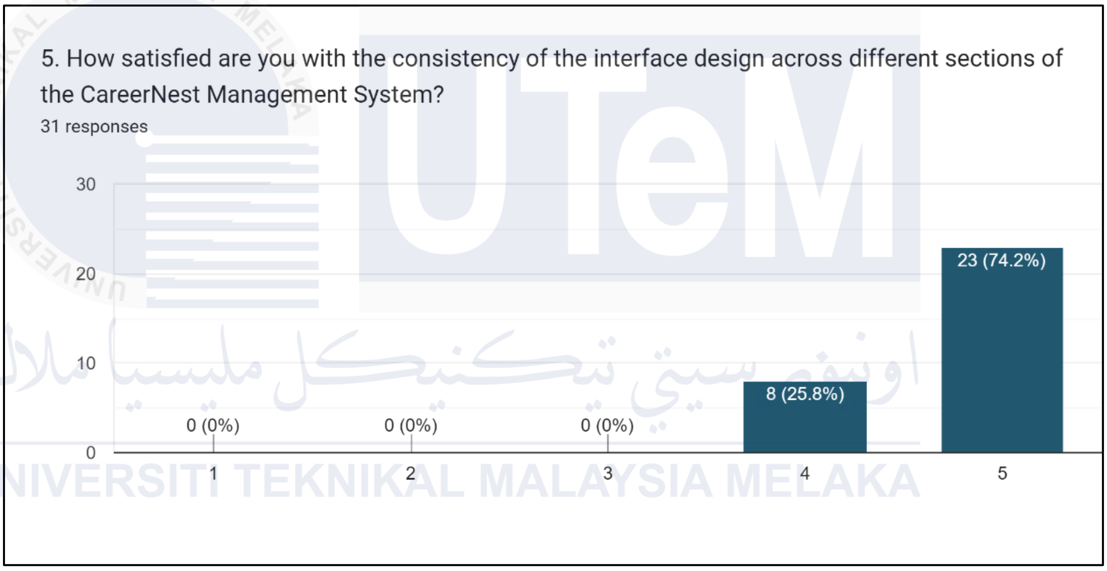
3. Are the buttons, menus, and links in the CareerNest Management System clearly labeled and easy to understand?

31 responses



4. Is the information presented on each page of the CareerNest Management System well-organized and easy to read?

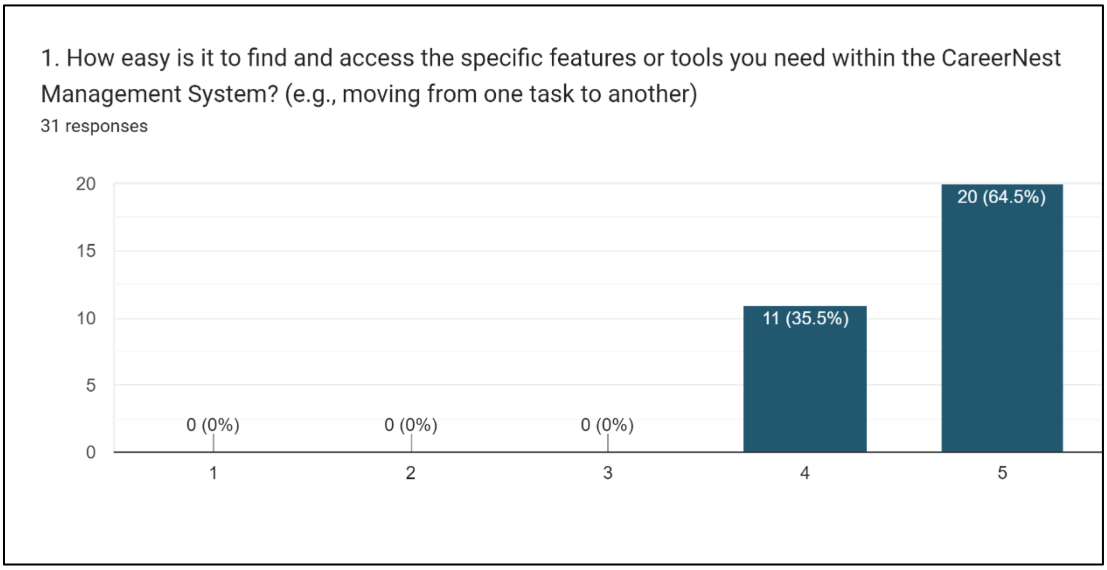
31 responses



5. How satisfied are you with the consistency of the interface design across different sections of the CareerNest Management System?

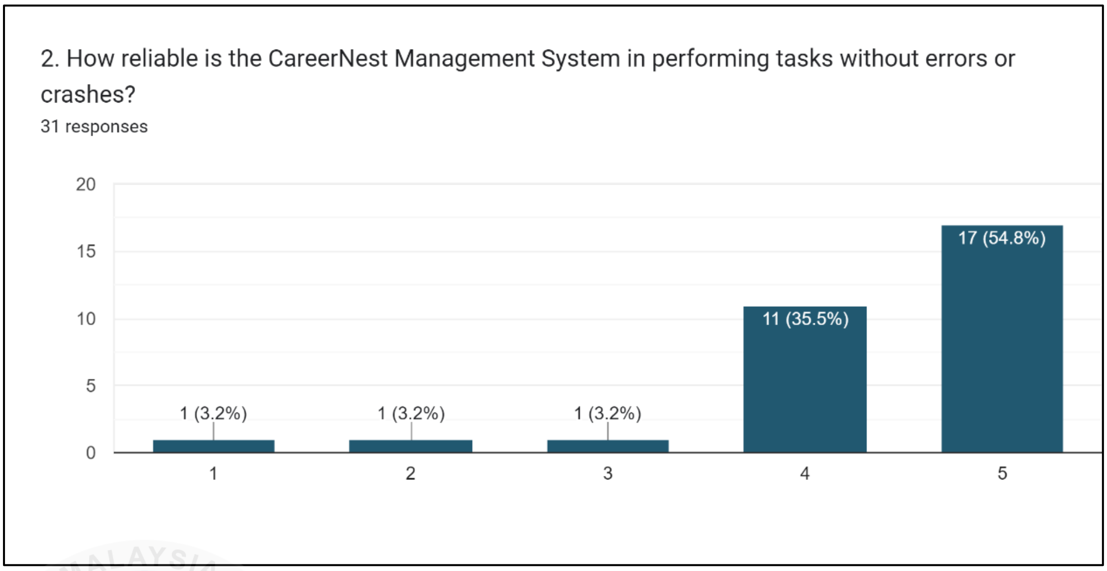
31 responses

ii. System Feedback



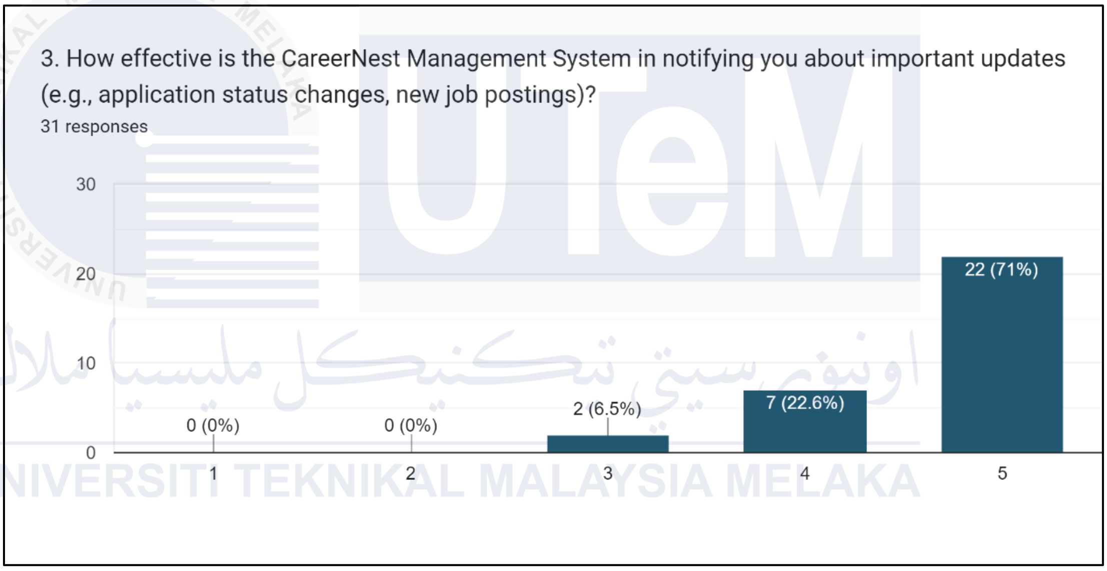
1. How easy is it to find and access the specific features or tools you need within the CareerNest Management System? (e.g., moving from one task to another)

31 responses



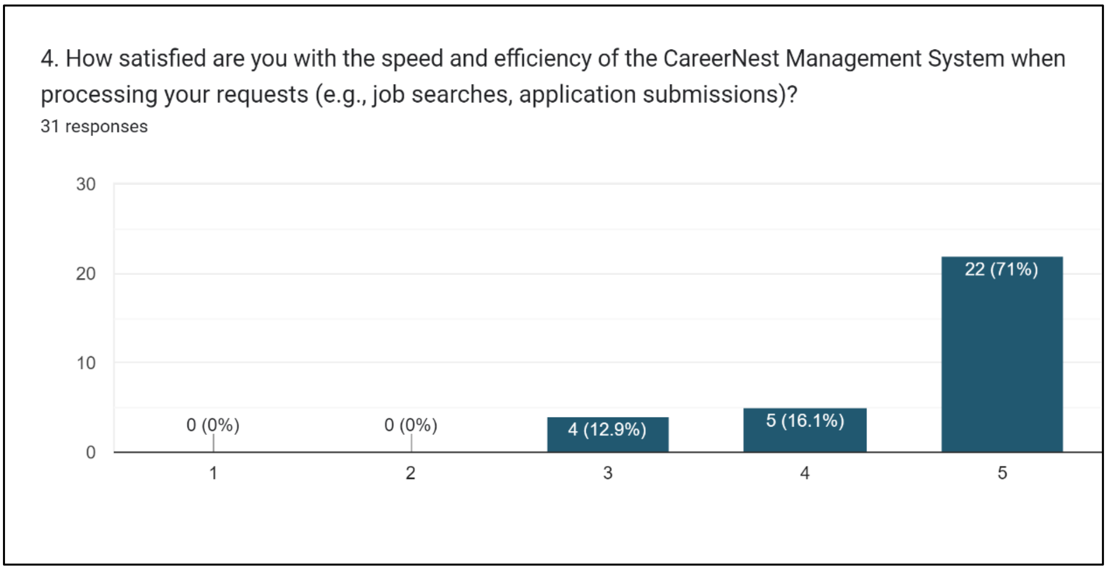
2. How reliable is the CareerNest Management System in performing tasks without errors or crashes?

31 responses



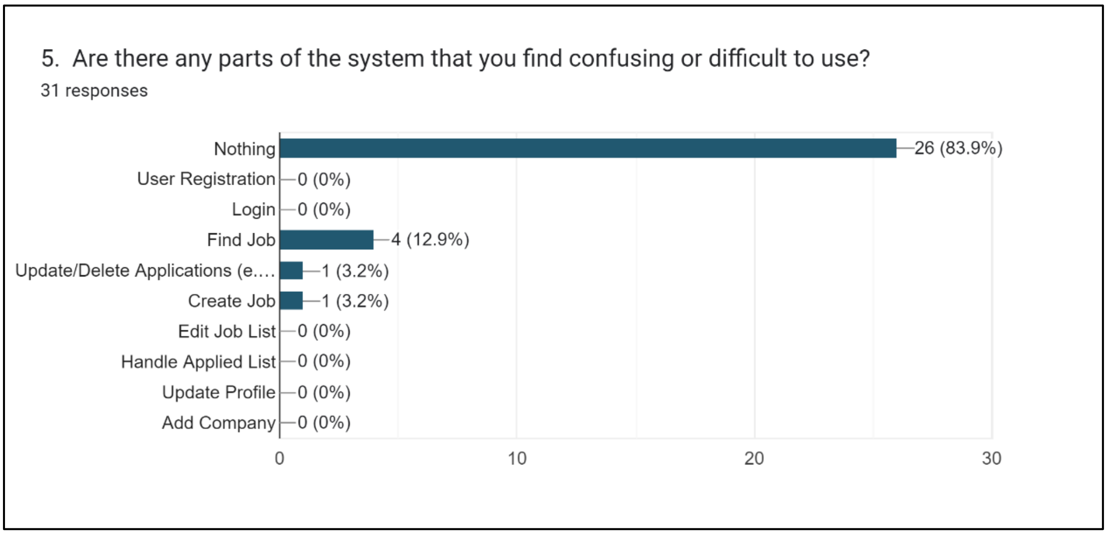
3. How effective is the CareerNest Management System in notifying you about important updates (e.g., application status changes, new job postings)?

31 responses



4. How satisfied are you with the speed and efficiency of the CareerNest Management System when processing your requests (e.g., job searches, application submissions)?

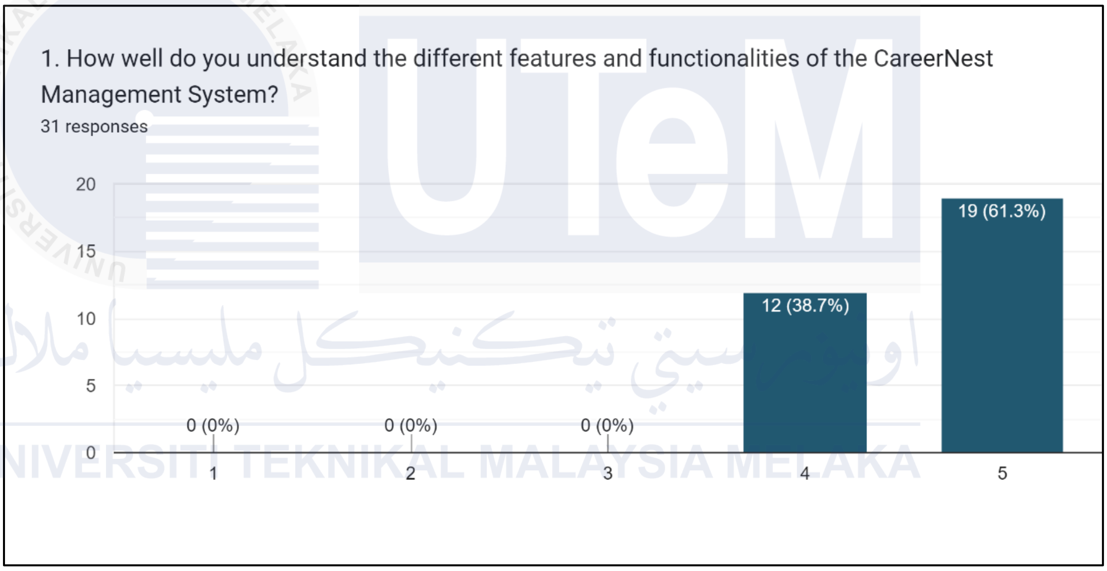
31 responses



5. Are there any parts of the system that you find confusing or difficult to use?

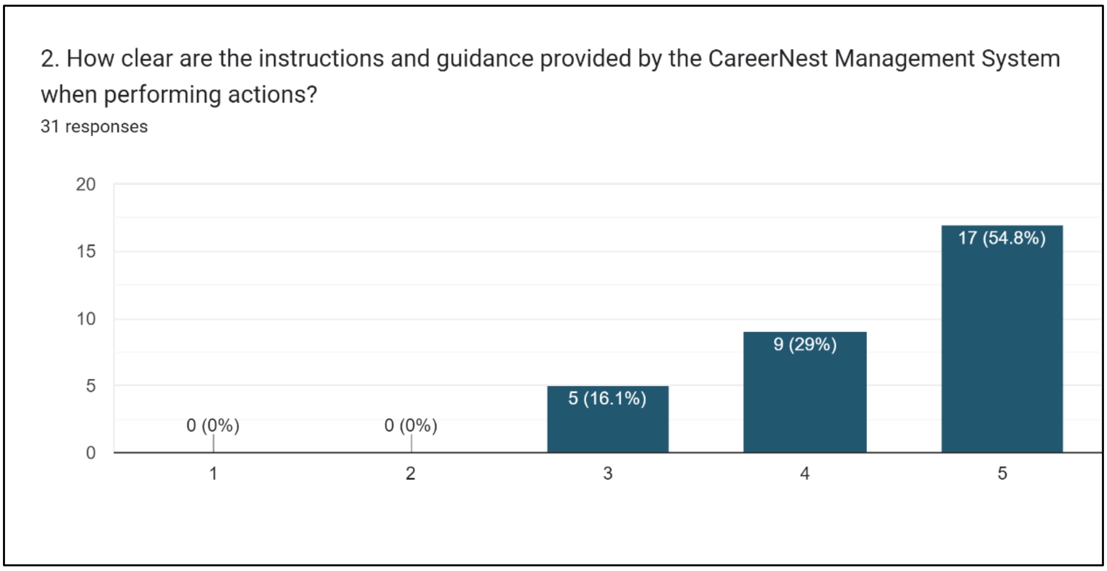
31 responses

iii. Knowledge Feedback



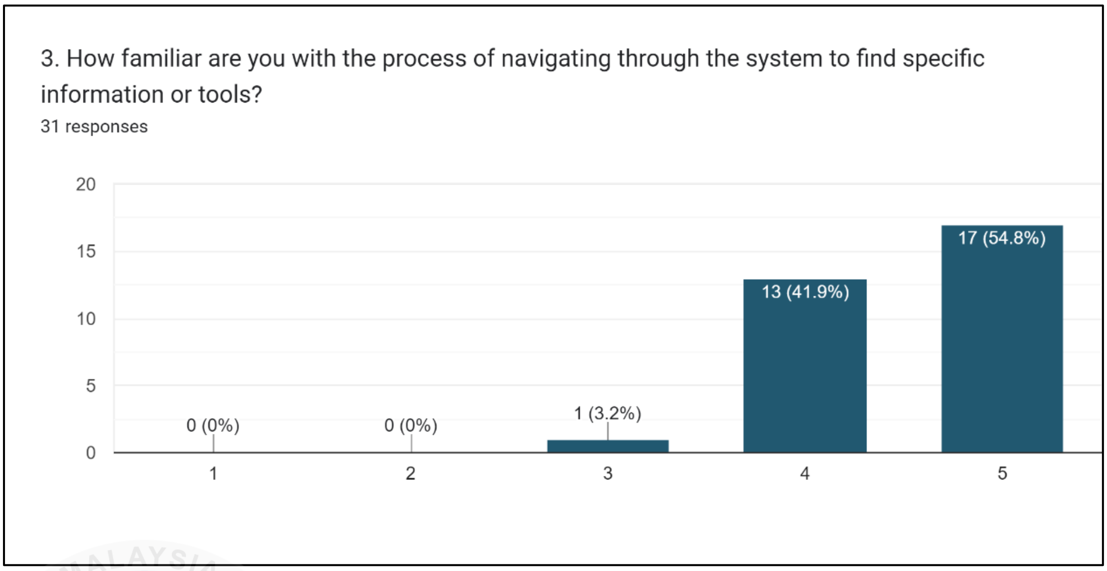
1. How well do you understand the different features and functionalities of the CareerNest Management System?

31 responses



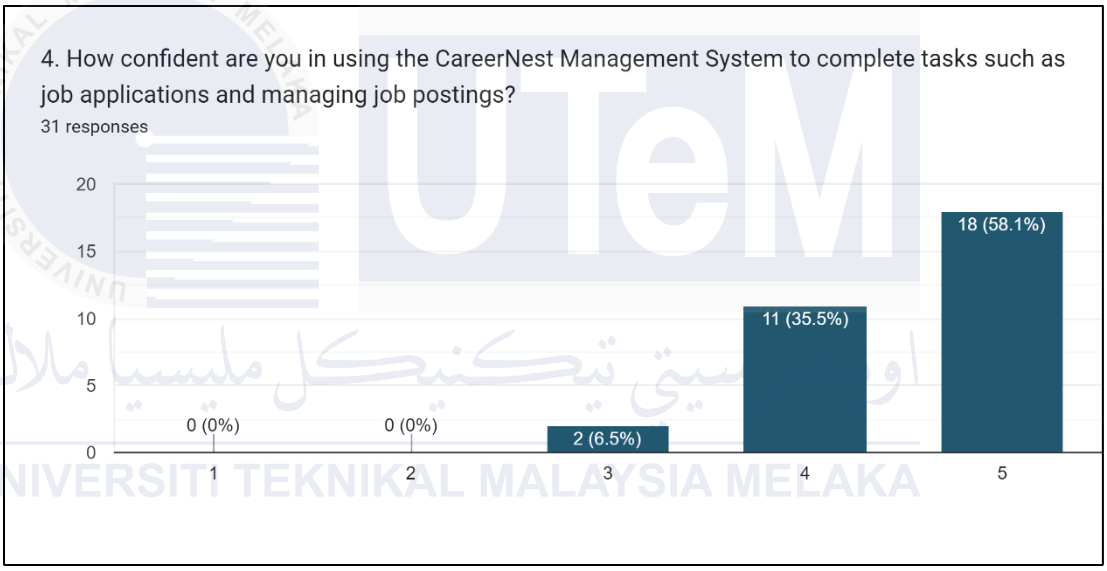
2. How clear are the instructions and guidance provided by the CareerNest Management System when performing actions?

31 responses



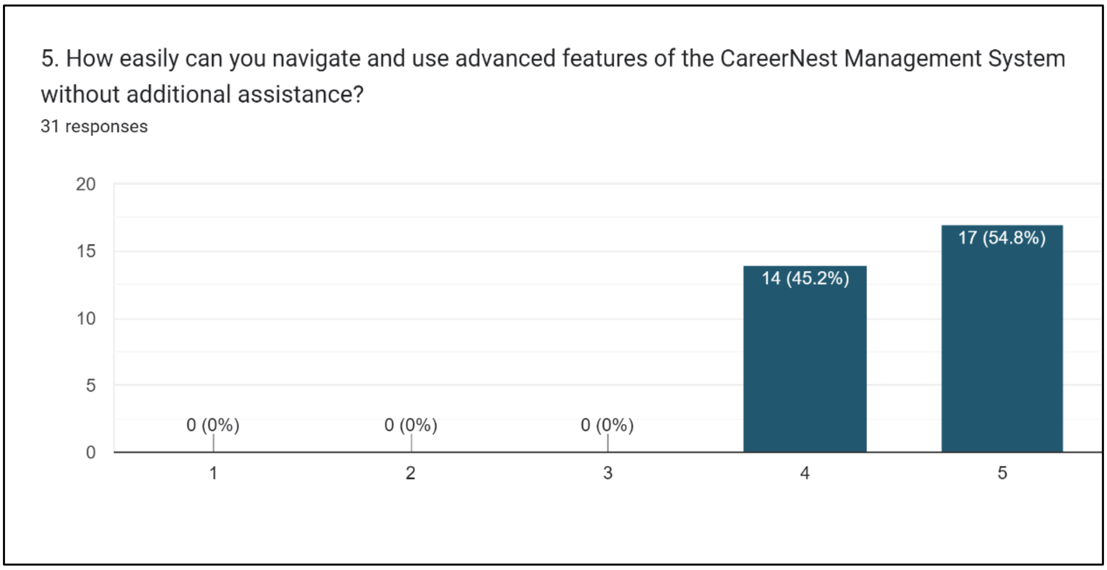
3. How familiar are you with the process of navigating through the system to find specific information or tools?

31 responses



4. How confident are you in using the CareerNest Management System to complete tasks such as job applications and managing job postings?

31 responses



5. How easily can you navigate and use advanced features of the CareerNest Management System without additional assistance?

31 responses