UTEM JOB PORTAL

SYED NOR IRFAN BIN SYED AHMAD SYAKIRIN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

UTeM JOB PORTAL

SYED NOR IRFAN BIN SYED AHMAD SYAKIRIN

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

UTeM JOB PORTAL

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

without citations.

STUDENT : SYED NOR IRFAN BIN SYED AHMAD SYAKIRIN Date : 30/08/24

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 : MRS NOOR AZILAH DRAMAN @ MUDA Date : 30/08/24

DEDICATION

I would like to express my profound gratitude to my family, especially my father, Syed Ahmad Syakirin Bin Syed Jamaluddin, and my mother, Nor Liza Binti Ahmad, for their unwavering support and encouragement. Their belief in me has been a constant source of motivation throughout this journey, and I sincerely hope that this project serves to enrich my database knowledge.

ACKNOWLEDGEMENTS

Alhamdulillah. praise to Allah for the strength and guidance throughout my Final Year Project (FYP) and report completion. First and foremost, I express my deepest appreciation to Mrs. Noor Azilah Draman @ Muda for her invaluable supervision and unwavering support throughout this project. Her guidance and insights have significantly enhanced the quality of my work. I am also grateful to my classmates, particularly those under Mrs. Noor Azilah's supervision, whose collaboration and support were crucial in overcoming challenges and completing this endeavor. Additionally, I extend my thanks to all lecturers and classmates who contributed, directly or indirectly, to the successful completion of this project.

Lastly, I extend my profound gratitude to Universiti Teknikal Malaysia Melaka (UTeM), for providing me with the opportunity to apply and refine my database management knowledge through this project. Special recognition is due to the Final Year Project (FYP) committee for their support in facilitating the necessary preparations for the FYP and its report.

ABSTRACT

The development of a Job Portal web application addresses the need for a centralized platform specifically designed for graduates of Universiti Teknikal Malaysia Melaka (UTeM). The current challenge faced by UTeM alumni lies in the fragmented job search process and the lack of a dedicated platform that connects them directly with potential employers. This project focuses on solving this problem by creating a secure, user-friendly, and efficient web-based portal. The application allows administrators to manage user accounts, oversee job postings, and generate insightful reports. Employers can post job vacancies, manage applications, and interact with potential candidates, while alumni can register, search, and apply for jobs, as well as track the status of their applications. The platform is built with a normalized relational database to ensure effective data management and optimized performance. Additionally, stringent security measures, including encryption and role-based access control, are implemented to safeguard user data. The project's development process includes rigorous testing to ensure the system's functionality and stability. The resulting Job Portal successfully provides a reliable and responsive platform that meets the specific needs of UTeM alumni and employers.

ABSTRAK

Pembangunan aplikasi web Portal Kerjaya ini bertujuan untuk memenuhi keperluan platform berpusat yang direka khusus untuk graduan Universiti Teknikal Malaysia Melaka (UTeM). Cabaran semasa yang dihadapi oleh alumni UTeM adalah proses pencarian pekerjaan yang terpecah-pecah dan kekurangan platform khusus yang menghubungkan mereka secara langsung dengan majikan yang berpotensi. Projek ini memberi tumpuan kepada penyelesaian masalah ini dengan mencipta portal web yang selamat, mesra pengguna, dan cekap. Aplikasi ini membolehkan pentadbir mengurus akaun pengguna, menyelia iklan pekerjaan, dan menjana laporan yang bermaklumat. Majikan boleh memaparkan kekosongan jawatan, mengurus permohonan, dan berinteraksi dengan calon yang berpotensi, manakala alumni boleh mendaftar, mencari, dan memohon pekerjaan serta menjejak status permohonan mereka. Platform ini dibina dengan pangkalan data hubungan yang dinormalisasi untuk memastikan pengurusan data yang berkesan dan prestasi yang optimum. Tambahan pula, langkah-langkah keselamatan yang ketat, termasuk penyulitan dan kawalan akses berasaskan peranan, dilaksanakan untuk melindungi data pengguna. Proses pembangunan projek ini melibatkan ujian yang teliti untuk memastikan fungsi dan kestabilan sistem. Portal Kerjaya yang dihasilkan berjaya menyediakan platform yang boleh dipercayai dan responsif yang memenuhi keperluan khusus alumni dan majikan UTeM.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

FYP	**-**	**Final Year Project**

LIST OF ATTACHMENTS

CHAPTER 1: INTRODUCTION

1.1 Introduction

The UTeM Job Portal website is designed to serve as a comprehensive job search platform specifically for the alumni of University Teknikal Malaysia Melaka (UTeM). This project aims to bridge the gap between recent graduates and potential employers by providing a centralized, user-friendly interface for job postings and applications. The portal is intended to generate additional revenue for UTeM while enhancing the career prospects of its graduates through streamlined job search and application processes.

1.2 Problem Statement(s)

i. UTeM graduates face challenges in finding job opportunities due to scattered job listings across multiple websites, leading to a slow and frustrating job search process.

ii. Employers struggle to quickly find and hire UTeM graduates because of the lack of direct access to a central pool of qualified candidates, making the hiring process inefficient.

iii. UTeM has difficulty staying connected with employers, limiting job placement opportunities and affecting the university's ability to support graduates and enhance its reputation.

1.3 Objective

i. To create a centralized job portal that gathers all job listings in one place, making it easier and faster for UTeM graduates to find and apply for relevant job opportunities.

ii. To provide employers with direct access to a central pool of UTeM graduates, streamlining the hiring process and enabling them to quickly find and recruit qualified candidates.

iii. To enhance UTeM's engagement with employers by developing a website that facilitates easy communication and collaboration, leading to more job placements, and partnerships that benefit both the university and its graduates.

1.4 Scope

i. User Registration and Management:

- Explanation: This includes the registration and management of alumni (applicants) and employers (companies) to facilitate secure and personalized access. It ensures that all users have unique profiles that can be managed and updated as needed.

ii. Job Posting and Management:

- Explanation: Allows employers to post job advertisements and manage them efficiently, providing alumni with updated job opportunities. Employers can edit, update, or delete job postings as necessary, ensuring that the job listings remain current and relevant.

iii. Job Application Process:

- Explanation: Enables alumni to browse, search, and apply for jobs, streamlining the application process. The system supports features like job filters, application tracking, and notifications to keep applicants informed about the status of their applications.

iv. Feedback and Rating System:

- Explanation: Collects feedback from users to improve the platform and provide valuable insights. This helps in continuously enhancing the user experience by addressing issues and implementing suggested improvements.

v. Payment and Subscription Management:

- Explanation: Handles payment transactions for advertisement plans, generating revenue for the university. This includes managing subscriptions, processing payments, and providing invoices to employers.

vi. Messaging System:

- Explanation: Facilitates communication between alumni, employers, and administrators. The messaging system supports both direct messages and group communications, helping users to stay connected and engaged.

vii. Other Specific Entities and Platform:

- Hardware and Software Tools:

    - Hardware: Standard hardware components such as personal computers or servers capable of running the required software stack for development and testing purposes.

- Software:

  - XAMPP: A cross-platform web server solution bundle that includes Apache, MySQL, PHP, and Perl, facilitating local server setup and development environment configuration.

  - PHP (Hypertext Preprocessor): A server-side scripting language widely used for web development, allowing dynamic content generation and interaction with databases.

  - MySQL: An open-source relational database management system (RDBMS) used for storing and retrieving data efficiently in a structured manner.

  - HTML / CSS (HyperText Markup Language / Cascading Style Sheets): Core technologies for building the front-end of web applications, defining the structure and styling of web pages.

  - JavaScript: A versatile scripting language commonly used for client-side web development to add interactivity and dynamic behaviour to web pages.

1.5 Project Significance

i. Alumni:

- Benefit: Simplified job search and application process, leading to better employment opportunities.
- How: Provides a centralized platform with comprehensive job listings and easy application features, reducing the time and effort required to find and apply for jobs.

ii. University:

- Benefit: Generates additional revenue and enhances its reputation.
- How: Through subscription fees from employers and successful job placements of graduates, the university can gain financial benefits and improve its standing as a supportive institution for its alumni.

iii. Employers:

- Benefit: Access to a pool of qualified candidates from UTeM.
- How: By posting job advertisements and receiving applications from alumni, employers can find and recruit suitable candidates more efficiently.

1.6 Expected Output

i. Output 1: Centralized job postings from various sources, providing alumni with a comprehensive view of available opportunities.

- Details: Implemented through aggregating job listings from different platforms into the U-Job Portal, ensuring that all relevant job opportunities are available in one place.

ii. Output 2: Streamlined application process allowing alumni to apply for multiple jobs using a single profile.

- Details: Includes features like pre-filled application forms and profile-based applications, reducing redundancy and making the application process more efficient.

iii. Output 3: Enhanced engagement between UTeM and employers, resulting in more internships, partnerships, and job placements.

- Details: Facilitated by features that allow communication and collaboration, such as messaging and feedback systems, creating opportunities for meaningful interactions and collaborations.

1.7 Conclusion

This chapter outlined the introduction, problem statements, objectives, scope, significance, and expected outputs of the UTeM Job Portal project. The project aims to address key issues related to job search efficiency, application process, and university-employer engagement. By centralizing job postings, streamlining the application process, and enhancing communication between the university and employers, the U-Job Portal will provide significant benefits to alumni, the university, and employers. The next steps involve a detailed literature review and the development of methodologies to achieve the project goals.

CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

This chapter provides an overview of the methodology and planning involved in the development of the UTeM Job Portal project. It outlines the Database Life Cycle (DBLC) phases that will be used, detailing each phase's tasks and their execution. The purpose of this methodology is to ensure a structured and systematic approach to the project, facilitating effective database management, implementation, and maintenance.

2.2 Project Methodology

The Database Life Cycle (DBLC) consists of six key phases, each critical to the successful development and operation of the UTeM Job Portal. These phases are:

Table 2.1: Database Life Cycle Phases

Database Initial Study	This phase involves identifying and documenting the issues that the UTeM Job Portal aims to solve, such as inefficient job searches and application processes. The first step in this phase is to conduct a thorough analysis of the existing job portal landscape, including platforms like MYFuture Jobs, LinkedIn, Hiredly, JobStreet, Indeed, and GrabJobs. This analysis helps in understanding the current market standards, user expectations, and gaps that the U-Job Portal can address. Following this, a feasibility study is conducted to assess the project's technical, economic, and operational viability. This includes a detailed cost-benefit analysis, which examines the potential costs and expected benefits of the project, ensuring that the investment is justified. Additionally, the feasibility study evaluates the technical resources and skills available, identifying any gaps that need to be filled to ensure the project's success. Stakeholder interviews are a crucial part of this phase, as they provide valuable insights into the needs and expectations of the primary users—alumni, employers, and university administrators. These interviews help in gathering detailed requirements and setting realistic goals for the project.

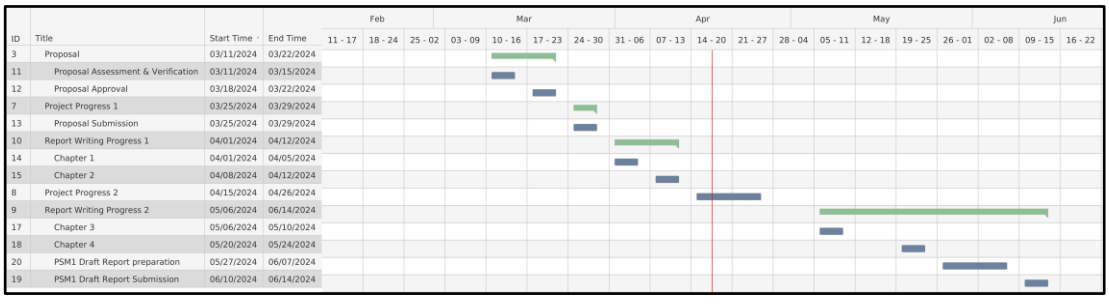
Database Design	The design phase starts with the development of a high-level Entity-Relationship Diagram (ERD) that outlines the main entities and their relationships within the UTeM Job Portal. This ERD serves as a blueprint for the database structure, ensuring that all necessary entities, such as users (applicants and employers), job postings, applications, and feedback, are accurately represented. Following the conceptual design, the project moves into the logical design phase, where the conceptual schema is transformed into a detailed logical schema. This involves defining tables, fields, primary keys, and foreign keys, ensuring that the database structure is both efficient and scalable. Normalization techniques are applied to eliminate data redundancy and ensure data integrity. The final step in the design phase is the physical design, where the logical schema is translated into a physical schema. This involves selecting appropriate data types, indexing strategies, and partitioning schemes to optimize performance, storage, and access speed. Considerations for backup and recovery, security, and data access are also addressed in this phase to ensure the database's robustness.

Implementation & Loading	In the implementation and loading phase, the database environment is set up using the selected Database Management System (DBMS), MySQL. This involves installing and configuring MySQL on the designated server and creating the database schema based on the physical design. Once the environment is set up, the initial data is imported into the database. This step involves preparing data migration scripts that ensure data integrity during the loading process. These scripts are carefully designed to handle data validation and transformation, ensuring that the data is clean and consistent. The loaded data is then validated against the original sources to ensure accuracy and completeness. This phase also includes the development of necessary backend functionalities, such as stored procedures, triggers, and views, which are essential for the database's operation. Proper documentation of the implementation process is maintained to facilitate future maintenance and troubleshooting.

Testing & Evaluation	The testing and evaluation phase is critical for ensuring that the database performs as expected and meets the users' requirements. This phase begins with unit testing, where individual database components, such as tables, queries, and stored procedures, are tested for correctness. Test cases are developed for each component, and any issues identified are promptly addressed. Following unit testing, integration testing is conducted to ensure that different parts of the database work together seamlessly. This involves testing the interactions between tables, relationships, and constraints to verify the overall database integrity. User Acceptance Testing (UAT) is then carried out with actual users to validate that the database meets their requirements. Feedback from UAT sessions is collected and used to make necessary adjustments and improvements. This phase ensures that the database is not only technically sound but also user-friendly and effective in meeting the project's objectives.
Operation	The operation phase involves deploying the database on the production server and ensuring all necessary configurations are in place. This phase includes setting up the production environment, transferring the database from the development environment, and performing any required configurations to ensure smooth operation. User training is provided to end-users and administrators to ensure they can effectively interact with the system. This training includes tutorials, user manuals, and hands-on sessions to familiarize users with the database's functionalities and features. Additionally, this phase involves setting up monitoring tools to track database performance and usage, ensuring that any issues can be promptly identified and addressed.

Maintenance & Evaluation	The maintenance and evaluation phase are an ongoing process that ensures the database continues to perform optimally and meets evolving requirements. This phase includes regular performance monitoring to identify and resolve any issues that may arise. Monitoring tools are used to track performance metrics, such as query response times, resource usage, and error rates, and set up alerts for potential problems. A robust backup and recovery strategy is implemented to protect data, including scheduling regular backups and testing recovery procedures to ensure data can be restored in case of a failure. Periodic evaluations are conducted with stakeholders to gather feedback and implement necessary updates or enhancements. These evaluations help in identifying new requirements, optimizing existing functionalities, and ensuring that the database continues to meet the users' needs and expectations. Documentation is continuously updated to reflect any changes or improvements made to the database.

2.3 Project Schedule and Milestones



ID	Title	Start Time	End Time
3	Proposal	03/11/2024	03/22/2024
11	Proposal Assessment & Verification	03/11/2024	03/15/2024
12	Proposal Approval	03/18/2024	03/22/2024
7	Project Progress 1	03/25/2024	03/29/2024
13	Proposal Submission	03/25/2024	03/29/2024
10	Report Writing Progress 1	04/01/2024	04/12/2024
14	Chapter 1	04/01/2024	04/05/2024
15	Chapter 2	04/08/2024	04/12/2024
8	Project Progress 2	04/15/2024	04/26/2024
9	Report Writing Progress 2	05/06/2024	06/14/2024
17	Chapter 3	05/06/2024	05/10/2024
18	Chapter 4	05/20/2024	05/24/2024
20	PSM1 Draft Report preparation	05/27/2024	06/07/2024
19	PSM1 Draft Report Submission	06/10/2024	06/14/2024

Figure 2.1: Gantt Chart

Table 2.2: Milestones

	Milestones	Expected Documents	Dates
1.	Problems identification and analysis	(a) Flow chart of the current system (b) Flow chart of the proposed system (c) DFD of the proposed system (d) Requirement of the proposed system (Functional, non-functional and devices)	6-April-2024
2.	Conceptual design of the proposed system	A complete ERD and Database	27-April-2024
3.	Implementation to the system	A complete GUI of the system	11-May-2024
4.	Testing of the system	A draft of final report of the system	26-June-2024
5.	Maintenance of the system	A complete final report of the system	27-June-2024

2.4 Conclusion

This chapter outlined the methodology and planning for the UTeM Job Portal project, detailing the Database Life Cycle (DBLC) phases and the tasks involved in each phase. By following this structured approach, the project aims to ensure the development of a robust and efficient database system. The next chapter will focus on the analysis of the problems identified, proposed improvements, and requirements of the new system.

CHAPTER 3: ANALYSIS

3.1 Introduction

The analysis phase is crucial for understanding the current state of the UTeM Job Portal project and identifying areas for improvement. This chapter provides an in-depth examination of the existing system, highlights the problems encountered, and proposes solutions for enhancing the system. The analysis phase involves investigating the current scenario, proposing improvements, and conducting a detailed requirement analysis of the to-be system. This structured approach ensures that the proposed system meets user needs effectively and efficiently.

3.2 Problem Analysis

The current (as-is) system scenario involves various challenges and inefficiencies that hinder the effective functioning of the UTeM Job Portal. One of the primary issues is the fragmented job search and application process, which results in a poor user experience for both applicants and employers. Additionally, there are problems related to data management, such as data redundancy and inconsistency, which affect the accuracy of information and reports generated by the system.

3.2.1 System Context Diagram Of The Existing System

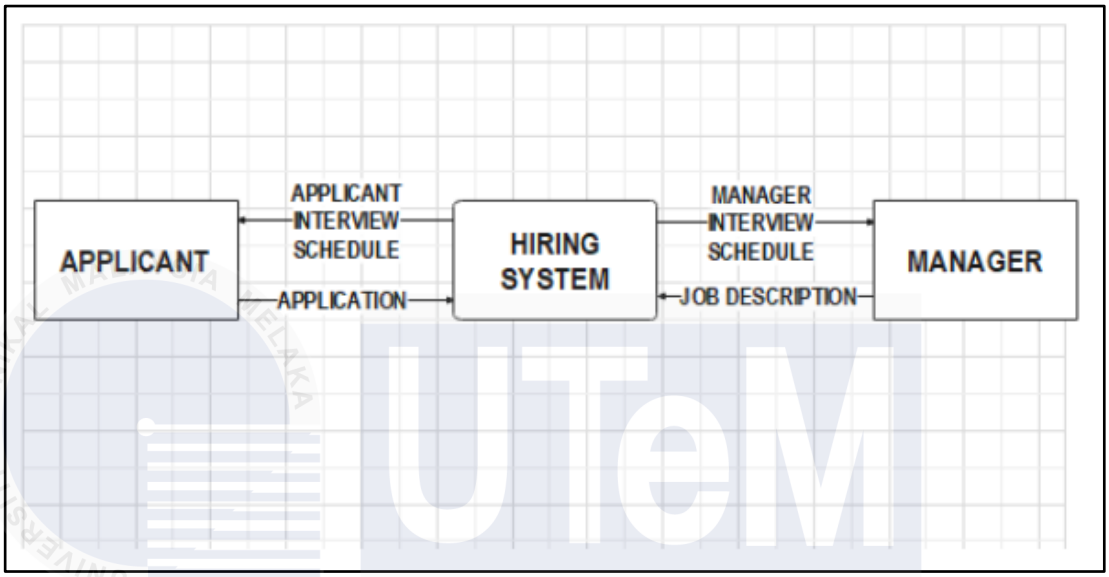


Figure 3.1: Context Diagram

3.2.2 Data Flow Diagram Of The Existing System

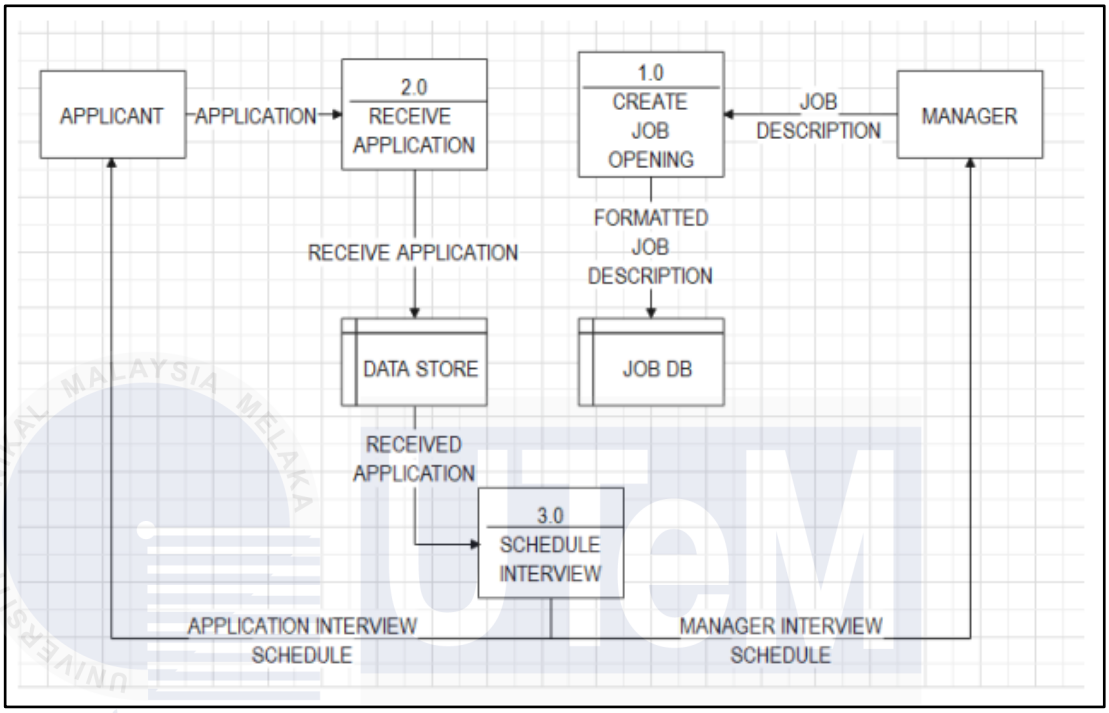


Figure 3.2: Data Flow Diagram

The existing job search systems are cost-effective, providing a shorter recruiting cycle time, and requiring less technical know-how for users. They offer the advantage of targeting specific audiences or niches, thereby reaching a more comprehensive range of applicants. This broad reach allows employers to access a diverse pool of candidates efficiently, making it easier to fill job openings.

Despite the strengths, the existing job search systems have notable weaknesses. They lack mechanisms or modules to help job seekers understand the companies they need, and they are not self-sufficient in providing platforms that facilitate effective communication of recruiters' needs to job seekers. Additionally, while these systems aim to show a wide array of job opportunities, they fail to filter and present job opportunities tailored to the specific preferences and qualifications of the job seekers, leading to inefficiencies in the job search process.

3.3 The Proposed Improvements / Solutions

To address the identified problems, the proposed improvements focus on enhancing the user experience, optimizing data management, and integrating the system with external platforms. The solutions aim to streamline the job search and application process, ensure data integrity, and expand the portal's reach.

Proposed System Flowchart Diagram:

Illustrates the improved process flowchart from job search to application, highlighting streamlined steps and enhanced user experience. This diagram shows the new features and functionalities added to the system, such as advanced search filters, job recommendations, and integration with external platforms.

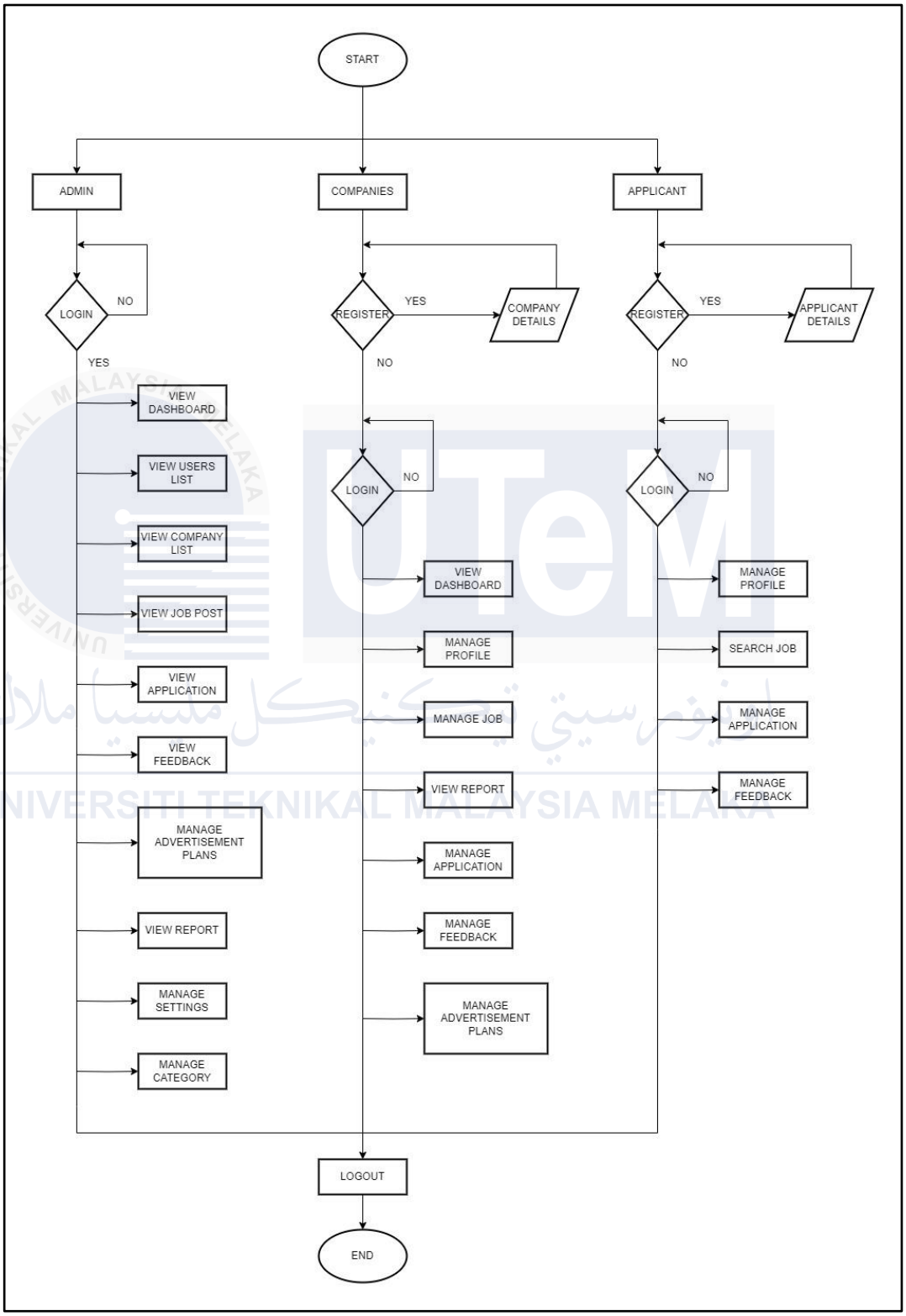


Figure 3.3: Flow Chart

3.4 Requirement Analysis of the To-Be System

3.4.1 Functional Requirement (Process Model)

The functional requirements of the to-be system specify how it records, computes, transforms, and transmits data to support its intended functions. This can be illustrated using Data Flow Diagrams (DFDs) or Unified Modelling Language (UML) diagrams, mapping out the flow of data and interactions between system components. Key functions include user registration, job search, application submission, and administrative management, each supported by corresponding data processing and communication processes.

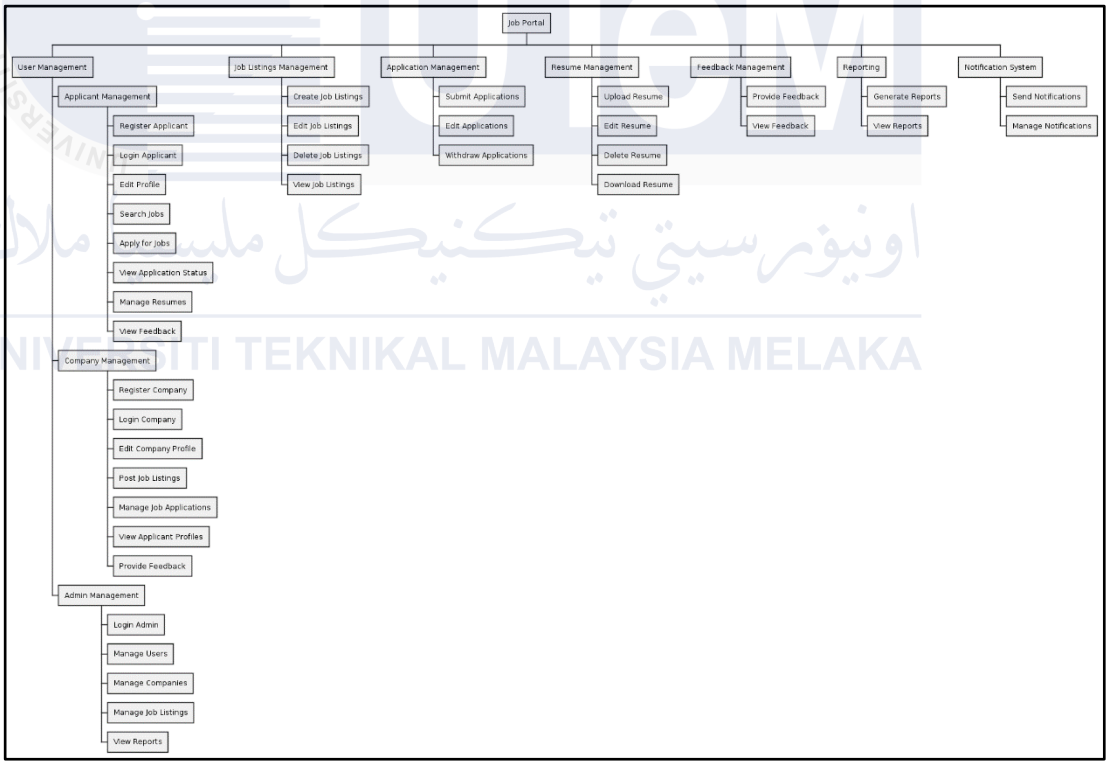


Figure 3.4: Structure Chart

Diagram:

- Data Flow Diagram (DFD):

  Illustrates the flow of data through the system, depicting how data is recorded, computed, transformed, and transmitted. The DFD shows the main processes involved in job posting management, user registration, job search, application submission, and reporting, highlighting data interactions and dependencies.

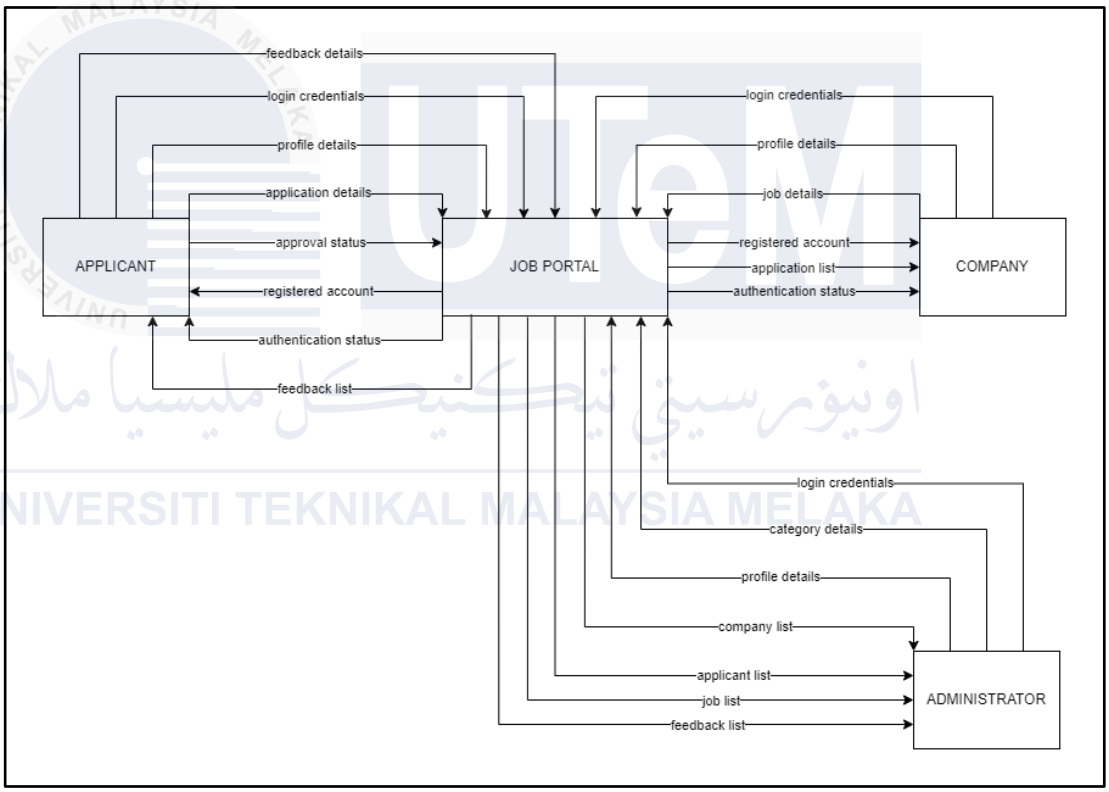


Figure 3.5: Context Diagram

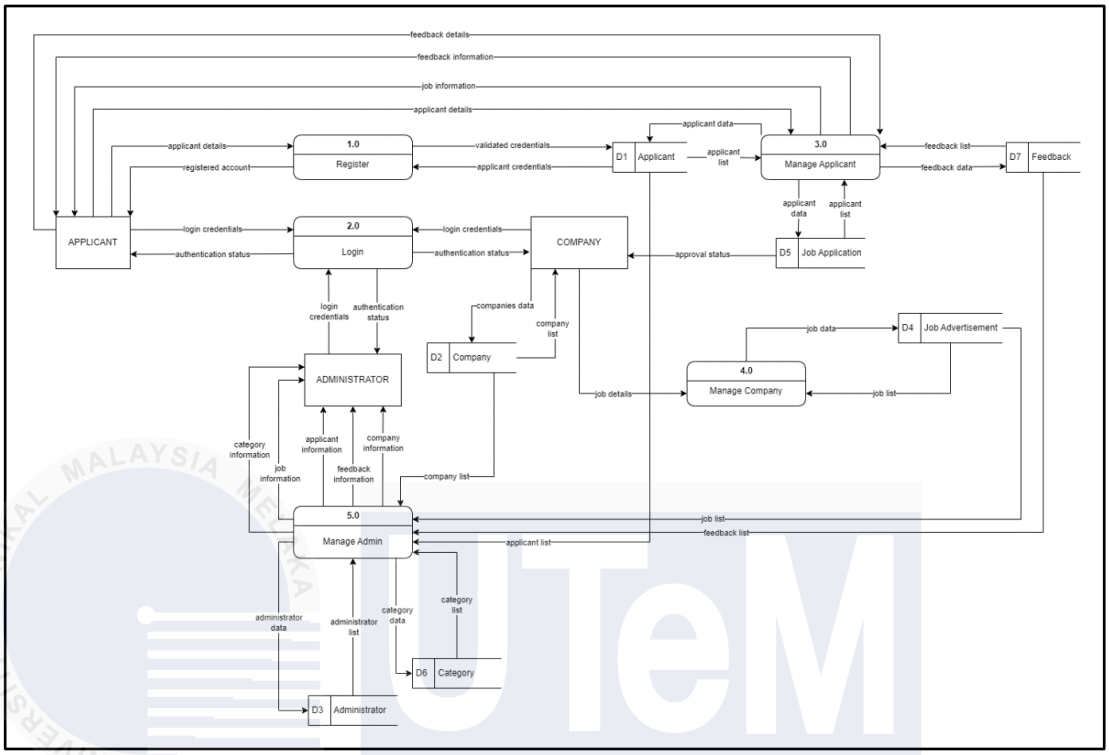


Figure 3.6: Data Flow Diagram (Level 0)

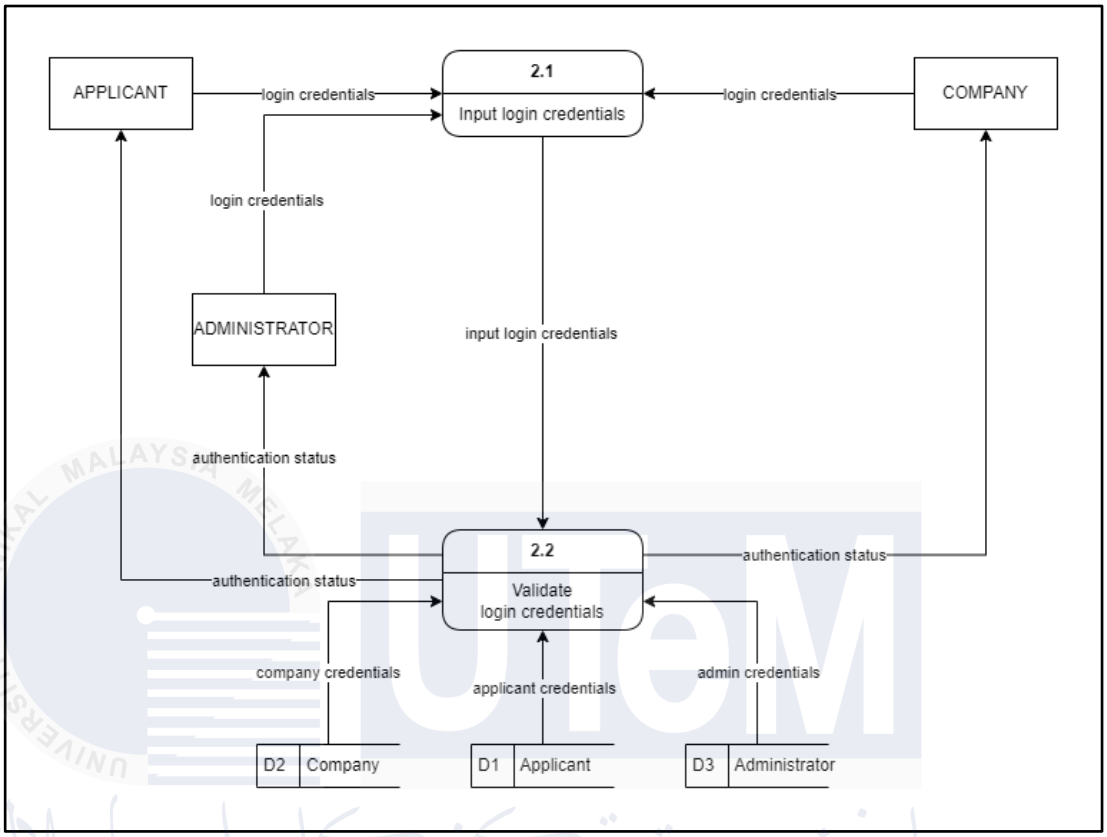


Figure 3.7: Data Flow Diagram (Level 1 - Login)

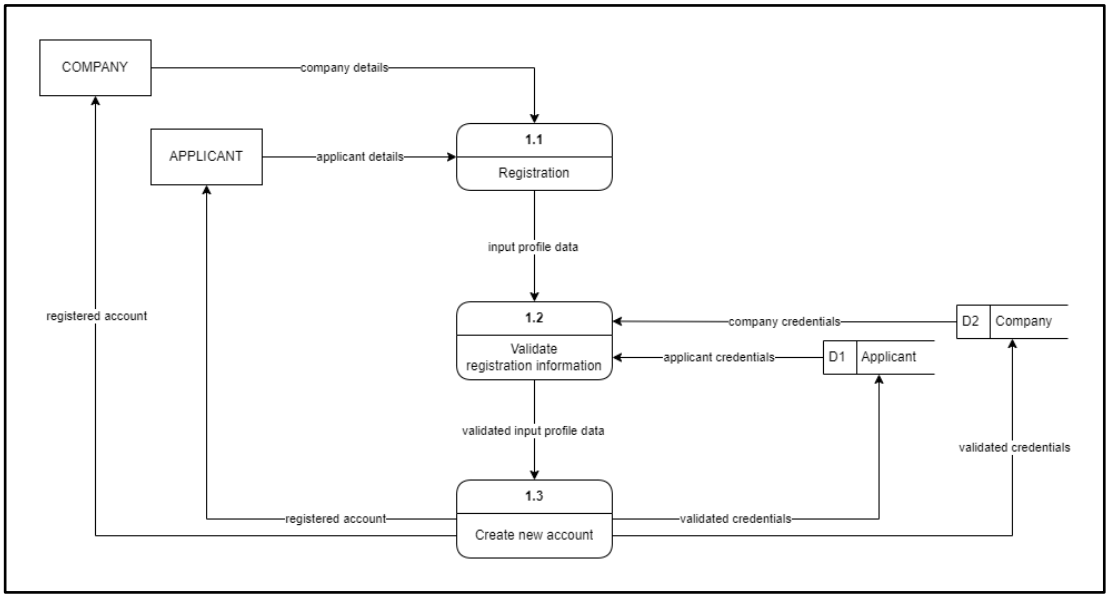


Figure 3.8: Data Flow Diagram (Level 1 - Register)

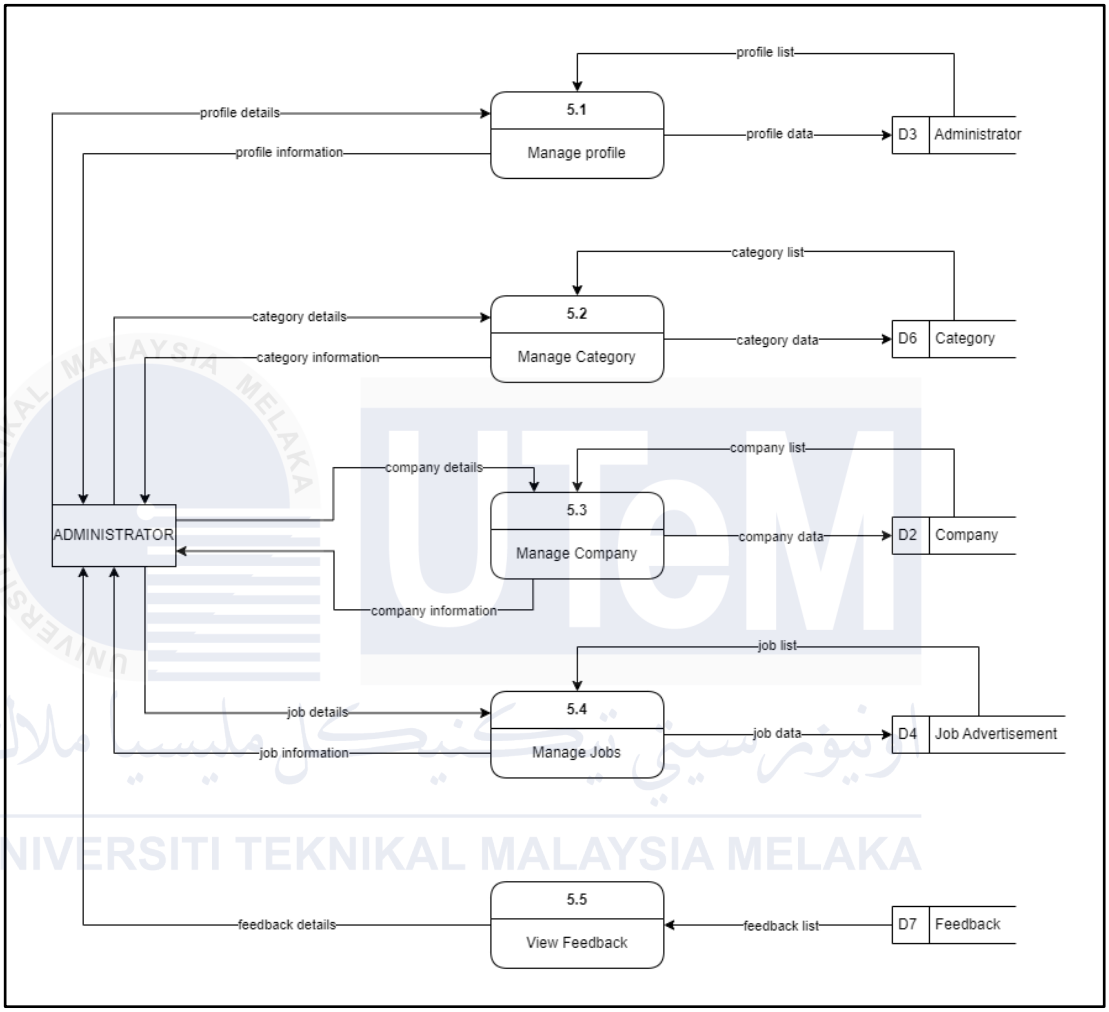


Figure 3.9: Data Flow Diagram (Level 1 - Administrator)

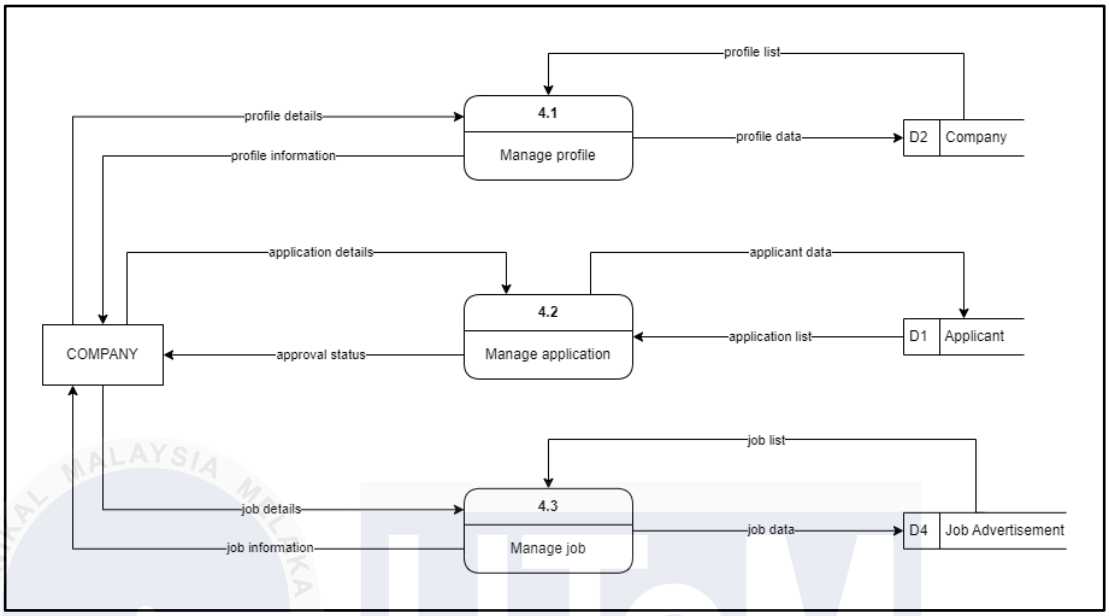


Figure 3.10: Data Flow Diagram (Level 1 - Company)

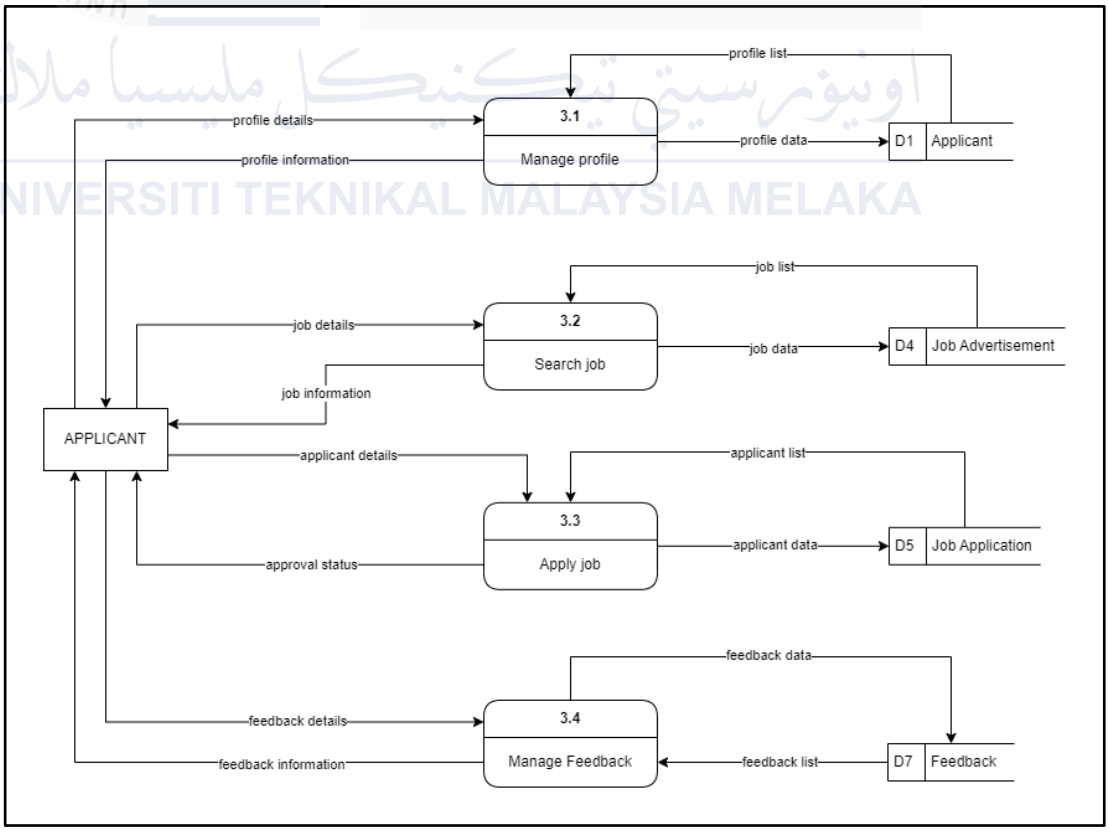


Figure 3.11: Data Flow Diagram (Level 1 - Applicant)

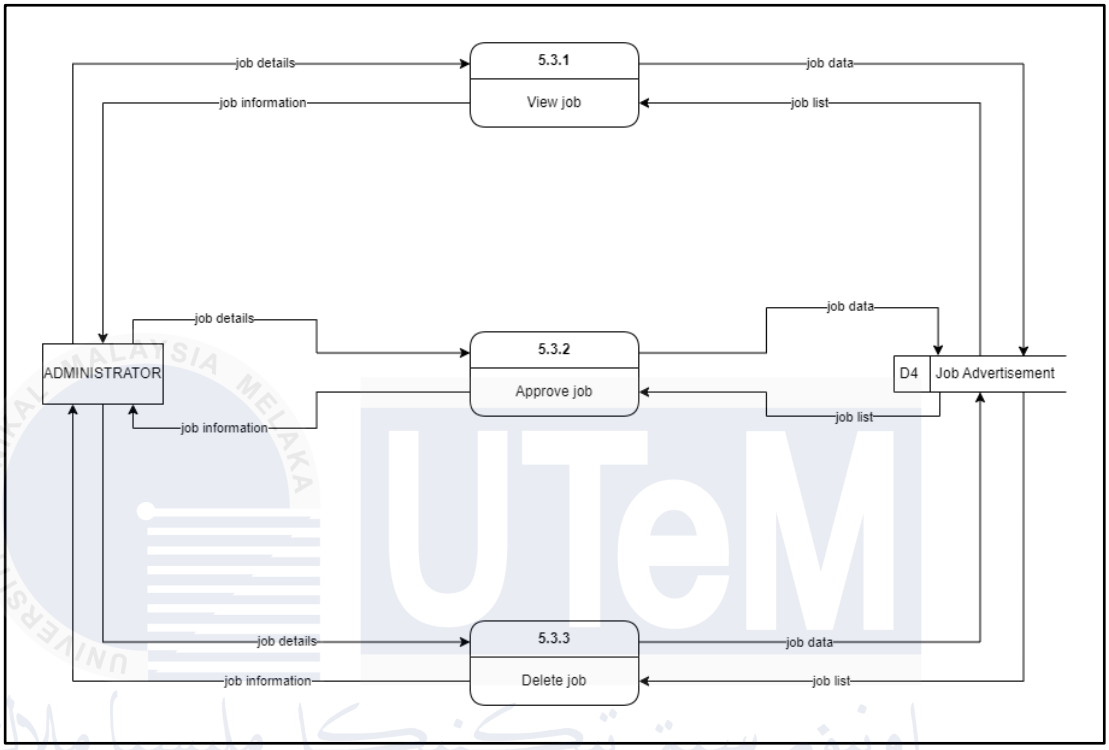


Figure 3.12: Data Flow Diagram (Level 2 Administrator – Manage Job)

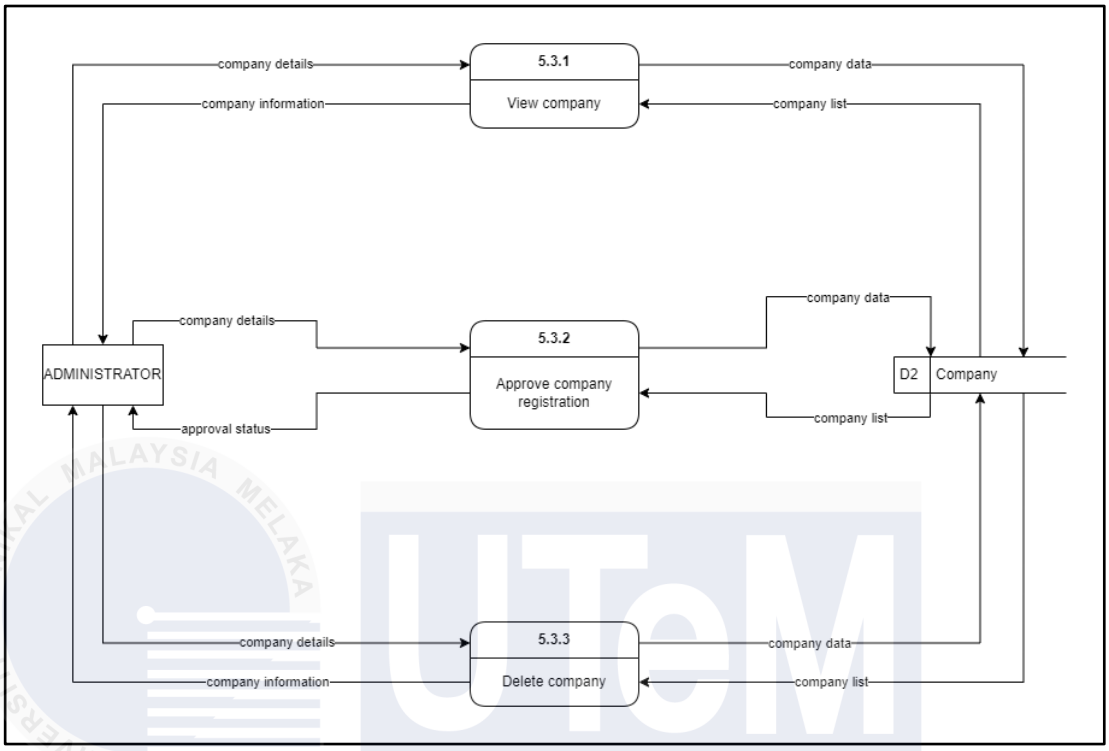


Figure 3.13: Data Flow Diagram (Level 2 Administrator – Manage Company)

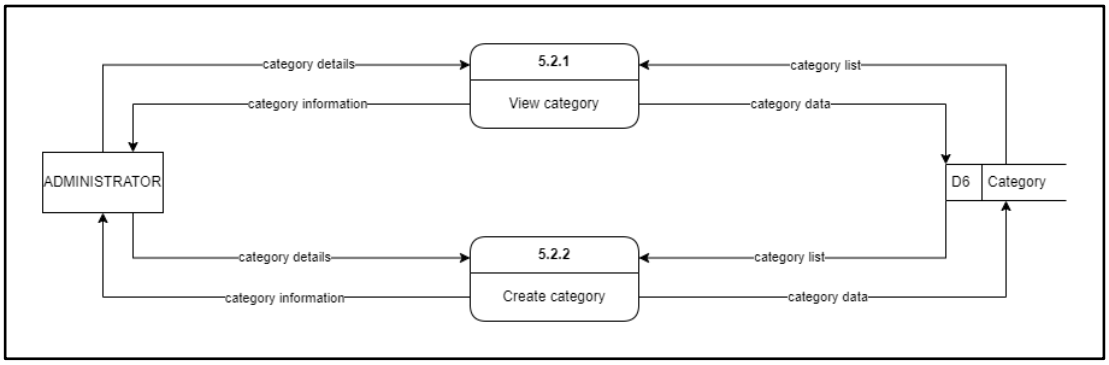


Figure 3.14: Data Flow Diagram (Level 2 Administrator – Manage Category)

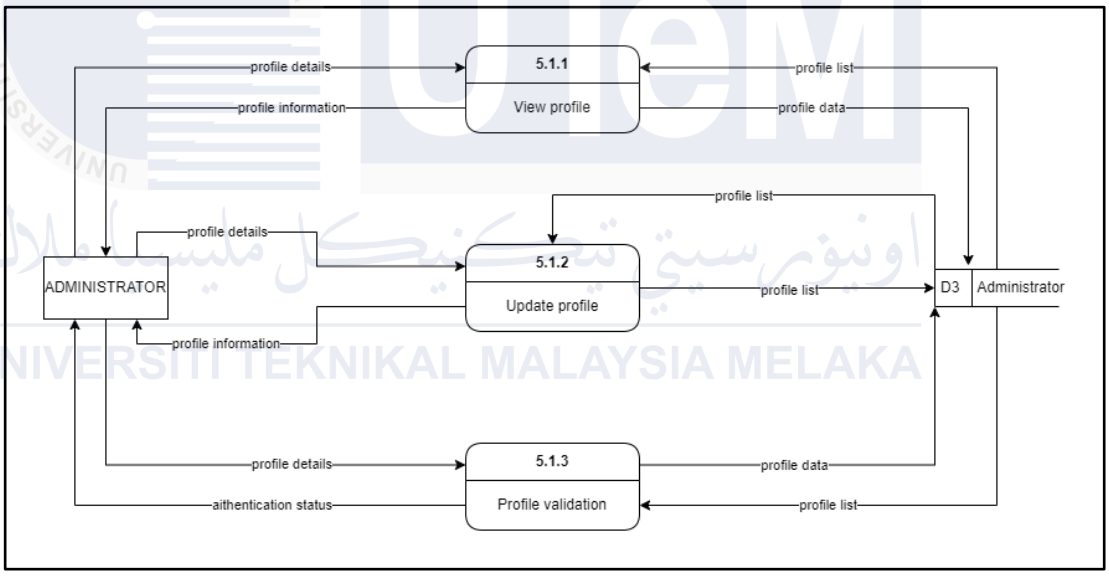


Figure 3.15: Data Flow Diagram (Level 2 Administrator – Manage Profile)

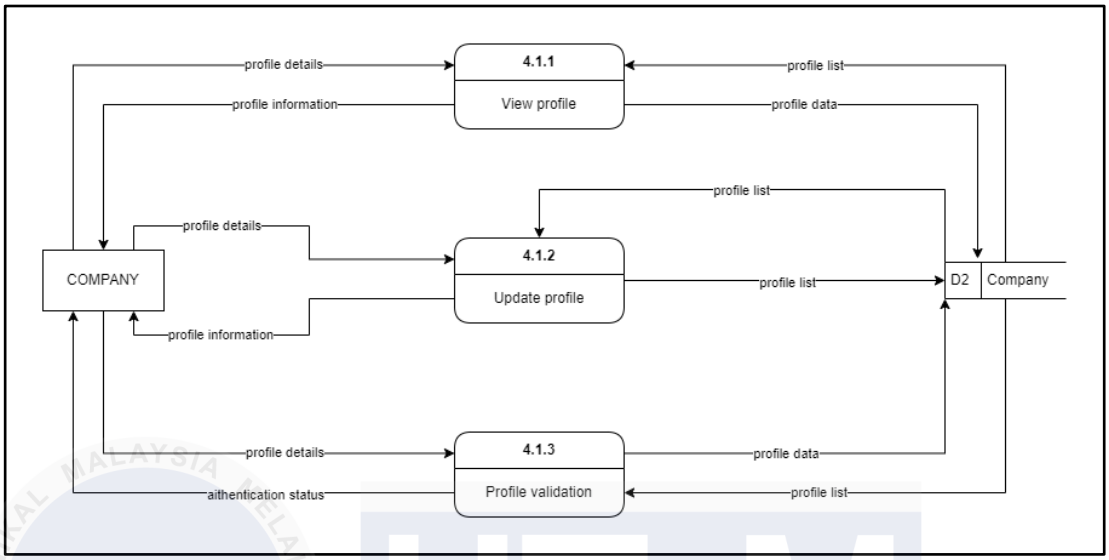


Figure 3.16: Data Flow Diagram (Level 2 Company – Manage Profile)

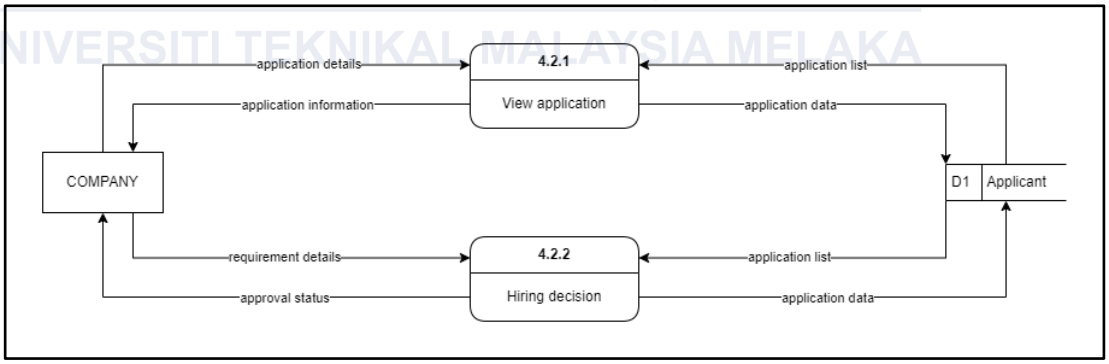


Figure 3.17: Data Flow Diagram (Level 2 Company – Manage Application)

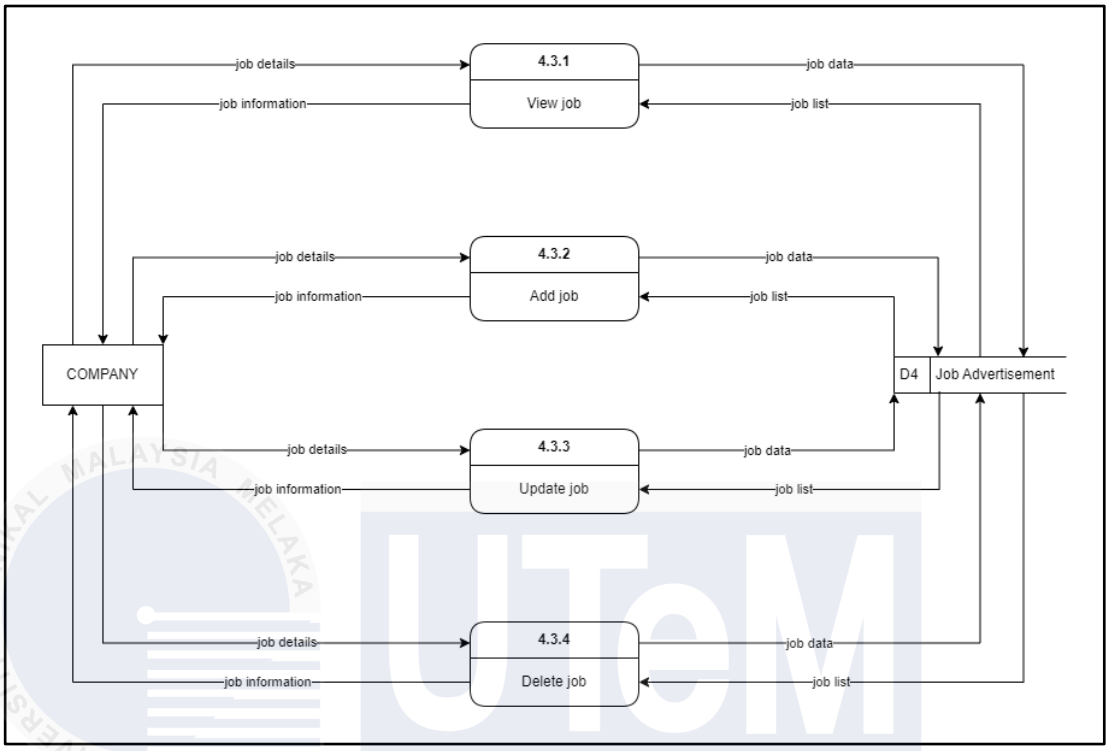


Figure 3.18: Data Flow Diagram (Level 2 Company – Manage Job)

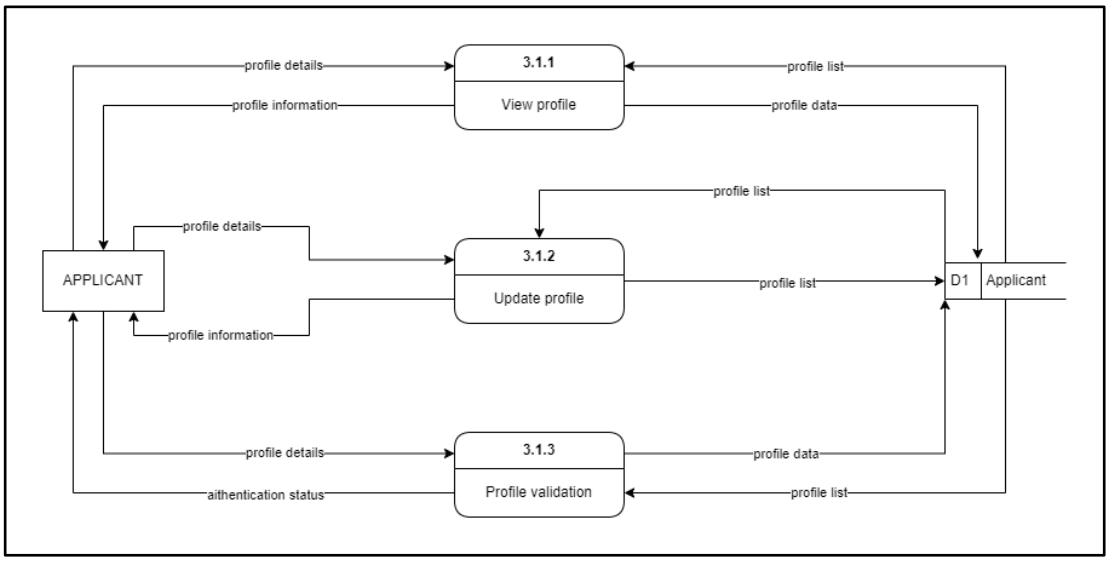


Figure 3.19: Data Flow Diagram (Level 2 Applicant – Manage Profile)

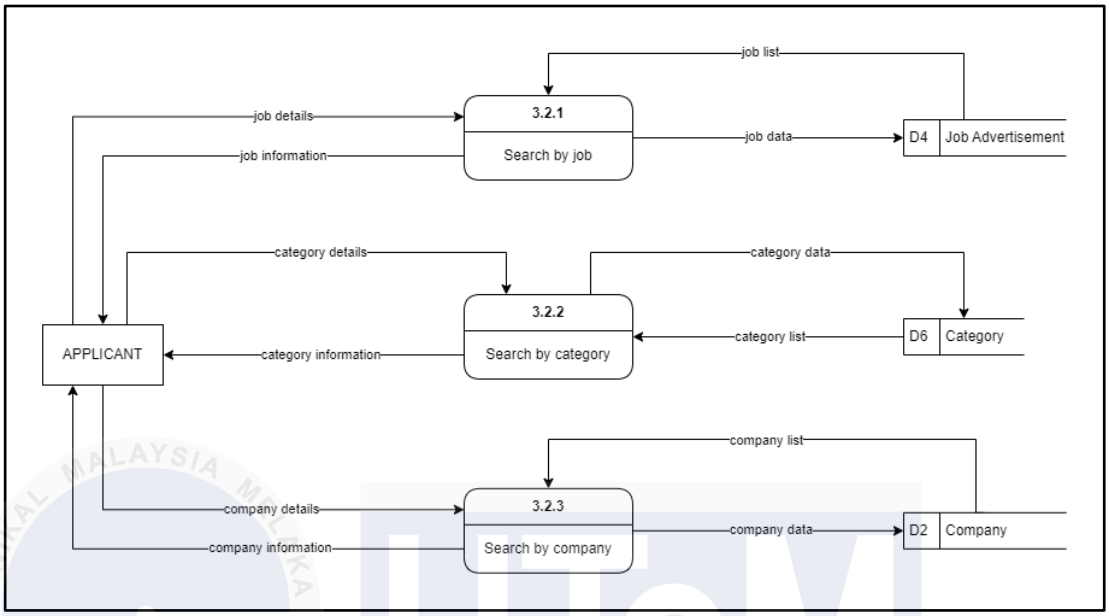


Figure 3.20: Data Flow Diagram (Level 2 Applicant – Search Job)

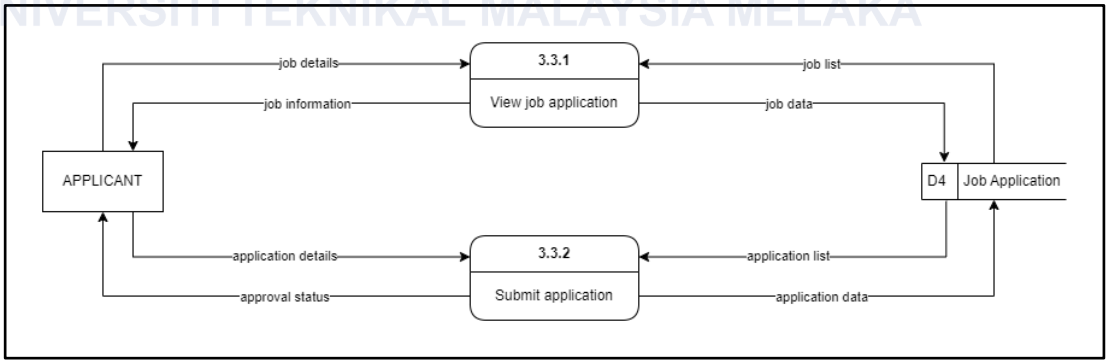


Figure 3.21: Data Flow Diagram (Level 2 Applicant – Apply Job)

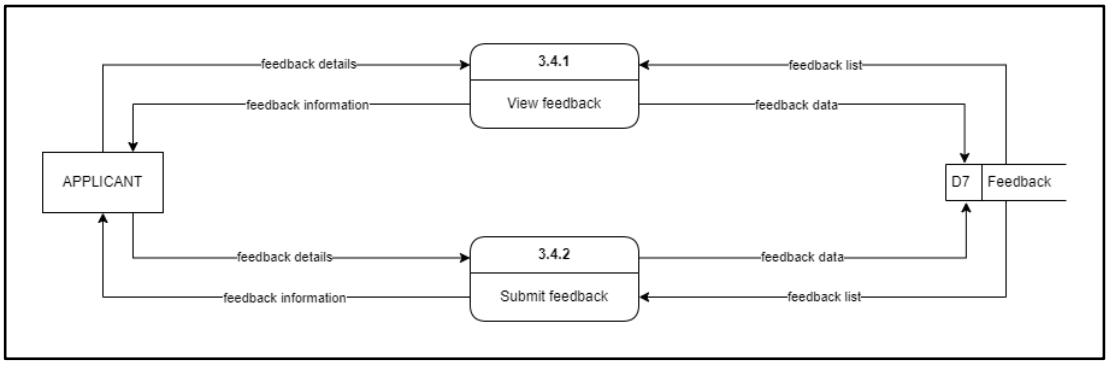


Figure 3.22: Data Flow Diagram (Level 2 Applicant – Manage Feedback)

3.4.2 Non-functional Requirement

The non-functional requirements specify how well the UTeM Job Portal system performs its intended functions. These requirements ensure that the system meets specific standards for performance, scalability, security, usability, and accessibility.

a) Performance

The UTeM Job Portal system is designed to ensure high responsiveness and efficiency, providing users with a seamless experience. The platform aims for quick page loads and minimal delays, with search results being returned within seconds and job application processes completed smoothly. This performance level is critical for maintaining user satisfaction and encouraging continuous engagement with the platform. The system's performance is regularly monitored and optimized to handle peak loads, ensuring consistent speed and reliability.

b) Scalability

Scalability is a core aspect of the UTeM Job Portal system, allowing it to adapt to an increasing number of users and data volume without compromising performance. The website is designed to handle growing demands efficiently, ensuring the platform remains functional and responsive even during high-traffic periods, such as university graduation seasons or major recruitment drives. This scalability ensures that the system can support future growth and accommodate more users and job listings over time.

c) Security

Security is paramount for the Job Portal system to protect sensitive user data, including personal details, resumes, job listings, and communications between job seekers and employers. The platform employs secure login protocols such as HTTPS, data encryption both at rest and in transit, and regular security audits to prevent unauthorized access and data breaches. Additionally, robust user authentication and access control mechanisms are in place to safeguard user privacy and maintain the integrity of the system.

d) Usability

The UTeM Job Portal system prioritizes usability to ensure that users with varying levels of technical expertise can navigate the platform effortlessly. The user interface is designed to be intuitive and straightforward, with clear menus and functionalities that make it easy for job seekers to search for jobs, submit applications, and manage their profiles. Employers can easily post job listings, review applications, and communicate with candidates. The platform's design minimizes the learning curve, enhancing the overall user experience.

3.4.3 Others Requirement

1. Software that will be used

a) Integrated Development Environment (IDE)

Visual Studio Code

Visual Studio Code (VS Code) is a powerful and versatile code editor that facilitates a streamlined development experience for web applications. VS Code offers features like syntax highlighting, code completion, and debugging capabilities to enhance developer productivity. Its large and active user community provides extensive documentation, forums, and support resources, ensuring developers have the tools and assistance they need.

b) Backend

phpMyAdmin

phpMyAdmin simplifies database management for the UTeM Job Portal. This user-friendly web interface provides intuitive features for users to create, modify, and interact with their MySQL database. Unlike complex programming, phpMyAdmin offers a visual interface, making it accessible to users with varying technical backgrounds.

c) Frontend

HTML

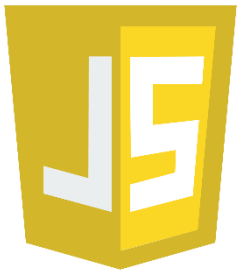
HTML (HyperText Markup Language) is the standard markup language for creating web pages and web applications. It provides the structure and content of a webpage by using a system of tags and elements. HTML defines the layout of text, images, links, and other multimedia elements, allowing browsers to render them appropriately. It serves as the backbone of web content, ensuring that information is presented in a coherent and accessible manner to users.

CSS

CSS (Cascading Style Sheets) is used for styling HTML pages to make them visually appealing and engaging. It allows developers to control the layout, colours, fonts, and overall appearance of web pages. By separating content (HTML) from presentation (CSS), developers can create consistent and attractive designs across multiple pages. CSS also supports responsive design, enabling web pages to adapt to different screen sizes and devices, providing a seamless user experience.

Bootstrap

Bootstrap is a popular front-end framework that significantly reduces development time by offering a collection of pre-designed UI components and templates. It provides a responsive grid system, along with various CSS and JavaScript components like navigation bars, buttons, forms, and modals. Developers can leverage these ready-to-use components to quickly build and prototype interfaces, ensuring a consistent and professional look across the application. Bootstrap's extensive documentation and community support make it an invaluable tool for front-end development.



JavaScript

JavaScript is a versatile scripting language used for client-side scripting and DOM (Document Object Model) manipulation. It enables developers to create dynamic and interactive web pages by responding to user actions, such as clicks, form submissions, and keyboard inputs. JavaScript can be used to update content, validate forms, create animations, and communicate with server-side scripts, enhancing the overall user experience. Its ability to interact with HTML and CSS makes it an essential component of modern web development.

d) Server-Side Scripting

PHP

PHP is used for server-side scripting to handle the back-end processes of the UTeM Job Portal. It allows for dynamic content generation and interaction with the database, making the system functional and responsive to user actions.

e) Database

MySQL

MySQL serves as the robust foundation for the UTeM Job Portal's database. This powerful relational database management system efficiently stores and manages all crucial data, from user information and job listings to communication details. MySQL ensures data integrity and facilitates efficient retrieval. The Job Portal leverages phpMyAdmin for seamless database management and interaction.

2. Hardware that will be used

Laptop

The development and operation of the UTeM Job Portal will primarily rely on laptop as the hardware of choice. Laptops offer portability and flexibility, allowing developers to work on the system from any location. For instance, the use of laptops enables developers to utilize databases, software, and programming tools effectively. Additionally, the lightweight and compact design of laptops enhances convenience and ease of use, ensuring that development tasks can be completed efficiently.

3. Other requirements
    a. Network Requirement

Wi-Fi and Mobile Network

The Job Portal system requires a stable Wi-Fi or mobile network connection to be functional. This ensures that users can access the platform, perform searches, submit applications, and manage job listings seamlessly, regardless of their location.

3.5 Conclusion

This chapter provided a comprehensive analysis of the current system, identified key problems, and proposed improvements to enhance the UTeM Job Portal. The requirement analysis detailed the functional and non-functional requirements of the to-be system, ensuring that it meets user needs and performs efficiently. The next chapter will focus on the design of the system, including database design and graphical user interface (GUI) design, to bring the proposed solutions to life. The analysis phase has set a clear foundation for the design and implementation phases, ensuring that the project objectives are aligned with user needs and expectations.

CHAPTER 4: DESIGN

4.1 Introduction

This chapter outlines the design phase of the UTeM Job Portal, detailing the system's architecture, database design, and graphical user interface (GUI) design. It provides a comprehensive overview of the design strategies employed to ensure the system meets the functional and non-functional requirements identified in the previous chapters. This chapter is divided into several sections, each focusing on a specific aspect of the design process, including the architectural view, database design, and GUI design.

4.2 Introductory preview to this chapter

The architecture of the UTeM Job Portal is designed using a multi-tier architecture to ensure scalability, maintainability, and security. The system is divided into three primary layers: the presentation layer, the application layer, and the data layer.

Presentation Layer:

- Purpose: This layer is responsible for the user interface and user experience. It includes all the web pages and forms that users interact with.
- Technologies Used: HTML, CSS, JavaScript, and Bootstrap.
- Components: Login page, registration forms, job search and application forms, dashboards for applicants and employers, and administrative interfaces.

Application Layer:

- Purpose: This layer contains the business logic and processes user requests, communicates between the presentation layer and the data layer, and ensures that business rules are enforced.
- Technologies Used: PHP for server-side scripting, Apache server for handling requests, and JavaScript for client-side interactions.
- Components: User authentication and authorization, job posting management, application processing, notifications, and reporting functionalities.

Data Layer:

- Purpose: This layer handles data storage, retrieval, and management. It ensures data integrity and security.
- Technologies Used: MySQL for database management.
- Components: Database schema, tables for storing user information, job postings, applications, feedback, and payment transactions.

4.3 Database Design

The database design phase involves creating a structured representation of the data required for the UTeM Job Portal. This phase is divided into conceptual design, logical design, and physical design.

4.3.1 Conceptual Design

The conceptual design phase involves developing an Entity Relationship Diagram (ERD) to represent the main entities and their relationships.

a) Entity Relationship Diagram (ERD)

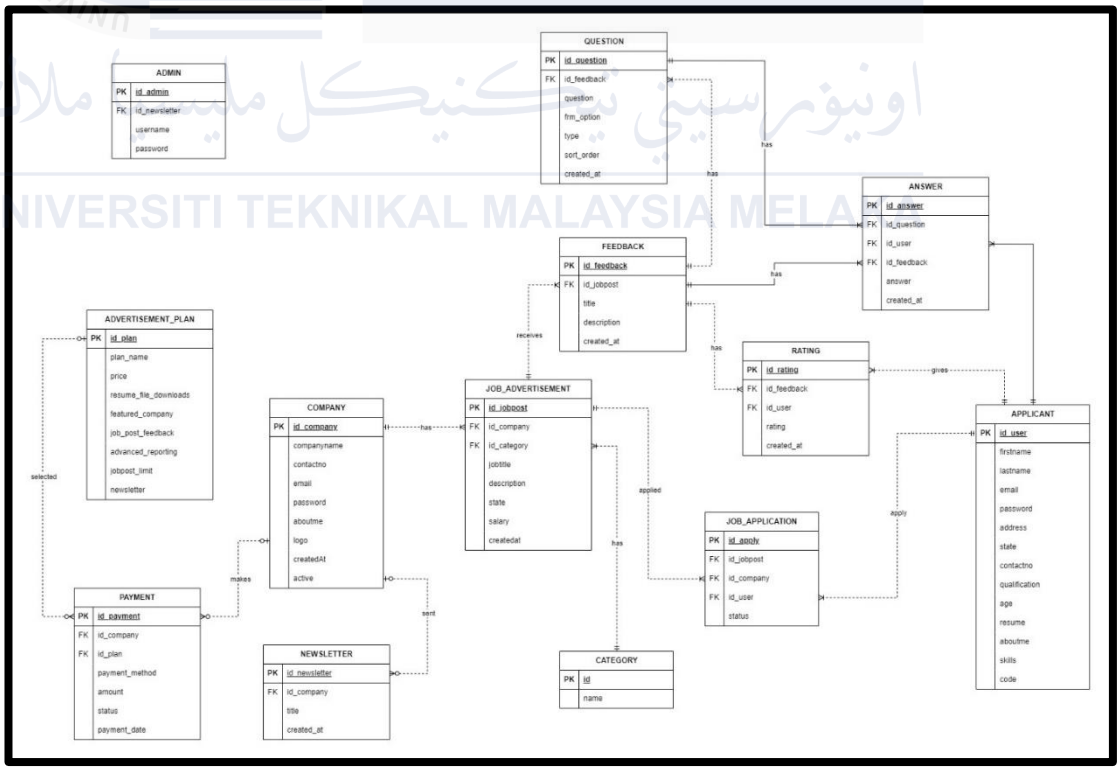


Figure 4.1: Entity Relationship Diagram

a) Business Rules

User Management

- Each applicant must register with a unique email address.
- Applicants can apply for multiple job advertisements, and each job advertisement can receive applications from multiple applicants.
- Only registered companies can post job advertisements and access applicant details.

Job Advertisement

- Each job advertisement is posted by one company and belongs to one category.
- A company can have multiple job advertisements in various categories.
- Job advertisements must include job title, description, location, and salary details.

Advertisement Plans

- Companies can choose from different advertisement plans, each with specific features and limitations.
- Advertisement plans dictate the number of job postings a company can make and additional features like resume file downloads and newsletter access.

Feedback and Rating

- Applicants can provide feedback for job advertisements they have applied to.
- Feedback is associated with specific job advertisements and can be rated by applicants.
- Each feedback entry must be linked to a job post and include a rating and detailed comments.

4.3.2 Logical Design

The logical design phase involves transforming the conceptual design into a detailed schema and ensuring data integrity through normalization.

a) Data Dictionary

Provides detailed descriptions of each database field, including data types, constraints, and relationships.

Table 4.1: `admin` table

Column Name	Data Type	Length Field	Data Format	Constraint	Description
id_admin	int	-	Integer	Primary key, auto-increment	Admin ID
username	varchar	255	Text	NOT NULL	Admin username
password	varchar	255	Text	NOT NULL	Admin password (hashed)

Table 4.2: `advertisement_plan` table

Column Name	Data Type	Length Field	Data Format	Constraint	Description
id_plan	int	-	Integer	Primary key, auto-increment	Plan ID
plan_name	varchar	50	Text	NOT NULL	Name of the advertisement plan
price	decimal	10,2	Decimal	NOT NULL	Price of the advertisement plan
resume_file_downloads	tinyint	1	Integer	NOT NULL DEFAULT 0	Indicates if resume downloads are allowed
featured_company	tinyint	1	Integer	NOT NULL DEFAULT 0	Indicates if company is featured
job_post_feedback	tinyint	1	Integer	NOT NULL DEFAULT 0	Indicates if job post feedback is allowed
advanced_reporting	tinyint	1	Integer	NOT NULL DEFAULT 0	Indicates if advanced reporting is allowed
jobpost_limit	int	-	Integer	NOT NULL	Limit of job posts allowed
newsletter	tinyint	1	Integer	NOT NULL DEFAULT 0	Indicates if newsletter is included

Table 4.3: `answer` table

Column Name	Data Type	Length Field	Data Format	Constraint	Description
id_answer	int	-	Integer	Primary key, auto-increment	Answer ID
id_question	int	-	Integer	NOT NULL	Foreign key to `question` table
id_user	int	-	Integer	NOT NULL	Foreign key to `applicant` table
id_feedback	int	-	Integer	NOT NULL	Foreign key to `feedback` table
answer	text	-	Text	NOT NULL	Answer text
created_at	timestamp	-	Timestamp	NOT NULL DEFAULT current_timestamp()	Timestamp of the answer creation

Table 4.4: `applicant` table

Column Name	Data Type	Length Field	Data Format	Constraint	Description
id_user	int	-	Integer	Primary key, auto-increment	Applicant ID
firstname	varchar	255	Text	NOT NULL	First name of the applicant
lastname	varchar	255	Text	NOT NULL	Last name of the applicant
email	varchar	255	Email	UNIQUE, NOT NULL	Unique email of the applicant
password	varchar	255	Text	NOT NULL	Password (hashed)
age	int	-	Integer		Age of the applicant
address	text	-	Text		Address of the applicant
state	varchar	255	Text		State of residence
contactno	varchar	255	Text		Contact number
qualification	varchar	255	Text		Qualification details
resume	varchar	255	Text		Path to resume file
aboutme	text	-	Text		About me description
skills	text	-	Text		Skills of the applicant
code	mediumint	50	Inte	NOT NULL	Verification or unique code

Table 4.5: `category` table

Column Name	Data Type	Length Field	Data Format	Constraint	Description
id_category	int	-	Integer	Primary key, auto-increment	Category ID
name	varchar	255	Text	NOT NULL	Name of the job category

Table 4.6: `company` table

Column Name	Data Type	Length Field	Data Format	Constraint	Description
id_company	int	-	Integer	Primary key, auto-increment	Company ID
companyname	varchar	255	Text	NOT NULL	Name of the company
contactno	varchar	255	Text	NOT NULL	Contact number of the company
email	varchar	255	Email	UNIQUE, NOT NULL	Unique email of the company
password	varchar	255	Text	NOT NULL	Password (hashed)
aboutme	varchar	255	Text		About the company description
logo	varchar	255	Text	NOT NULL	Path to company logo
id_plan	int	-	Integer	NOT NULL DEFAULT 1	Foreign key to `advertisement_plan`
createdAt	timestamp	-	Timestamp	NOT NULL DEFAULT current_timestamp()	Timestamp of company creation
active	int	-	Integ ↓	NOT NULL DEFAULT 2	Active status

Table 4.7: `feedback` table

Column Name	Data Type	Length Field	Data Format	Constraint	Description
id_feedback	int	-	Integer	Primary key, auto-increment	Feedback ID
id_jobpost	int	-	Integer	NOT NULL	Foreign key to `job_advertisement`
title	varchar	255	Text	NOT NULL	Title of the feedback
description	text	-	Text	NOT NULL	Description of the feedback
created_at	timestamp	-	Timestamp	NOT NULL DEFAULT current_timestamp()	Timestamp of feedback creation

Table 4.8: `job_advertisement` table

Column Name	Data Type	Length Field	Data Format	Constraint	Description
id_jobpost	int	-	Integer	Primary key, auto-increment	Job Post ID
id_company	int	-	Integer	NOT NULL	Foreign key to `company`
id_category	int	-	Integer	NOT NULL	Foreign key to `category`
jobtitle	varchar	255	Text	NOT NULL	Title of the job
description	text	-	Text	NOT NULL	Description of the job
state	varchar	255	Text	NOT NULL	State where the job is located
salary	decimal	10,2	Decimal		Salary offered for the job
createdat	timestamp	-	Timestamp	NOT NULL DEFAULT current_timestamp()	Timestamp of job post creation

Table 4.9: `job_application` table

Column Name	Data Type	Length Field	Data Format	Constraint	Description
id_apply	int	-	Integer	Primary key, auto-increment	Application ID
id_jobpost	int	-	Integer	NOT NULL	Foreign key to `job_advertisement`
id_company	int	-	Integer	NOT NULL	Foreign key to `company`
id_user	int	-	Integer	NOT NULL	Foreign key to `applicant`
status	int	-	Integer	NOT NULL DEFAULT 0	Status of the job application

Table 4.10: `mailbox` table

Column Name	Data Type	Length Field	Data Format	Constraint	Description
id_mailbox	int	-	Integer	Primary key, auto-increment	Mailbox ID
id_fromuser	int				

b) Query Design

i. Aggregate Queries

- Number of Job Applications per Advertisement:

```
SELECT
     job_advertisement.jobtitle,
     COUNT(job_application.id_apply)AS
     application_count
FROM
     job_advertisement
LEFT JOIN
     job_application ON
     job_advertisement.id_jobpost =
     job_application.id_jobpost
     GROUP BY
    job_advertisement.jobtitle;
```

- Average Ratings for Feedback:

```
SELECT
      feedback.title,
      AVG(rating.rating) AS
      average_rating
FROM
      feedback
LEFT JOIN
      rating ON feedback.id_feedback =
      rating.id_feedback
      GROUP BY feedback.title;
```

- Number of Applicants per State:

```
SELECT
      applicant.state,
      COUNT(applicant.id_user) AS applicant_count
FROM
      applicant
      GROUP BY applicant.state;
```

ii. Subqueries

- Detailed Information about Applicants who Applied for a Specific Job:

```
SELECT
     applicant.*
FROM
     applicant
WHERE
     applicant.id_user IN (
     SELECT job_application.id_user
     FROM job_application
     WHERE job_application.id_jobpost =
     1
);
```

- Job Applications with Applicants' Names:

```
SELECT
     job_application.id_apply,
     job_advertisement.jobtitle,
     CONCAT(applicant.firstname, ' ',
     applicant.lastname) AS
     applicant_name,
     job_application.status
FROM
     job_application
JOIN
     job_advertisement ON
     job_application.id_jobpost =
     job_advertisement.id_jobpost
JOIN
applicant ON job_application.id_user =
applicant.id_user;
```

- Applicants Who Applied for a Job in a Specific Company:

```
SELECT
     applicant.*
FROM
     applicant
WHERE
     applicant.id_user IN (
     SELECT
     job_application.id_user
     FROM
     job_application
     JOIN
     company ON
     job_application.id_company =
     company.id_company
     WHERE
     company.companyname = 'IKEA'
);
```

iii. Join Queries:

- Linking Applicants and Job Applications to Generate Comprehensive Reports:

```
SELECT
     applicant.firstname,
     applicant.lastname,
     job_advertisement.jobtitle,
     job_application.status
FROM
     job_application
JOIN
     applicant ON
     job_application.id_user =
     applicant.id_user
JOIN
job_advertisement ON
job_application.id_jobpost =
job_advertisement.id_jobpost;
```

- Linking Companies and Job Advertisements:

```
SELECT
     company.companyname,
     job_advertisement.jobtitle,
     job_advertisement.state,
     job_advertisement.salary
FROM
     job_advertisement
JOIN
company ON job_advertisement.id_company
= company.id_company;
```

- Detailed Job Application Report:

```
SELECT
     job_application.id_apply,
     applicant.firstname,
     applicant.lastname,
     job_advertisement.jobtitle,
     company.companyname,
     job_application.status
FROM
     job_application
JOIN
     applicant ON
     job_application.id_user =
     applicant.id_user
JOIN
     job_advertisement ON
     job_application.id_jobpost =
     job_advertisement.id_jobpost
JOIN
company ON job_application.id_company =
company.id_company;
```

4.3.3 Physical Design

The physical design phase involves implementing the logical schema on the chosen DBMS and configuring database objects to enhance performance and security.

b) DBMS Selection:

MySQL:

Chosen for its reliability, performance, and support for complex queries and transactions.

c) Database Objects:

- Stored Procedure:

1. Procedure to Add a New Job Advertisement:

```
DELIMITER //

CREATE PROCEDURE AddJobAdvertisement(

     IN company_id INT,
     IN category_id INT,
     IN job_title VARCHAR(255),
     IN job_description TEXT,
     IN job_state VARCHAR(255),
     IN job_salary DECIMAL(10,2)
)
BEGIN
     INSERT INTO job_advertisement (id_company,
     id_category, jobtitle, description, state,
     salary)

     VALUES (company_id, category_id, job_title,
     job_description, job_state, job_salary);
     END //
DELIMITER ;
```

2. Procedure to Update Applicant Information:

```
DELIMITER //

CREATE PROCEDURE UpdateApplicant(

     IN user_id INT,
     IN new_firstname VARCHAR(255),
     IN new_lastname VARCHAR(255),
     IN new_email VARCHAR(255),
     IN new_password VARCHAR(255),
     IN new_age INT,
     IN new_address TEXT,
     IN new_state VARCHAR(255),
     IN new_contactno VARCHAR(255),
     IN new_qualification VARCHAR(255),
     IN new_resume VARCHAR(255),
     IN new_aboutme TEXT,
     IN new_skills TEXT,
     IN new_code MEDIUMINT
)

BEGIN

     UPDATE applicant
     SET firstname = new_firstname, lastname =
     new_lastname, email = new_email, password =
     new_password, age = new_age, address =
     new_address, state = new_state,
     contactno = new_contactno,
     qualification = new_qualification,
     resume = new_resume, aboutme = new_aboutme,
     skills = new_skills,
     code = new_code
     WHERE id_user = user_id; END //
DELIMITER ;
```

3. Procedure to Delete a Company by ID:

```
DELIMITER //

CREATE PROCEDURE DeleteCompany(IN company_id INT)

BEGIN

    DELETE FROM company WHERE id_company =
company_id;

END //

DELIMITER ;
```

- Triggers:

i. Trigger to Automatically Update createdat Timestamp on Job Advertisement Update:


```
DELIMITER //

     CREATE TRIGGER
     UpdateJobAdvertisementTimestamp
     BEFORE UPDATE ON job_advertisement
     FOR EACH ROW
     BEGIN
         SET NEW.createdat = CURRENT_TIMESTAMP;
     END //
DELIMITER ;
```

ii. Trigger to Log Deletions in job_application Table:

```
DELIMITER //

     CREATE TRIGGER LogJobApplicationDeletion
     AFTER DELETE ON job_application
     FOR EACH ROW
     BEGIN
         INSERT INTO deleted_job_applications
     (id_apply, id_jobpost, id_company, id_user,
     status, deleted_at)
         VALUES (OLD.id_apply, OLD.id_jobpost,
     OLD.id_company, OLD.id_user, OLD.status,
     CURRENT_TIMESTAMP);
     END //
DELIMITER ;
```

iii. Trigger to Ensure Email Uniqueness in applicant Table:

```
DELIMITER //
     CREATE TRIGGER EnsureUniqueEmail
     BEFORE INSERT ON applicant
     FOR EACH ROW
     BEGIN
         DECLARE email_count INT;
         SET email_count = (SELECT COUNT(*) FROM
     applicant WHERE email = NEW.email);
         IF email_count > 0 THEN
             SIGNAL SQLSTATE '45000' SET
     MESSAGE_TEXT = 'Duplicate email detected';
         END IF;
     END //
DELIMITER ;
```

d) Security Mechanisms:

- Privileges:

The Job Portal website implements a role-based access control (RBAC) system to ensure that only authorized users can perform specific actions. The system defines three primary roles: Admin, Company, and Applicant. Each role has specific privileges associated with it, ensuring users have access to only the necessary features and data.

Admin Privileges:

- Manage Companies: Add, edit, delete company records.
- Manage Job Vacancies: Post, edit, delete job advertisements.
- Manage Employees: View and manage employer profiles.
- Manage Applicants: View and manage job seeker profiles.
- Manage Categories: Add, edit, delete job categories.
- Manage Users: Add, edit, delete users across all roles.

Company Privileges:

- Post Job Vacancies: Create and manage job advertisements.
- Manage Company Profile: Update company information and upload logo.
- View Applicants: Review applications received for job postings.
- Communication: Send messages to job seekers and receive their responses.

Applicant Privileges:

- Search Jobs: Browse and search for job vacancies.
- Apply for Jobs: Submit applications to available job postings.
- Manage Profile: Update personal information, upload resumes, and manage skills.
- Communication: Send messages to employers and receive their responses.

- Passwords:

  Passwords are stored securely using bcrypt hashing algorithm to ensure that even if the database is compromised, the passwords remain protected. The bcrypt algorithm is chosen for its strength and ability to add a salt to each password, making it resistant to rainbow table attacks.

```
// Hashing a password
$password = 'user_password';
$hashed_password = password_hash($password,
PASSWORD_BCRYPT);

// Verifying a password
if (password_verify($password, $hashed_password))
{
    // Password is correct
} else {
    // Password is incorrect
}
```

- User-level Security:

  User-level security is enforced through secure registration and authentication processes, ensuring that only authenticated users can access the system.

Secure Registration:

- Users must provide a valid email address, which is verified during registration.
- Passwords must meet complexity requirements (e.g., minimum length, inclusion of special characters).

```
// Registration example

$email = $_POST['email'];

$password = $_POST['password'];


// Check if email already exists

$query = "SELECT * FROM users WHERE email = ?";

$stmt = $conn->prepare($query);

$stmt->bind_param('s', $email); $stmt->execute();

$result = $stmt->get_result();

if ($result->num_rows > 0) { // Email already
exists

} else {

    // Hash the password and insert new user

    $hashed_password = password_hash($password,
PASSWORD_BCRYPT);

    $query = "INSERT INTO users (email, password)
VALUES (?, ?)";

    $stmt = $conn->prepare($query);

    $stmt->bind_param('ss', $email,
$hashed_password); $stmt->execute();}
```

Secure Login:

- Users must authenticate using their email and password.
- Sessions are used to maintain user authentication state securely.

```
// Login example

$email = $_POST['email'];

$password = $_POST['password'];


// Retrieve the user's hashed password from the
database

$query = "SELECT * FROM users WHERE email = ?";

$stmt = $conn->prepare($query);

$stmt->bind_param('s', $email); $stmt->execute();

$result = $stmt->get_result();

if ($result->num_rows == 1) {

    $user = $result->fetch_assoc();

    if (password_verify($password, $user['password']))
{

        // Password is correct, start a session

        session_start();

        $_SESSION['user_id'] = $user['id'];

        $_SESSION['user_role'] = $user['role'];

        // Redirect to dashboard } else {

        // Password is incorrect }

} else { // User not found }
```

Session Management:

- Sessions are used to track logged-in users securely.
- Session hijacking prevention mechanisms, such as regenerating session IDs, are implemented.

```
// Starting a session
session_start();


// Regenerate session ID to prevent session hijacking
session_regenerate_id(true);


// Store user information in session variables
$_SESSION['user_id'] = $user_id;
$_SESSION['user_role'] = $user_role;

// Checking if user is logged in
if (!isset($_SESSION['user_id'])) {
    // User is not logged in, redirect to login page
    header('Location: login.php');
    exit();
}
```

4.4 Graphical User Interface (GUI) Design

The GUI design focuses on providing an intuitive and user-friendly interface for all users, including applicants, employers, and administrators.

Navigation Flow:

- Applicants:

  Register/Login >>> Search Jobs >>> View Job Details >>> Apply for Jobs >>> View Application Status >>> Give Feedback.

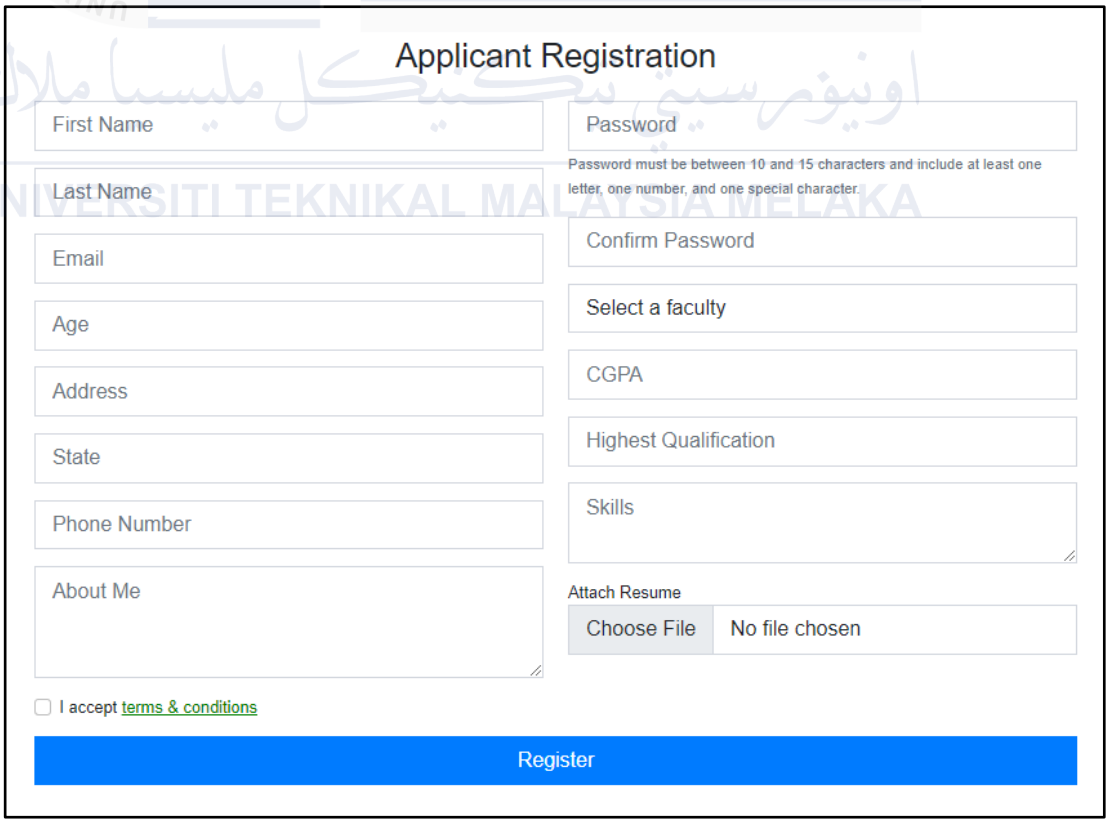


Figure 4.2: Register (Applicant)

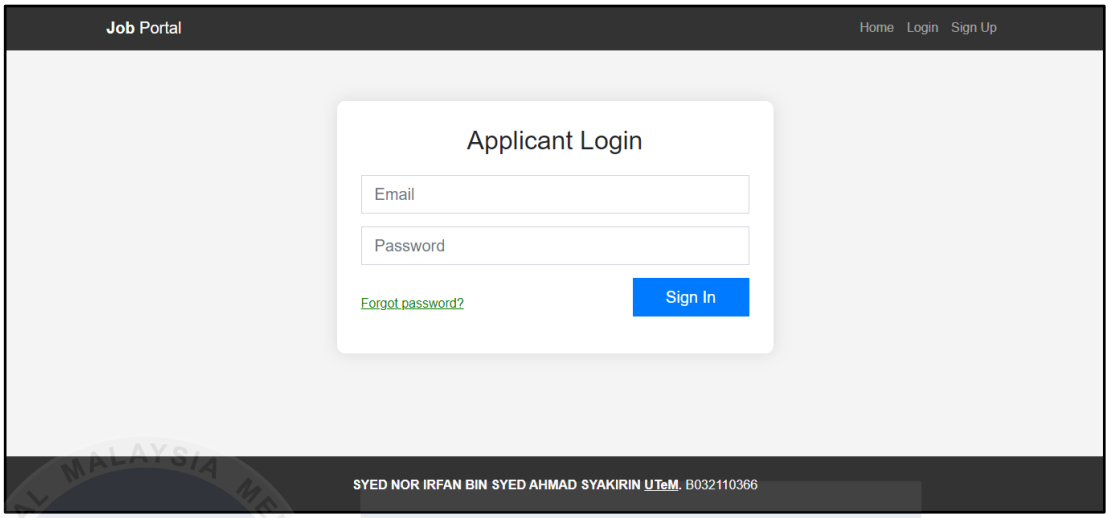


Figure 4.3: Login (Applicant)

Figure 4.4: Job Listing (Applicant)

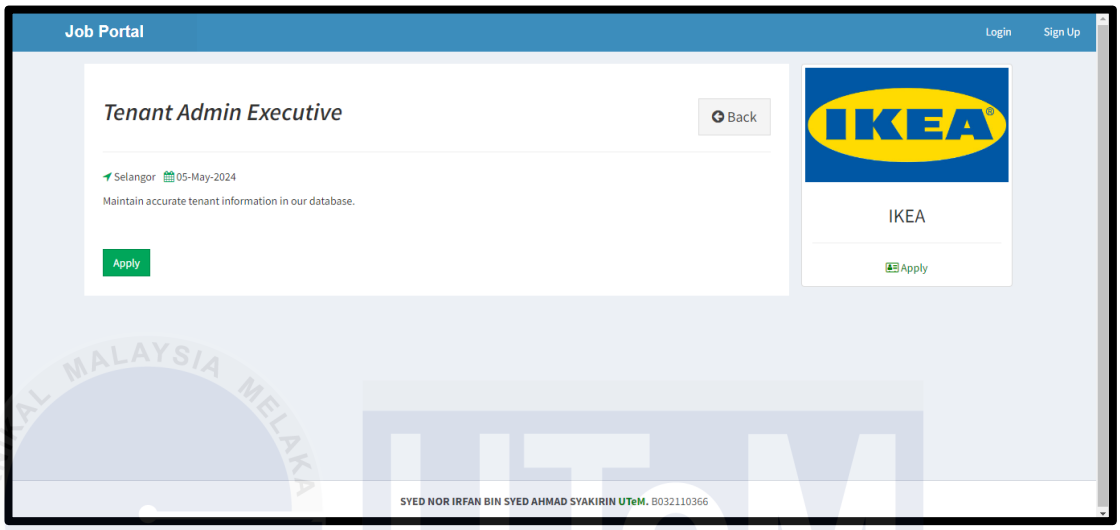


Figure 4.5: Job Details (Applicant)

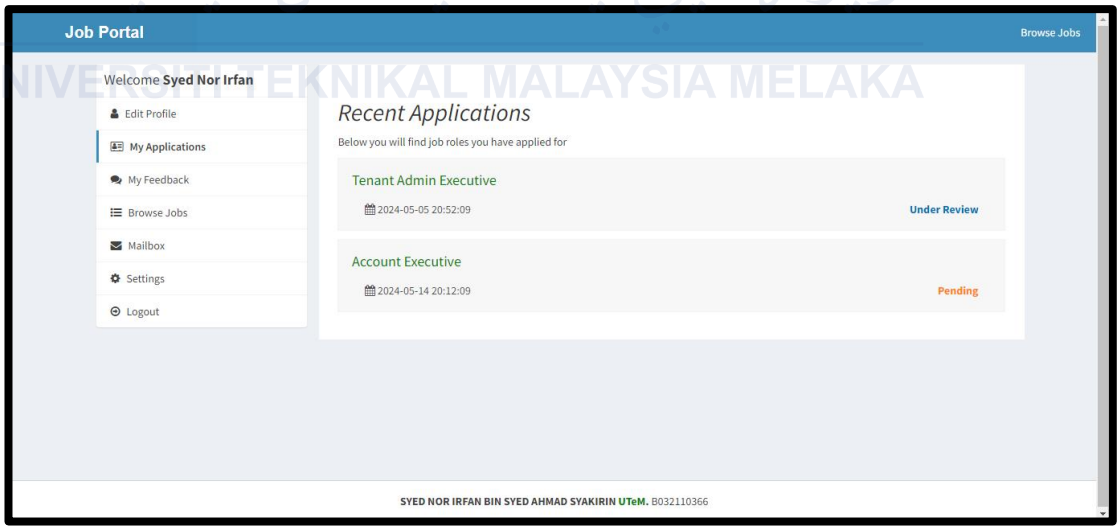


Figure 4.6: Job Application (Applicant)

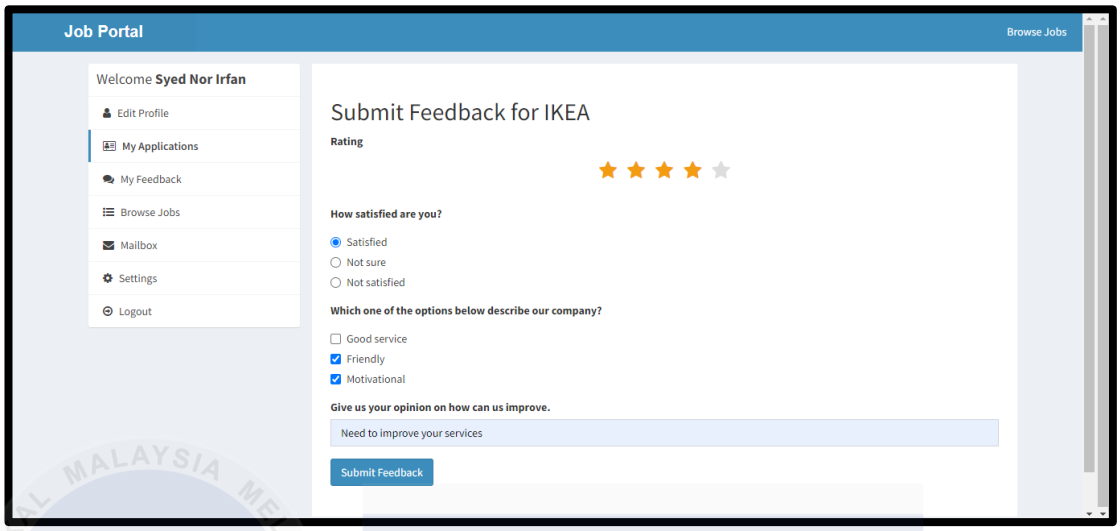


Figure 4.7: Feedback (Applicant)

- Companies:

  Register/Login >>> Post Jobs >>> Manage Job Listings >>> Review Applications >>> Generate Reports.

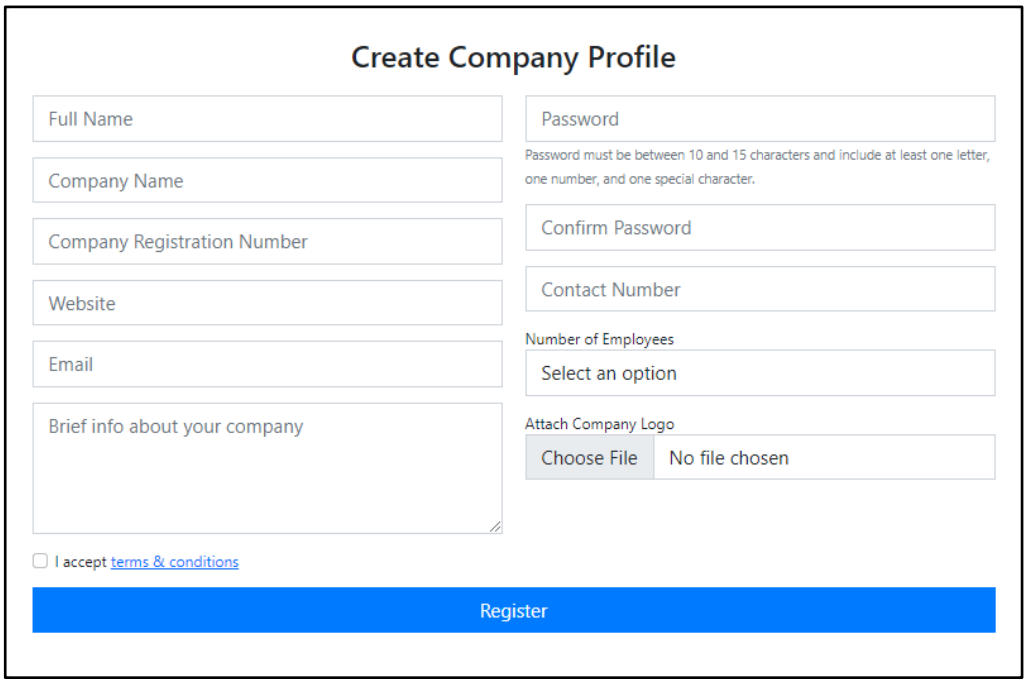


Figure 4.8: Register (Company)

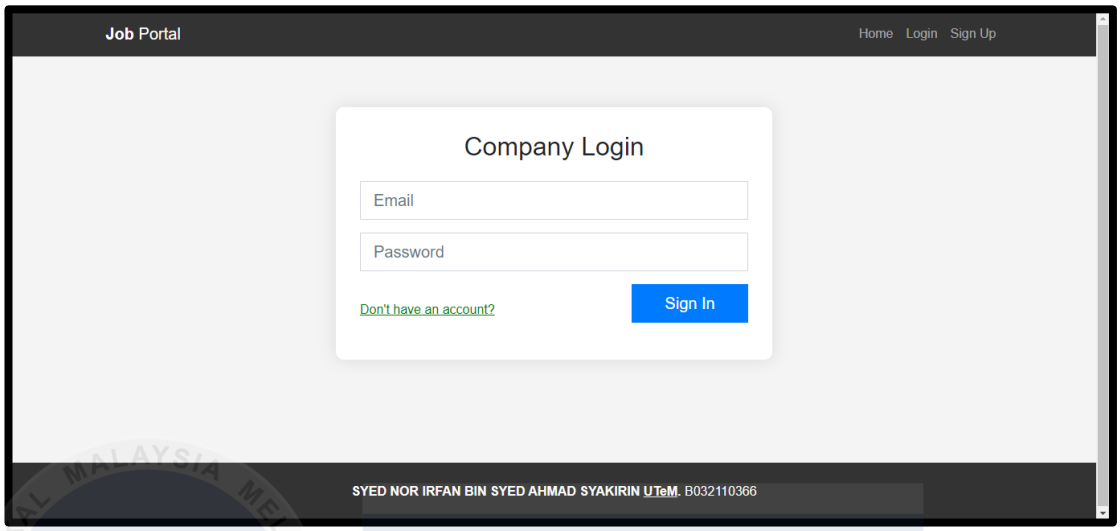


Figure 4.9: Login (Company)

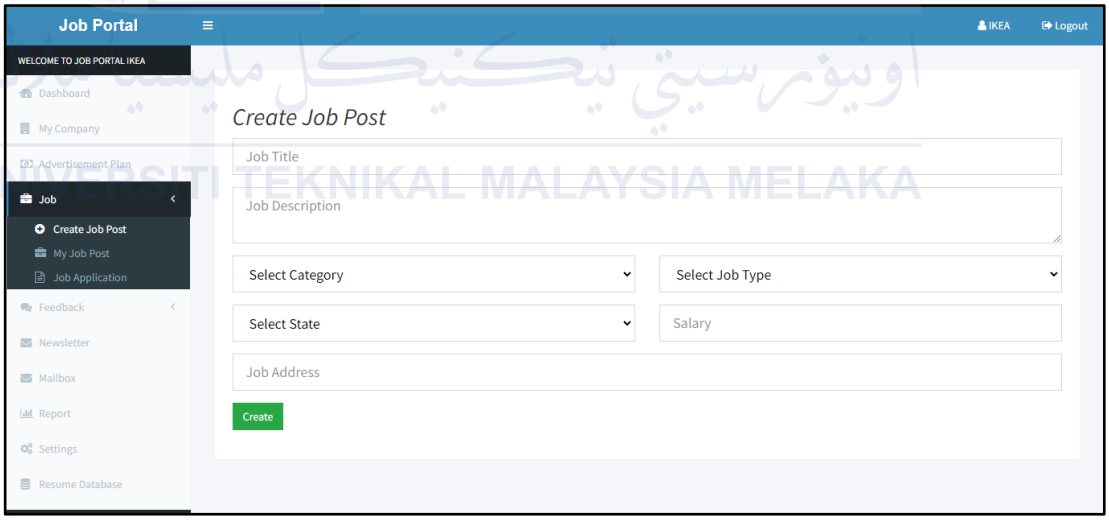


Figure 4.10: Create Job (Company)

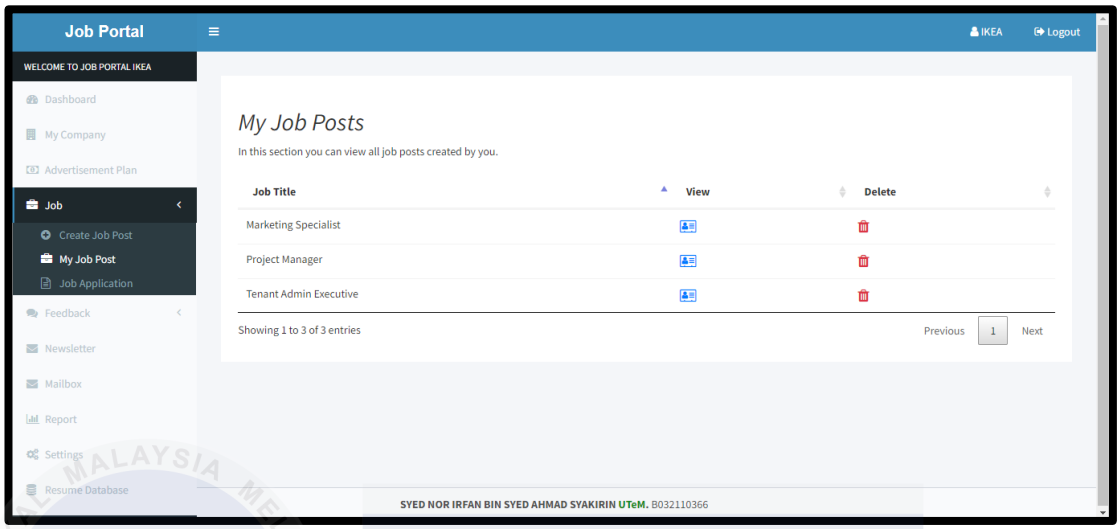


Figure 4.11: Manage Job (Company)

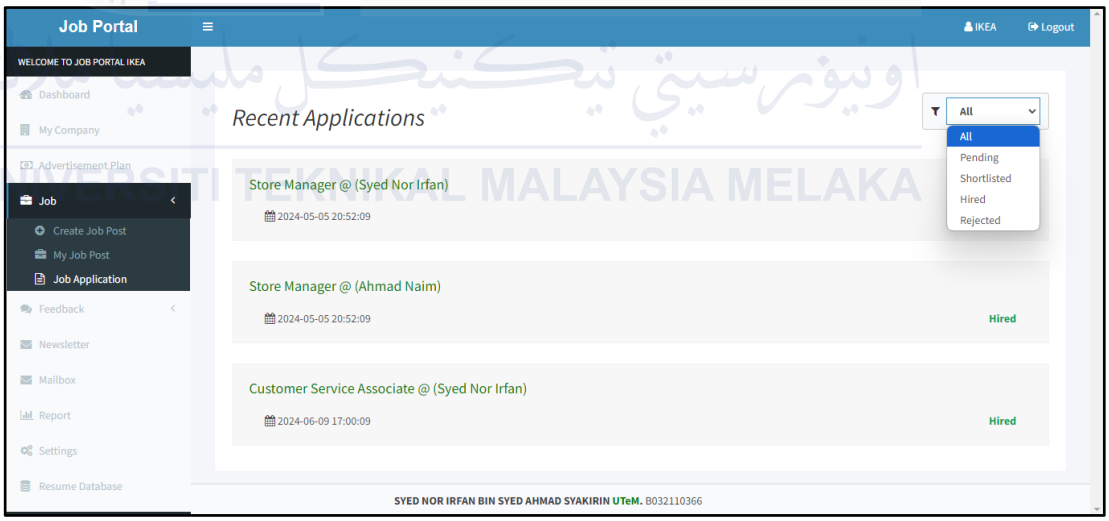


Figure 4.12: Job Application (Company)

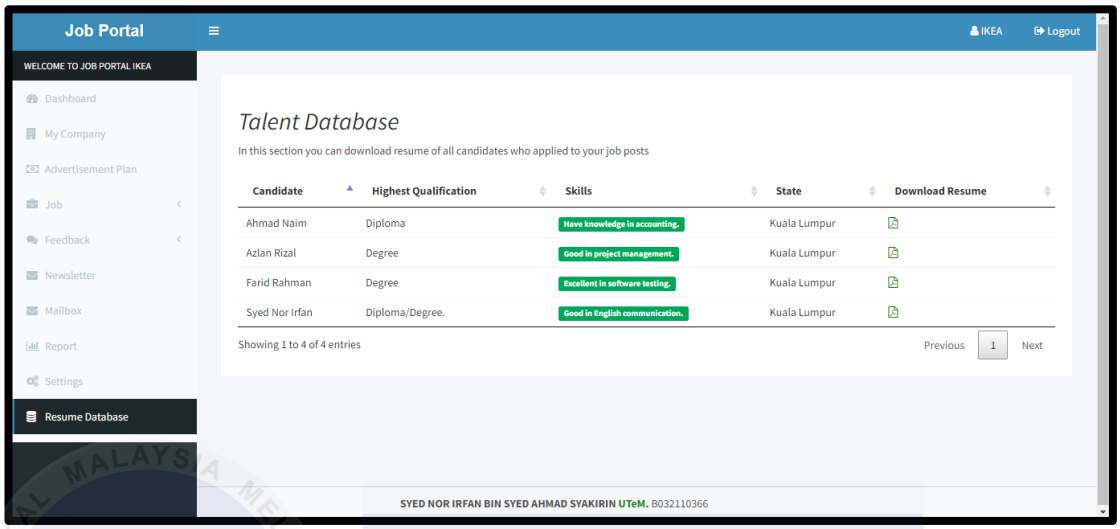


Figure 4.13: Resume Database (Company)

- Administrators:

Login >>> Manage Users >>> Manage Job Listings >>> View Analytics.

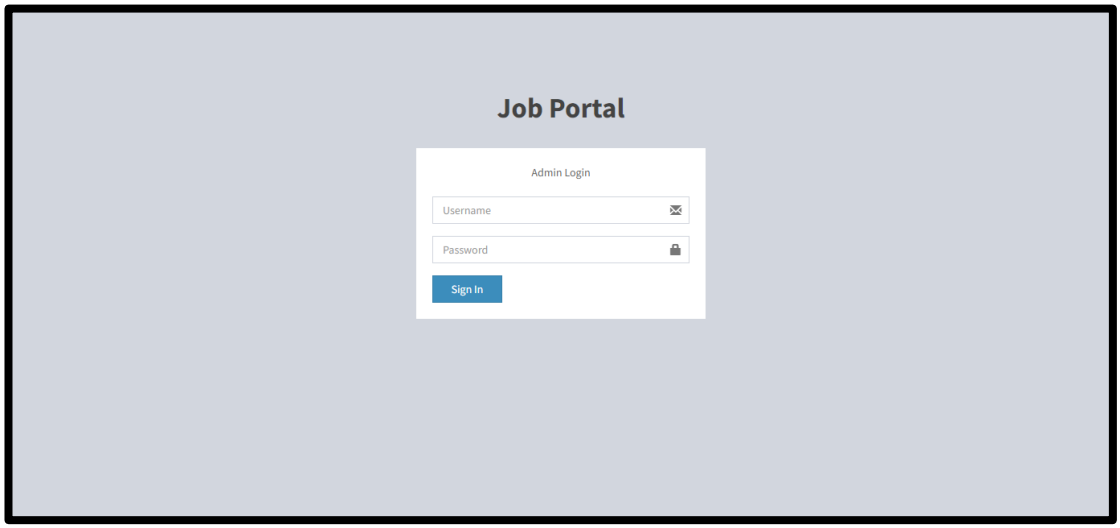


Figure 4.14: Login (Admin)

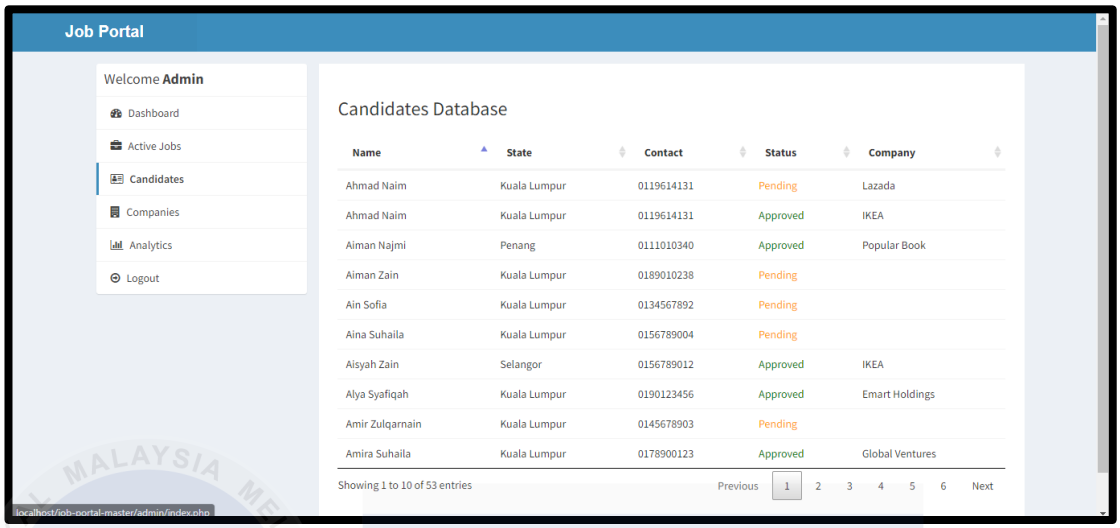


Figure 4.15: Applicant Listing (Admin)

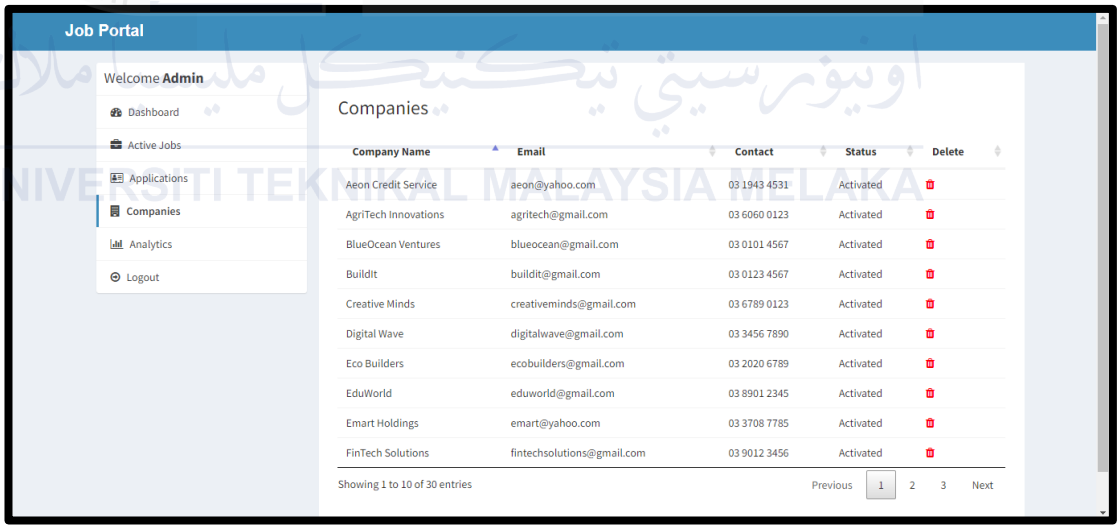


Figure 4.16: Company Listing (Admin)

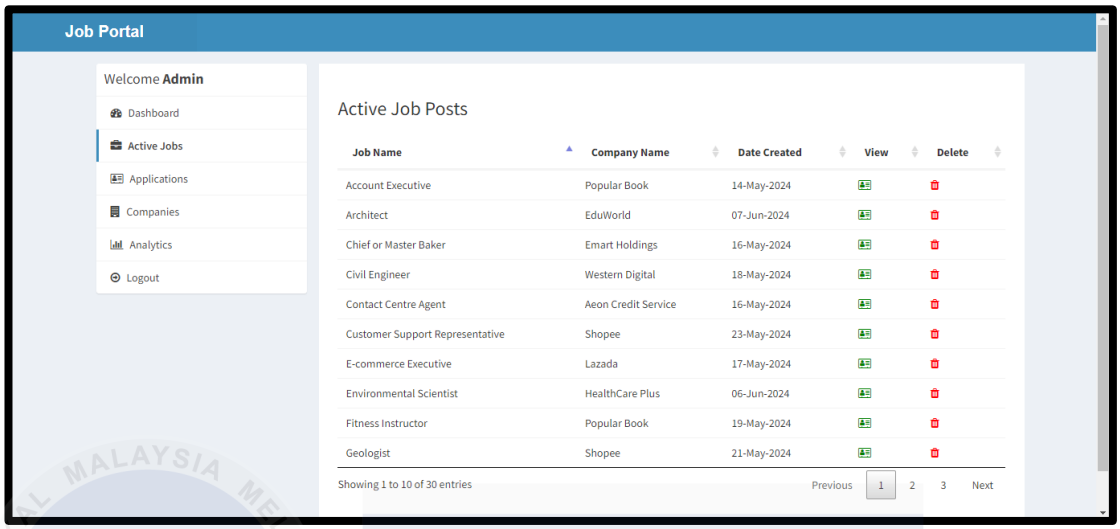


Figure 4.17: Job Listing (Admin)

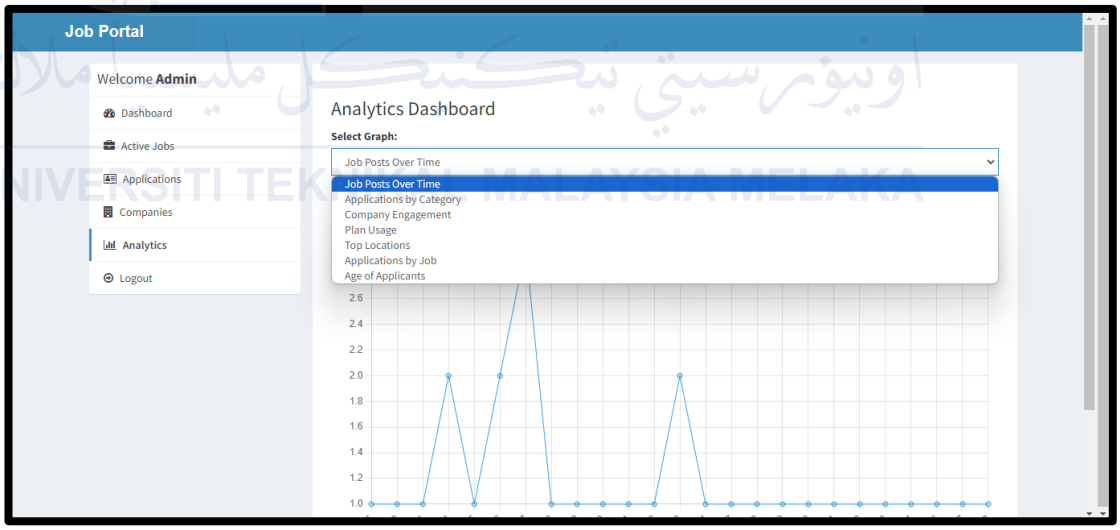


Figure 4.18: Analytics (Admin)

Input and Output:

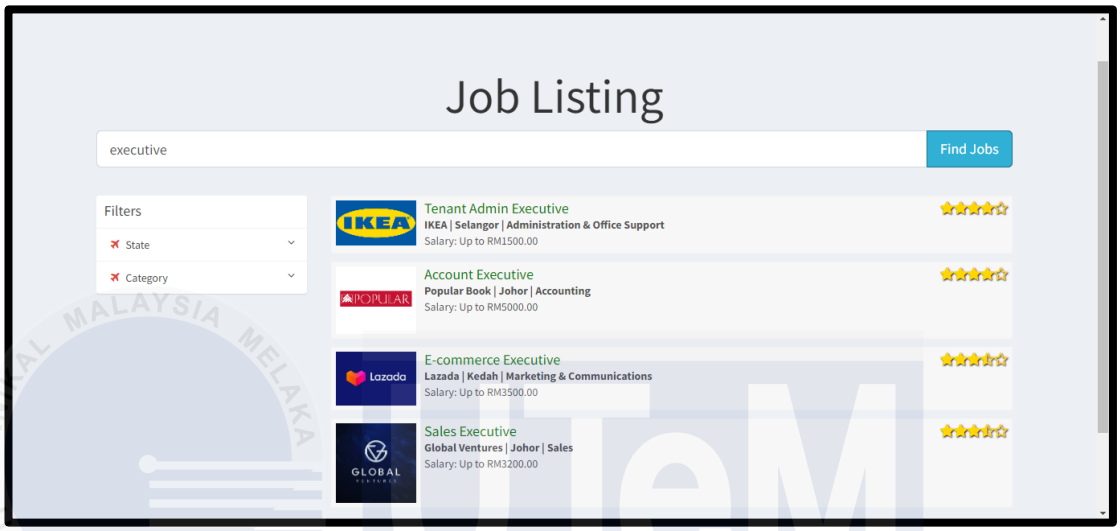


Figure 4.19: Job Search

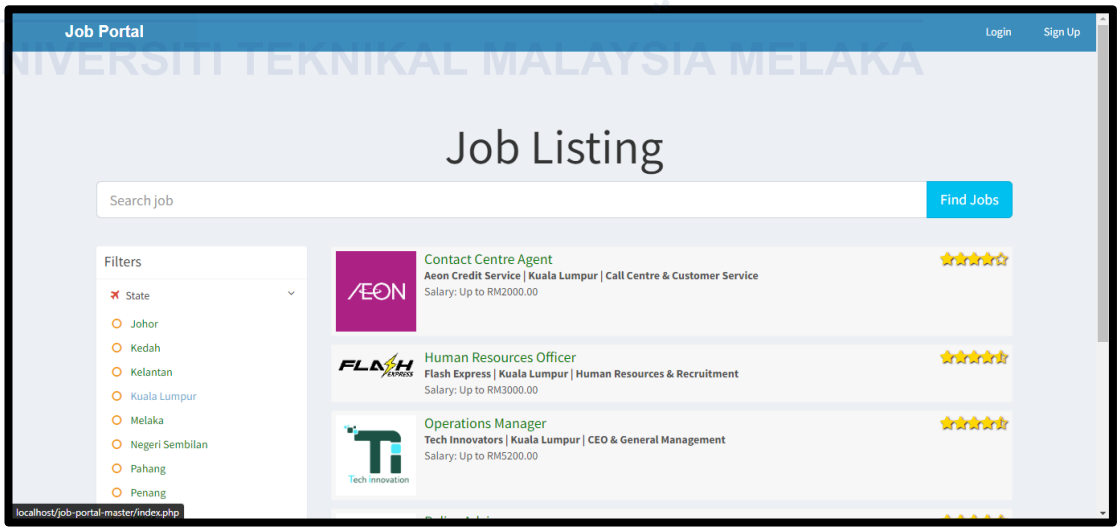


Figure 4.20: Filter by State

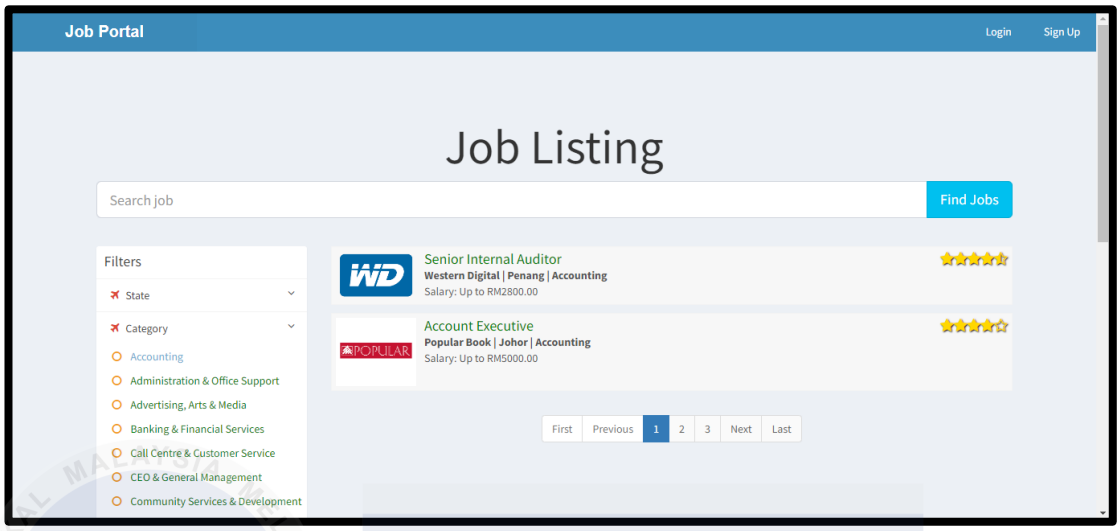


Figure 4.21: Filter by Category

4.5 Conclusion

This chapter provided a detailed overview of the design phase for the U-Job Portal, including the system architecture, database design, and graphical user interface design. The design strategies ensure that the system meets the identified requirements and provides a robust, scalable, and user-friendly platform for applicant and companies.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

In this chapter, we will delve into the implementation phase of the UTeM Job Portal system. This phase marks the transition from design to a functional product. The primary activities in this phase include setting up the software development environment, creating and configuring the database, implementing core functionalities, and populating the database with initial data. The expected outcome of this phase is a fully operational UTeM Job Portal system that can be deployed and tested for further enhancements.

5.2 Software Development Environment Setup

Setting up the software development environment is a critical step in the implementation process. It ensures that all the necessary tools and configurations are in place to support the development and execution of the UTeM Job Portal system.

i. System and Database Environment Setup

For the UTeM Job Portal, XAMPP was used as the development environment. XAMPP is an easy-to-install package that includes Apache, MySQL (MariaDB), PHP, and phpMyAdmin, which are all essential for developing PHP-based web applications.

ii. Installation and Configuration Steps

a) XAMPP Installation

Figure 5.1: XAMPP website

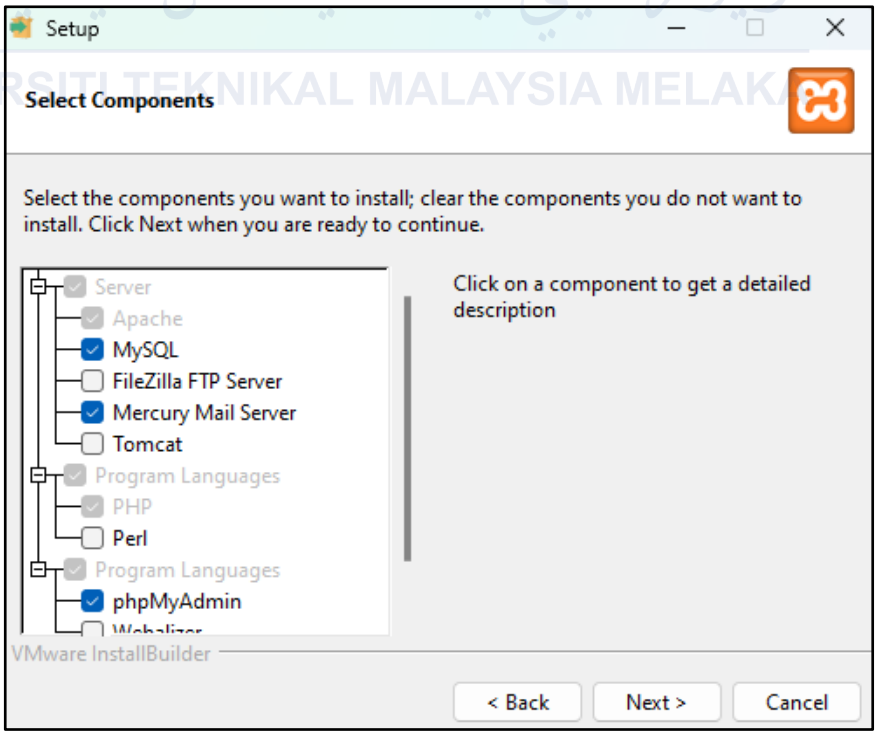


Figure 5.2: Components Selection

- Download the latest version of XAMPP from the official Apache Friends website at https://www.apachefriends.org/.

- Run the installer and select the components you need (Apache, MySQL, PHP, phpMyAdmin).

- Complete the installation and launch the XAMPP Control Panel.

b) Starting Apache and MySQL

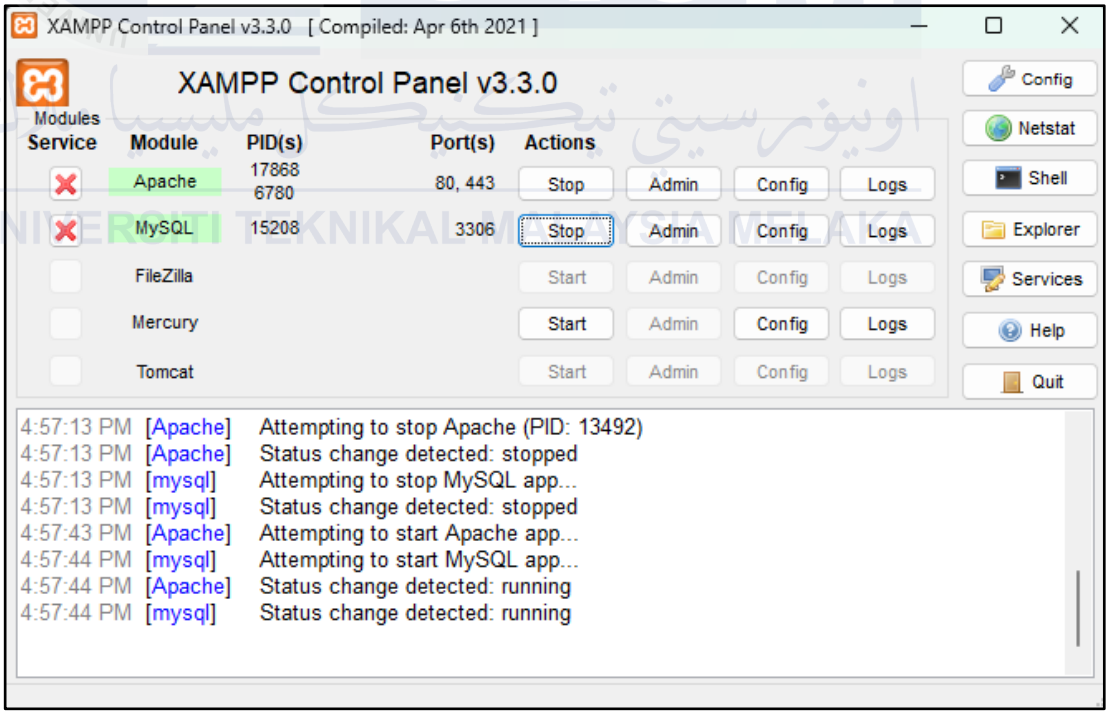


Figure 5.3: XAMPP Control Panel

- Open the XAMPP Control Panel.
- Start the Apache and MySQL services by clicking the "Start" buttons next to each.
- Verify that both services are running (indicated by green status lights).

c) Accessing phpMyAdmin

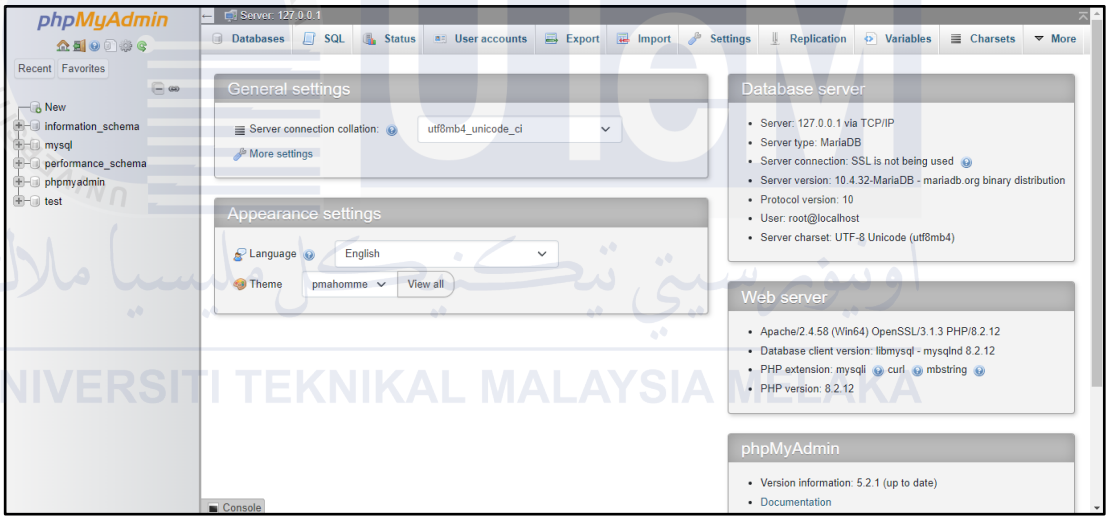
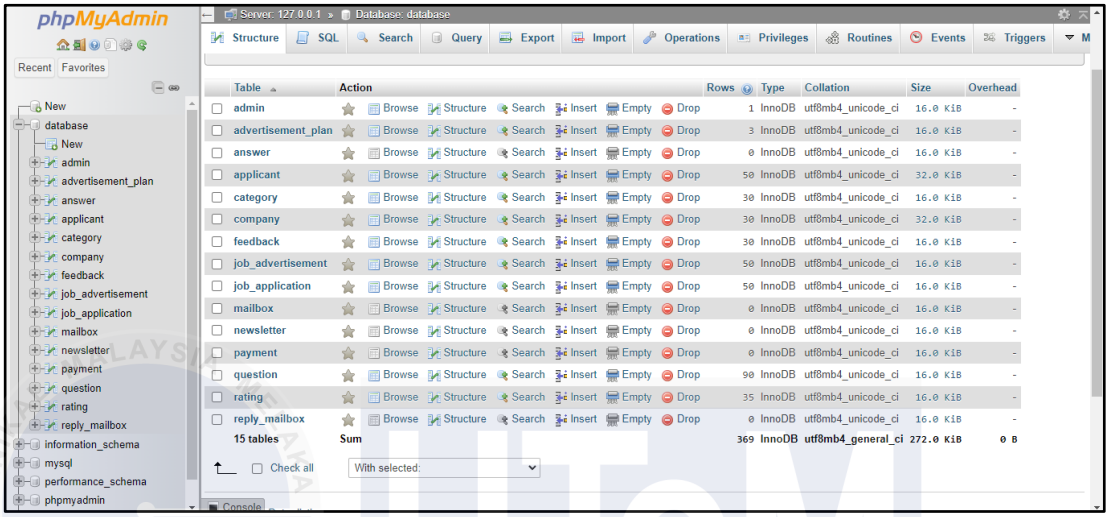


Figure 5.4: phpMyAdmin

- In the XAMPP Control Panel, click on the "Admin" button next to MySQL to open phpMyAdmin in your web browser.
- phpMyAdmin can also be accessed by navigating to http://localhost/phpmyadmin in a browser.

d) Setting Up the Database



- In phpMyAdmin, create the database for the UTeM Job Portal for all further SQL operations.

5.3 Database Implementation

The database implementation for the UTeM Job Portal involved several key steps, including the creation of tables, setting up relationships between those tables, and implementing necessary constraints. This section details the DDL (Data Definition Language) statements used to define the database schema, as well as the logic behind some of the primary processes implemented in the database.

a) Data Definition Language (DDL) Statements

DDL statements were used to create and define the structure of the database. These statements include commands like `CREATE TABLE`, `ALTER TABLE`, and `DROP TABLE`, which are essential for defining the schema of the database. Below are examples of the DDL statements used in the UTeM Job Portal project.

Table 5.1: Create Table

SQL Statement	Description
CREATE TABLE `admin` (`id_admin` int(11) NOT NULL, `username` varchar(255) NOT NULL, `password` varchar(255) NOT NULL, PRIMARY KEY (`id_admin`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;	The `admin` table stores credentials for system administrator who manage the job portal.
CREATE TABLE `applicant` (`id_user` int(11) NOT NULL, `firstname` varchar(255) NOT NULL, `lastname` varchar(255) NOT NULL, `email` varchar(255) NOT NULL, `password` varchar(255) NOT NULL, `age` int(11) DEFAULT NULL, `address` text DEFAULT NULL,	The `applicant` table stores detailed information about job applicants, including their personal details, qualifications, and resumes.

`state` varchar(255) DEFAULT NULL, `contactno` varchar(255) DEFAULT NULL, `qualification` varchar(255) DEFAULT NULL, `resume` varchar(255) DEFAULT NULL, `aboutme` text DEFAULT NULL, `skills` text DEFAULT NULL, `faculty` varchar(255) DEFAULT NULL, `cgpa` DECIMAL(3,2) NOT NULL, `code` mediumint(50) NOT NULL, PRIMARY KEY (`id_user`), UNIQUE KEY `email` (`email`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;	
CREATE TABLE `company` (`id_company` int(11) NOT NULL, `registration` varchar(20) NOT NULL, `companyname` varchar(255) NOT NULL, `contactno` varchar(255) NOT NULL,	The `company` table holds information about companies that post job advertisements. It includes fields for the company name, registration

`email` varchar(255) NOT NULL, `password` varchar(255) NOT NULL, `aboutme` varchar(255) DEFAULT NULL, `logo` varchar(255) NOT NULL, `numemployees` varchar(255) NOT NULL, `id_plan` int(11) NOT NULL DEFAULT 1, `createdAt` timestamp NOT NULL DEFAULT current_timestamp(), `active` int(11) NOT NULL DEFAULT 2, PRIMARY KEY (`id_company`), UNIQUE KEY `email` (`email`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;	details, contact information, and more.
CREATE TABLE `job_advertisement` (`id_jobpost` int(11) NOT NULL, `id_company` int(11) NOT NULL, `id_category` int(11) NOT NULL, `jobtitle` varchar(255) NOT NULL, `description` text NOT NULL,	The `job_advertisement` table captures the details of job postings made by companies. Each job advertisement is linked to a specific company and category.

`jobtype` varchar(255) NOT NULL, `address` varchar(255) NOT NULL, `state` varchar(255) NOT NULL, `salary` decimal(10,2) DEFAULT NULL, `createdat` timestamp NOT NULL DEFAULT current_timestamp(), PRIMARY KEY (`id_jobpost`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;	
CREATE TABLE `job_application` (`id_apply` int(11) NOT NULL, `id_jobpost` int(11) NOT NULL, `id_company` int(11) NOT NULL, `id_user` int(11) NOT NULL, `status` int(11) NOT NULL DEFAULT 0, PRIMARY KEY (`id_apply`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;	The `job_application` table stores records of job applications submitted by applicants for specific job postings. Each record links an applicant to a job advertisement.
CREATE TABLE `advertisement_plan` (	This table defines the different

`id_plan` int(11) NOT NULL, `plan_name` varchar(50) NOT NULL, `price` decimal(10,2) NOT NULL, `resume_file_downloads` tinyint(1) NOT NULL DEFAULT 0, `featured_company` tinyint(1) NOT NULL DEFAULT 0, `job_post_feedback` tinyint(1) NOT NULL DEFAULT 0, `advanced_reporting` tinyint(1) NOT NULL DEFAULT 0, `jobpost_limit` int(11) NOT NULL, PRIMARY KEY (`id_plan`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;	advertisement plans available to companies on the job portal. Each plan includes details like the plan name, price, and various features.
CREATE TABLE `answer` (`id_answer` int(11) NOT NULL, `id_question` int(11) NOT NULL, `id_user` int(11) NOT NULL,	This table stores responses to questions posed in feedback forms. Each answer is associated with a specific question,

`id_feedback` int(11) NOT NULL, `answer` text NOT NULL, `created_at` timestamp NOT NULL DEFAULT current_timestamp(), PRIMARY KEY (`id_answer`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;	user, and feedback session.
CREATE TABLE `category` (`id_category` int(11) NOT NULL, `name` varchar(255) NOT NULL, PRIMARY KEY (`id_category`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;	This table categorizes the different job types available on the portal, such as Accounting, Engineering, Healthcare, etc. Each job advertisement is linked to a specific category.
CREATE TABLE `feedback` (`id_feedback` int(11) NOT NULL, `id_jobpost` int(11) NOT NULL, `title` varchar(255) NOT NULL, `description` text NOT NULL,	This table stores feedback provided by companies about job postings. Each feedback entry is tied to a specific job post and includes a

`created_at` timestamp NOT NULL DEFAULT current_timestamp(), PRIMARY KEY (`id_feedback`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;	title, description, and timestamp.
CREATE TABLE `mailbox` (`id_mailbox` int(11) NOT NULL, `id_fromuser` int(11) NOT NULL, `fromuser` varchar(255) NOT NULL, `id_touser` int(11) NOT NULL, `subject` varchar(255) NOT NULL, `message` text NOT NULL, `createdAt` timestamp NOT NULL DEFAULT current_timestamp(), PRIMARY KEY (`id_mailbox`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;	The `mailbox` table manages messages sent between users on the platform. It includes fields for sender and receiver IDs, subject, message content, and timestamps.
CREATE TABLE `payment` (`id_payment` int(11) NOT NULL, `id_company` int(11) NOT NULL,	The `payment` table records payment transactions made by companies for

`payment_method` varchar(50) NOT NULL, `amount` decimal(10,2) NOT NULL, `status` varchar(20) NOT NULL, `payment_date` timestamp NOT NULL DEFAULT current_timestamp(), PRIMARY KEY (`id_payment`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;	advertisement plans. It includes details such as the payment method, amount, status, and date.
CREATE TABLE `question` (`id_question` int(11) NOT NULL, `id_feedback` int(11) NOT NULL, `question` varchar(255) NOT NULL, `frm_option` text DEFAULT NULL, `type` enum('radio_opt','check_opt','textfield_s') NOT NULL, `sort_order` int(11) NOT NULL, `created_at` timestamp NOT NULL DEFAULT current_timestamp(), PRIMARY KEY (`id_question`)	This table contains questions associated with feedback forms. It supports different types of input, such as radio buttons, checkboxes, and text fields.

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;	
CREATE TABLE `rating` (`id_rating` int(11) NOT NULL, `id_feedback` int(11) NOT NULL, `id_user` int(11) NOT NULL, `rating` decimal(3,2) NOT NULL, `created_at` timestamp NOT NULL DEFAULT current_timestamp(), PRIMARY KEY (`id_rating`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;	The `rating` table stores user ratings for job advertisements or services provided on the portal. Each rating is associated with specific feedback and user.
CREATE TABLE `reply_mailbox` (`id_reply` int(11) NOT NULL, `id_mailbox` int(11) NOT NULL, `id_user` int(11) NOT NULL, `usertype` varchar(255) NOT NULL, `message` text NOT NULL, `createdAt` timestamp NOT NULL DEFAULT current_timestamp(),	This table stores replies to messages in the `mailbox` table, allowing for threaded conversations between users.

PRIMARY KEY (`id_reply`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;	

Table 5.2: Create Indexes

SQL Statement	Description
ALTER TABLE `admin` ADD PRIMARY KEY (`id_admin`);	Stores information related to the administrators of the job portal, including login credentials.
ALTER TABLE `advertisement_plan` ADD PRIMARY KEY (`id_plan`);	Contains details about the different advertisement plans available on the job portal, including pricing and features.
ALTER TABLE `answer` ADD PRIMARY KEY (`id_answer`);	Stores the answers provided by users to feedback or survey questions related to job advertisements.
ALTER TABLE `applicant` ADD PRIMARY KEY (`id_user`), ADD UNIQUE KEY `UQ_applicant_email` (`email`);	Holds detailed information about job applicants, including personal information, qualifications, and uploaded resumes.

ALTER TABLE `category` ADD PRIMARY KEY (`id_category`);	Lists the various job categories available on the job portal, such as Engineering, IT, and Healthcare.
ALTER TABLE `company` ADD PRIMARY KEY (`id_company`), ADD UNIQUE KEY `UQ_company_email` (`email`);	Contains information about the companies registered on the job portal, including contact details and subscription plans.
ALTER TABLE `feedback` ADD PRIMARY KEY (`id_feedback`);	Stores feedback or survey data related to specific job advertisements, including the feedback title and description.
ALTER TABLE `job_advertisement` ADD PRIMARY KEY (`id_jobpost`);	Contains information about job postings made by companies, including job title, description, type, and salary details.
ALTER TABLE `job_application` ADD PRIMARY KEY (`id_apply`);	Tracks applications submitted by applicants to various job advertisements, including the application status.
ALTER TABLE `mailbox`	Manages the messaging system between users, storing sent and received messages with subject and content.

ADD PRIMARY KEY (`id_mailbox`);	
ALTER TABLE `payment` ADD PRIMARY KEY (`id_payment`);	Records payment transactions made by companies for advertisement plans or other services on the job portal.
ALTER TABLE `question` ADD PRIMARY KEY (`id_question`);	Contains questions related to feedback or surveys associated with job advertisements, along with the type and sort order.
ALTER TABLE `rating` ADD PRIMARY KEY (`id_rating`);	Stores ratings given by users to various feedback entries, capturing user ID, feedback ID, and the rating value.
ALTER TABLE `reply_mailbox` ADD PRIMARY KEY (`id_reply`);	Manages replies to messages in the mailbox system, linking replies to the original message and the user who replied.

b) Stored Procedures and Triggers

To handle complex operations within the database, stored procedures and triggers were implemented.

Table 5.3: Create Trigger

SQL Statement	Description

DELIMITER $$
 	
CREATE TRIGGER update_job_application_status
 	
AFTER UPDATE ON job_application
 	
FOR EACH ROW
 	
BEGIN
 	
IF NEW.status = 2 THEN
 	
UPDATE job_application
 	
SET status = 2
 	
WHERE id_apply = NEW.id_apply;
 	
END IF; END$$
 	
DELIMITER ;
	This trigger automatically updates the status of a job application to 'SHORTLISTED' (status = 2) when a company marks it as such.

DELIMITER $$ CREATE TRIGGER log_payment_update AFTER UPDATE ON payment FOR EACH ROW BEGIN INSERT INTO payment_log (id_payment, old_status, new_status, updated_at) VALUES (NEW.id_payment, OLD.status, NEW.status, NOW()); END$$ DELIMITER ;	This trigger logs any update made to the payment table, particularly when a payment status changes.

Table 5.4: Create Procedure

SQL Statement	Description
DELIMITER // CREATE PROCEDURE AddJobAdvertisement(IN company_id INT, IN category_id INT,	A stored procedure was created to add new job advertisements efficiently

```
IN job_title VARCHAR(255),

IN job_description TEXT,

IN job_type VARCHAR(255),

IN job_address VARCHAR(255),

IN job_state VARCHAR(255),

IN job_salary DECIMAL(10,2)

)

BEGIN

INSERT INTO job_advertisement
(id_company, id_category, jobtitle,
description, jobtype, address,
state, salary)
VALUES (company_id, category_id,
job_title, job_description,
job_type, job_address, job_state,
job_salary);

END //

DELIMITER ;
```

c) Data Loading Process

Data loading involved populating the database with initial records for testing and demonstration purposes. This process was carried out using SQL INSERT statements. Below are examples of initial data entries

SQL Statement	Data Name
INSERT INTO applicant (id_user, firstname, lastname, email, password, age, address, state, contactno, qualification, resume, aboutme, skills, faculty, cgpa, code) VALUES (1, 'Syed', 'Nor Irfan', 'syednorirfan26@gmail.com', '$2y$10$xhE03pJGZIkNCVwHzek34uptu9vyt5Al6SuULpj8uEdtQe2BirBuy', 21, 'Jalan Tunku Abdul Rahman, 50100', 'Kuala Lumpur', '0169796052', 'Bachelor Degree in Electrical Engineering', '666d1c556ff19.pdf', 'A dedicated and innovative Electrical Engineering graduate passionate about developing and implementing electrical systems.', 'Circuit design, Power systems, MATLAB, AutoCAD', 'Faculty of Electrical Technology and Engineering', 3.75, 0);	Applicant Data
INSERT INTO company (id_company, registration, companyname, contactno, email, password,	Company Data

aboutme, logo, numemployees, id_plan, createdAt, active) VALUES (1, '123456-A', 'IKEA', '0348947263', 'ikea@yahoo.com', '$2y$10$xhE03pJGZIkNCVwHzek34uptu9vyt5Al6SuULpj8uEdtQe2BirBuy', 'IKEA is a Swedish-founded multinational group that designs and sells ready-to-assemble furniture, kitchen appliances, and home accessories.', 'ikea.jpg', '500+', 3, '2024-04-10 12:44:33', 1);	
INSERT INTO job_advertisement (`id_jobpost`, `id_company`, `id_category`, `jobtitle`, `description`, `jobtype`, `address`, `state`, `salary`, `createdat`) VALUES (1, 1, 25, 'Store Manager', 'Oversee the daily operations of the store, ensuring smooth workflow and high customer satisfaction.', 'Full-time', 'Jalan Tun Razak', 'Kuala Lumpur', 4500.00, '2024-05-05 12:52:09');	Job Advertisement Data

5.4 Conclusion

In this chapter, the implementation phase of the UTeM Job Portal system has been thoroughly discussed. This phase involved setting up the software development environment, creating the database and its objects, implementing core processes, and loading initial data. With the successful completion of this phase, the UTeM Job Portal is now a functional system ready for testing. The next steps involve system testing to identify any issues or bugs and making necessary adjustments to ensure that the system meets all specified requirements.

CHAPTER 6: TESTING

6.1 Introduction

This chapter provides a detailed overview of the testing phase conducted for the UTeM Job Portal system. The testing phase is a critical part of the software development life cycle, aimed at ensuring the system's reliability, functionality, and performance. During this phase, various tests were conducted to validate that the system meets the specified requirements and performs as expected under different conditions. The primary focus was on identifying and fixing any bugs or issues that could affect the user experience or system integrity. The testing strategy adopted for this project included both White Box and Black Box testing methods, ensuring a comprehensive evaluation of the system's internal logic and external behavior. This chapter will detail the test plan, environment, schedule, strategy, design, and results, culminating in an analysis of the system's performance and user satisfaction.

6.2 Test Plan

The test plan outlines the systematic approach taken to ensure that the UTeM Job Portal meets all functional and non-functional requirements. It serves as a roadmap for the testing activities, providing a structured framework for identifying, documenting, and resolving any issues or bugs encountered during the testing phase. The test plan covers the objectives, scope, criteria, and schedule of the testing process, ensuring that every aspect of the system is thoroughly tested before deployment.

6.2.1 Test Organization

The testing phase for the UTeM Job Portal system is organized to ensure that all aspects of the system are thoroughly tested and evaluated. The test organization consists of three primary roles which are admin, applicant and employer. Each role brings specific responsibilities to the table. The roles are designed to cover different aspects of testing, from planning to execution and feedback. Table 6.1 below outlines the key personnel involved in the testing phase.

Table 6.1: List of User's Responsibilities

Tester ID	Name	Position	Responsibilities
T1	Syed Nor Irfan	System developer and Test Manager	- Prepare test plan and test script - Perform unit testing and integration testing
T2	Chaessler, Fakhrul Akram, Nur Izzati	Administrator	- User Management module
T3	Hani Mastura, Hana Sofea, Khairul Khatami	Applicant	- Job Search module - Application Submission module
T4	Danial, Muhammad Shah, Muhammad Darwisy, Ahmad Amirul Farhan	Employer	- Job Posting module - Application Review & Management module

6.2.2 Test Environment

The test environment is critical for ensuring that the UTeM Job Portal system is evaluated under conditions that closely mimic its eventual deployment environment. This includes both hardware and software configurations that will be used during the testing phase. Setting up a proper test environment helps in identifying potential issues that may arise in the production environment, ensuring a smoother transition once the system is fully deployed.

Table 6.2: Hardware Equipment

No.	Environment Specification	Description
1	Laptop	Aspire A315-23
2	Processor	AMD Athlon Silver 3050U with Radeon Graphics
3	Installed RAM	4 GB
4	Mouse	Gaming Mouse M160
5	Storage	237 GB

Table 6.3: Software Environment

No.	Environment Specification	Description
1	Operating System	Windows 11 The system operates on Windows 10, providing a stable and widely used environment for testing web applications.
2	Database	MySQL MySQL is the database management system used to store and manage the data for the UTeM Job Portal.
3	Web Server	XAMPP XAMPP is used as the local server to host the UTeM Job Portal during testing. It includes Apache for web serving and MySQL for database management.
4	Development Tools	Visual Studio Code Visual Studio Code is used for coding, debugging, and testing the UTeM Job Portal. It supports HTML, CSS, JavaScript, PHP, and more.
5	Documentation	Microsoft Word

		It is essential for documentation throughout the project lifecycle, ensuring that all written materials are professional and well-organized.
6	Browser	Google Chrome Testing is conducted across multiple web browsers to ensure cross-browser compatibility.
7	Diagramming Tool	Draw.io Diagrams created in Draw.io help visualize the system's structure, aiding in both development and testing.

6.2.3 Test Schedule

The test schedule for the UTeM Job Portal system is designed to ensure that all key modules of the system are thoroughly tested within a structured timeline. This schedule serves as a guide for the testing team, helping them to manage time effectively and ensure that all testing activities are completed before the system is deployed.

Table 6.4: Test Schedule

Module	Start Date	End Date	Test Type	Tester
User Management	16/07/2024	25/07/2024	Unit Testing	T1, T2
Job Search	26/07/2024	28/07/2024	Integration Testing	T3
Application Submission	29/07/2024	31/07/2024	System Testing	T3
Job Posting	01/08/2024	03/08/2024	Unit Testing	T4
Application Review and Management	04/08/2024	06/08/2024	Integration Testing	T4

6.3 Test Strategy

The test strategy for the UTeM Job Portal system was designed to ensure thorough and comprehensive testing of both the internal workings and external functionalities of the system. The strategy employed a combination of White Box Testing and Black Box Testing methods, each selected to address specific aspects of the system's architecture and user interactions.

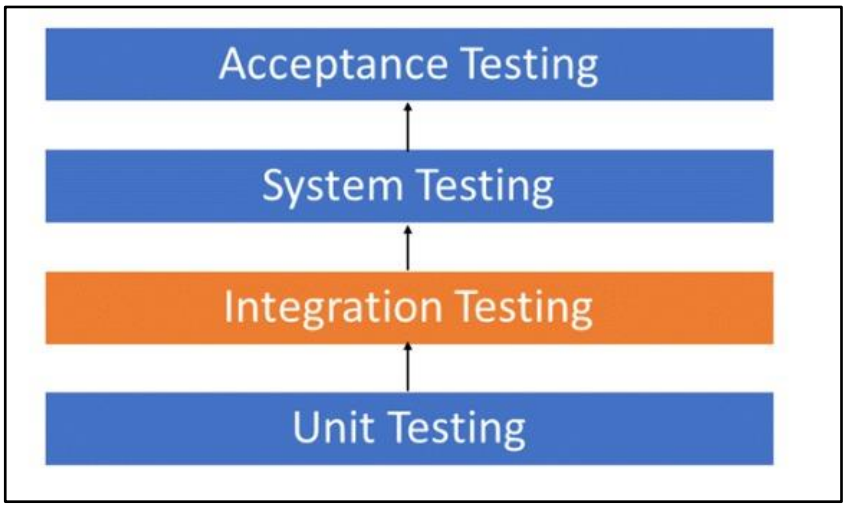


Figure 6.1: Level of Testing (Hamilton, 2024)

i. White Box Testing

This approach was used to validate the internal code structure, logic, and flow of the system. The primary focus was on unit testing and integration testing, where individual components and their interactions were examined to ensure they function correctly. This method helped in identifying logical errors, inefficient code, and security vulnerabilities within the system.

ii. Black Box Testing

This approach focused on the external functionalities of the system without considering the internal code structure. It included system testing and user acceptance testing (UAT), where the system was tested from the user's perspective. This method was essential for validating that the system meets the user requirements and performs as expected under various scenarios.

The overall strategy adopted a top-down approach, beginning with higher-level modules and progressively testing more detailed components. This ensured that the critical functionalities of the UTeM Job Portal were verified early in the testing phase, allowing for timely identification and resolution of any major issues.

6.3.1 Classes of Tests

The UTeM Job Portal system underwent several classes of tests to ensure its functionality, security, and performance met the project's objectives. These tests were designed to cover various aspects of the system, from verifying the correct output to ensuring the system's security and its ability to handle stress under different conditions.

i. Functionality

Functionality testing focused on verifying that all features of the UTeM Job Portal operated according to the specified requirements. This included testing essential functionalities such as user registration, job search, application submission, employer job posting, and admin management. Each module was tested to ensure that it

produced the correct outputs and interacted seamlessly with other modules. The primary goal was to confirm that users could perform all expected tasks without encountering errors.

ii. Security Testing

Security testing was conducted to ensure that the UTeM Job Portal was protected against unauthorized access and data breaches. This involved testing the system's login mechanisms, data encryption processes, and user authentication protocols. Tests were also performed to check for vulnerabilities such as SQL injection, cross-site scripting (XSS), and other common security threats. The objective was to safeguard user data and maintain the integrity of the system by preventing unauthorized access or malicious activities.

iii. Integration Testing

Integration testing focused on verifying that the different modules of the UTeM Job Portal worked together as intended. This class of tests ensured that data flowed correctly between modules such as the job posting system, applicant tracking, and user management. Integration testing was crucial in identifying issues that could arise when individual modules interact, ensuring a seamless and consistent user experience across the platform.

6.4 Test Design

Test design is a critical phase in the testing process, where detailed test cases are developed to guide the testing of the UTeM Job Portal system. The objective of test design is to ensure that all system functionalities are thoroughly examined and validated against the requirements. Test design encompasses two essential components, the test description and the test data.

6.4.1 Test Description

The test description section provides a detailed outline of the test cases for each module within the UTeM Job Portal system. This segment meticulously covers the test design, identifying crucial elements such as the test case identifier, test date, test case specifications, test type, and expected outcome. These test cases serve as a benchmark for ensuring that the system meets the specified requirements and operates consistently under various scenarios.

Table 6.5: Test Description for Applicant Registration

Test Module	User Management – Applicant Registration		
Test Type	Unit Testing		
Date	16/7/24 – 18/7/24	**Test Strategy**	Black Box Testing
Description	Testing the user registration functionality to ensure new applicants can successfully register and their accounts are properly created.		
Test Case ID	**Test Case Description**	**Test Step**	**Expected Result**
TC1_1	Verify applicant registration functionality with all required fields correctly filled.	1. Go to the registration page. 2. Enter valid details. 3. Submit the form.	Applicant account is created successfully, and a confirmation prompt is displayed.
TC1_2	Verify applicant registration fails if	1. Go to the registration page.	Registration fails with an error prompt indicating that all

	required fields are left blank.	2. Leave one or more required fields blank. 3. Attempt to submit the form.	required fields must be filled.
TC1_3	Verify user cannot register with an email that is already in use.	1. Go to the registration page. 2. Enter an email address that is already registered. 3. Submit the form.	Registration fails with an error prompt indicating that the email is already in use.

Table 6.6: Test Description for Applicant Login

Test Module	User Management – Applicant Login		
Test Type	Unit Testing		
Date	19/7/24 – 21/7/24	**Test Strategy**	Black Box Testing
Description	Testing the login functionality and profile management to ensure applicants can log in securely and update their profile information.		
Test Case ID	**Test Case Description**	**Test Step**	**Expected Result**
TC2_1	Verify successful login with valid credentials.	1. Go to the login page. 2. Enter valid email and	Applicant successfully logs in and is redirected to the dashboard.

		password. 3. Click the login button.	
TC2_2	Verify login fails with invalid credentials.	1. Go to the login page. 2. Enter an invalid email or password. 3. Attempt to log in.	Login fails with an error message indicating that the credentials are incorrect.

Table 6.7: Test Description for Password Recovery

Test Module	User Management - Password Recovery		
Test Type	Unit Testing		
Date	22/7/24	**Test Strategy**	Black Box Testing
Description	Testing the password recovery functionality to ensure users can reset their passwords if forgotten.		
Test Case ID	**Test Case Description**	**Test Step**	**Expected Result**
TC3_1	Verify password reset functionality for users who forgot their password.	1. Go to the login page. 2. Click on "Forgot Password". 3. Enter the registered email.	A password reset link is sent to the user's email, and a confirmation message is displayed.

		4. Submit the form.	
TC3_2	Verify password reset fails if an unregistered email is entered.	1. Go to the login page. 2. Click on "Forgot Password". 3. Enter an unregistered email. 4. Submit the form.	Reset request fails with a prompt indicating that the email is not associated with an account.

Table 6.8: Test Description for Company Registration and Approval

Test Module	User Management - Company Registration and Approval		
Test Type	Unit Testing		
Date	23/7/24 – 25/7/24	**Test Strategy**	Black Box Testing
Description	Testing the company registration process and the subsequent approval by an admin to ensure that only verified companies can registered.		
Test Case ID	**Test Case Description**	**Test Step**	**Expected Result**
TC4_1	Verify that companies can submit registration requests with all required details.	1. Go to the company registration page. 2. Enter all required	Registration request is submitted successfully, and a message is displayed indicating that the application is under review.

		company details. 3. Submit the registration form.	
TC4_2	Verify that admin can view and approve pending company registration requests.	1. Log in as an admin. 2. Navigate to the company registration approval page. 3. Review the details of a pending request. 4. Approve the registration.	The company registration is approved.
TC4_3	Verify that admin can view and reject pending company registration requests.	1. Log in as an admin. 2. Navigate to the company registration approval page. 3. Review the details of a pending request. 4. Reject the registration.	The company registration is rejected.

Table 6.9: Test Description for Job Search

Test Module	Job Search		
Test Type	Integration Testing		
Date	26/7/24 – 28/7/24	**Test Strategy**	Black Box Testing
Description	Testing the job search functionality to ensure users can search for jobs using various filters like location, job type, and industry, and that the search results are accurate and relevant.		
Test Case ID	**Test Case Description**	**Test Step**	**Expected Result**
TC5_1	Verify job search functionality and multiple filters (state or category).	1. Go to the job search page. 2. Select a state filter. 3. Select a category filter. 4. Click the search button.	Jobs matching both the state and category filters are displayed correctly in the results.
TC5_2	Verify job search functionality when no filters are applied.	1. Go to the job search page. 2. Leave all filters unselected. 3. Click the search button.	All available jobs are displayed in the search results without any filtering.
TC5_3	Verify that job search functionality handles invalid input gracefully.	1. Go to the job search page. 2. Enter invalid characters in the	The system handles the invalid input without crashing and provides an

		keyword search field. 3. Click the search button.	appropriate error message if needed.
TC5_4	Verify that job search returns relevant results for a specific keyword search.	1. Go to the job search page. 2. Enter a relevant keyword in the search field (Example = "Accountant"). 3. Click the search button.	Jobs matching the keyword are displayed in the search results.

Table 6.10: Test Description for Application Submission

Test Module	Application Submission		
Test Type	System Testing		
Date	29/7/24 – 31/7/24	**Test Strategy**	White Box Testing
Description	Testing the application submission process to ensure applicants can apply for jobs successfully, and that the system tracks and updates the status of applications correctly.		
Test Case ID	**Test Case Description**	**Test Step**	**Expected Result**
TC6_1	Verify successful submission of a job application.	1. Log in as an applicant. 2. Search and select a job.	Application is submitted successfully, and a confirmation message is

		3. Click the apply button. 4. Fill in application details. 5. Submit the application.	displayed to the applicant.
TC6_2	Verify that the system prevents duplicate applications for the same job.	1. Log in as an applicant. 2. Apply for a job. 3. Attempt to apply for the same job again.	The system prevents the duplicate application and displays a message indicating that the application has already been submitted.
TC6_3	Verify that applicants can view and track the status of their submitted applications.	1. Log in as an applicant. 2. Navigate to the application status page. 3. View the status of submitted applications.	Applicants can see the current status of each submitted application (Example = Pending, Shortlisted, Hired and Rejected).

Table 6.11: Test Description for Job Posting

Test Module	Job Posting		
Test Type	Unit Testing		
Date	1/8/24 – 3/8/24	**Test Strategy**	Black Box Testing
Description	Testing the job posting functionality to ensure employers can create, update, and delete job listings as needed, and that these listings are displayed accurately in the job search results.		
Test Case ID	**Test Case Description**	**Test Step**	**Expected Result**
TC7_1	Verify that employers can successfully create a new job posting.	1. Log in as an employer. 2. Navigate to the job posting page. 3. Enter all required job details. 4. Submit the job posting.	The job posting is created successfully and is visible in the job search results.
TC7_2	Verify that the system prevents job posting if required fields are left blank.	1. Log in as an employer. 2. Navigate to the job posting page. 3. Leave one or more required fields blank. 4. Attempt to	The system prevents the submission and displays an error message indicating that all required fields must be filled.

		submit the job posting.	
TC7_3	Verify that employers can edit an existing job posting.	1. Log in as an employer. 2. Navigate to the job posting management page. 3. Select a job to edit. 4. Modify job details. 5. Save changes.	The job posting is updated successfully, and the changes are reflected in the job search results.
TC7_4	Verify that employers can delete a job posting.	1. Log in as an employer. 2. Navigate to the job posting management page. 3. Select a job to delete. 4. Confirm deletion.	The job posting is deleted successfully and is no longer visible in the job search results.

Table 6.12: Test Description for Application Review and Management

Test Module	Application Review and Management		
Test Type	Integration Testing		
Date	4/8/24 – 6/8/24	**Test Strategy**	White Box Testing
Description	Testing the functionality where employers review applicants' details and resumes, and decide to Shortlist, Hire, or Reject applicants. The system also handles notifications to applicants about their application status.		
Test Case ID	**Test Case Description**	**Test Step**	**Expected Result**
TC8_1	Verify that employers can view the details and resume of applicants.	1. Log in as an employer. 2. Navigate to the list of applicants for a job posting. 3. Select an applicant. 4. View the applicant's details and resume.	The employer can view all the details and the resume of the selected applicant.
TC8_2	Verify that employers can shortlist an applicant.	1. Log in as an employer. 2. Select an applicant from the list. 3. Choose the	The applicant is marked as shortlisted, and the status is updated in the system.

		"Shortlisted" option. 4. Confirm the action.	
TC8_3	Verify that a message is sent to the applicant's mailbox when they are shortlisted.	1. Shortlist an applicant as an employer. 2. Log in as the applicant. 3. Check the mailbox for a new message.	A message is sent to the applicant's mailbox informing them that they have been shortlisted.
TC8_4	Verify that employers can mark an applicant as "Hired."	1. Log in as an employer. 2. Select a shortlisted applicant. 3. Choose the "Hired" option. 4. Confirm the action.	The applicant is marked as hired, and the status is updated in the system.
TC8_5	Verify that employers can mark an applicant as "Rejected."	1. Log in as an employer. 2. Select an applicant from the list. 3. Choose the "Reject" option. 4. Confirm the action.	The applicant is marked as rejected, and the status is updated in the system.

TC8_6	Verify that a message is sent to the applicant's mailbox when they are hired or rejected.	1. Mark an applicant as hired or rejected as an employer. 2. Log in as the applicant. 3. Check the mailbox for a new message.	A message is sent to the applicant's mailbox informing them of the hiring or rejection decision.
TC8_7	Verify that the status of the application is updated in the applicant's account after the employer's decision.	1. Log in as an applicant. 2. Navigate to the application status page. 3. Check the status of the application.	The status of the application is updated to reflect the employer's decision (e.g., Shortlisted, Hired, Rejected).

6.4.2 Test Data

The test data was used to verify the functionality and accuracy of the UTeM Job Portal system. The data simulates real-world scenarios to ensure that all modules perform as expected under various conditions. The results obtained from these tests confirm the system's reliability and effectiveness.

Table 6.13: Test Data for Applicant Registration

Test Data ID	Name	Email	Password	Confirm Password
TD1_1	Syed Nor Irfan	syed26@gmail.com	010203@S2d	010203@S2s
TD1_2		syed26@gmail.com	010203@S2s	010203@S2s
TD1_3	Syed Aiman	syed26@gmail.com	112233@S2a	111213@S2a

Table 6.14: Test Data for Applicant Login

Test Data ID	Email	Password
TD2_1	syed26@gmail.com	010203@S2s
TD2_2	syed28@gmail.com	010203@S2s

Table 6.15: Test Data for Password Recovery

Test Data ID	Email	Password
TD3_1	syed26@gmail.com	010203@S2s
TD3_2	ikea@gmail.com	918273@I2k

Table 6.16: Test Data for Company Registration and Approval

Test Data ID	Email	Password
TD4_1	ikea@gmail.com	918273@I2k
TD4_2	ikea@gmail.com	918273@I2k
TD4_3	petronas@gmail.com	123456@P6a

Table 6.17: Test Data for Job Search

Test Data ID	Key Word	Filter (State)	Filter (Category)
TD5_1		Pahang	
TD5_2	Accountant		
TD5_3	Accountan@		
TD5_4	Accountant		

Table 6.18: Test Data for Application Submission

Test Data ID	User ID	Job Title	Category	State	Salary
TD6_1	26	Web Developer	Information & Communication Technology	Selangor	1500
TD6_2	26	Web Developer	Information & Communication Technology	Selangor	1500
TD6_3	26	Web Developer	Information & Communication Technology	Selangor	1500

Table 6.19: Test Data for Job Posting

Test Data ID	TD7_1
Job Title	Store Manager
Description	Oversee the daily operations of the store, ensuring smooth workflow and high customer satisfaction Oversee the daily operations of the store, ensuring smooth workflow and high customer satisfaction Oversee the daily operations of the store, ensuring smooth workflow and high customer satisfaction
Category	Retail & Consumer Products
Job Type	Full Time
State	Kuala Lumpur
Salary	RM4500
Job Address	Jalan Tun Razak

Test Data ID	TD7_2
Job Title	Store Manager
Description	Oversee the daily operations of the store, ensuring smooth workflow and high customer satisfaction Oversee the daily operations of the store, ensuring smooth workflow and high customer satisfaction Oversee the daily operations of the store, ensuring smooth workflow and high customer satisfaction

Category	
Job Type	
State	Kuala Lumpur
Salary	RM4500
Job Address	Jalan Tun Razak

Test Data ID	TD7_3

Job Title	Store Manager
Description	Oversee the daily operations of the store, ensuring smooth workflow and high customer satisfaction Oversee the daily operations of the store, ensuring smooth workflow and high customer satisfaction Oversee the daily operations of the store, ensuring smooth workflow and high customer satisfaction
Category	Retail & Consumer Products
Job Type	Full Time
State	Selangor
Salary	RM4000
Job Address	Jalan Tun Razak

Test Data ID	TD7_4

Job Title	Store Manager
Description	Oversee the daily operations of the store, ensuring smooth workflow and high customer satisfaction Oversee the daily operations of the store, ensuring smooth workflow and high customer satisfaction Oversee the daily operations of the store, ensuring smooth workflow and high customer satisfaction
Category	Retail & Consumer Products
Job Type	Full Time
State	Selangor
Salary	RM4000
Job Address	Jalan Tun Razak

Table 6.20: Table Data for Application Review & Management

Test Data ID	Applicant's ID	Take Action (Button)
TD8_1	1	View
TD8_2	1	Shortlisted
TD8_3	1	Shortlisted
TD8_4	1	Hired
TD8_5	2	Rejected
TD8_6	2	Rejected
TD8_7	1	Shortlisted

6.5 Test Results and Analysis

This section presents the results of the test cases executed on the UTeM Job Portal System. It includes the identification of each test case, the tester responsible, the results (Success/Fail), and a detailed documentation of any failed test cases. Additionally, it provides an overall satisfaction assessment from both the intended users and the tester.

Table 6.21: Test Results and Analysis

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
User Management – Applicant Registration				
TC1_1	TD1_1	Applicant account is created successfully, and a confirmation prompt is displayed.	Applicant account was created successfully, and the confirmation prompt was displayed as expected.	Pass
TC1_2	TD1_1	Registration fails with an error prompt indicating that all required fields must be filled.	Registration failed as expected, displaying the correct error prompt for missing required fields.	Pass
TC1_3	TD1_1	Registration fails with an error prompt indicating that the email is already in use.	Registration failed as expected, displaying the correct error prompt for duplicate email usage.	Pass

User Management – Applicant Login				
TC2_1	TD2_1	Applicant successfully logs in and is redirected to the dashboard.	Applicant successfully logged in and was redirected to the dashboard as expected.	Pass
TC2_2	TD2_2	Login fails with an error message indicating that the credentials are incorrect.	Login failed as expected, with the system displaying the correct error message for incorrect credentials.	Pass
User Management - Password Recovery				
TC3_1	TD3_1	A password reset link is sent to the user's email, and a confirmation message is displayed.	The password reset link was sent successfully, and the confirmation message was displayed as expected.	Pass
TC3_2	TD3_2	Reset request fails with a prompt indicating that the email is not associated with an account.	Password reset failed as expected, displaying an error message indicating that the email is not associated with an account.	Pass

User Management – Company Registration & Approval				
TC4_1	TD4_1	Registration request is submitted successfully, and a message is displayed indicating that the application is under review	The registration request was submitted successfully, and the expected message was displayed.	Pass
TC4_2	TD4_2	The company registration is approved.	The admin was able to view and approve the registration request as expected.	Pass
TC4_3	TD4_3	The company registration is rejected.	The admin was able to reject the registration, and the company received the rejection notification when they try to login as expected.	Pass
Job Search				
TC5_1	TD5_1	Jobs matching both the state and category filters are displayed correctly in the results.	Jobs were displayed correctly in the search results, matching the selected filters for state and category.	Pass
TC5_2	TD5_2	All available jobs are displayed in the search results without any filtering.	All available jobs were displayed as expected when no filters were applied.	Pass
TC5_3	TD5_3	The system handles the invalid input without	The system handled invalid input	Pass

		crashing and provides an appropriate error message if needed.	gracefully, displaying an appropriate message without crashing.	
TC5_4	TD5_4	Jobs matching the keyword are displayed in the search results.	Jobs matching the entered keyword were displayed accurately in the search results.	Pass
Application Submission				
TC6_1	TD6_1	Application is submitted successfully, and a confirmation message is displayed to the applicant.	Application was submitted successfully, and the confirmation message was displayed as expected.	Pass
TC6_2	TD6_2	The system prevents the duplicate application and displays a message indicating that the application has already been submitted	Duplicate application attempts were blocked, and the appropriate message was displayed.	Pass
TC6_3	TD6_3	Applicants can see the current status of each submitted application (Example = Pending, Shortlisted, Hired and Rejected).	Applicants were able to view and track the status of their applications accurately, with statuses updated as expected.	Pass

Job Posting				
TC7_1	TD7_1	The job posting is created successfully and is visible in the job search results.	The job posting was created successfully and appeared in the job search results as expected.	Pass
TC7_2	TD7_2	The system prevents the submission and displays an error message indicating that all required fields must be filled.	The system correctly prevented the job posting and displayed the appropriate error message when required fields were left blank.	Pass
TC7_3	TD7_3	The job posting is updated successfully, and the changes are reflected in the job search results.	The job posting was updated successfully, and the changes were reflected in the job search results.	Pass
TC7_4	TD7_4	The job posting is deleted successfully and is no longer visible in the job search results.	The job posting was deleted successfully, and it was no longer visible in the job search results.	Pass
Application Review and Management				
TC8_1	TD8_1	The employer can view all the details and the resume of the selected applicant.	The employer was able to view the applicant's details and resume as expected.	Pass

TC8_2	TD8_2	The applicant is marked as shortlisted, and the status is updated in the system.	The applicant was marked as shortlisted, and the status was updated correctly in the system.	Pass
TC8_3	TD8_3	A message is sent to the applicant's mailbox informing them that they have been shortlisted.	A message was sent to the applicant's mailbox informing them of the shortlist status, as expected.	Pass
TC8_4	TD8_4	The applicant is marked as hired, and the status is updated in the system.	The applicant was marked as hired, and the status was updated correctly in the system.	Pass
TC8_5	TD8_5	The applicant is marked as rejected, and the status is updated in the system.	The applicant was marked as rejected, and the status was updated correctly in the system.	Pass
TC8_6	TD8_6	A message is sent to the applicant's mailbox informing them of the hiring or rejection decision	A message was sent to the applicant's mailbox informing them of the hiring or rejection decision, as expected.	Pass
TC8_7	TD8_7	The status of the application is updated to reflect the employer's decision (e.g., Shortlisted, Hired, Rejected).	The application status was updated in the applicant's account to reflect the employer's decision.	Pass

6.5.1 Test Data

User Acceptance Testing (UAT) is the final phase of the testing process, where real users interact with the system to verify that it meets their needs and expectations. The primary goal of UAT is to ensure that the UTeM Job Portal system is fully functional and ready for deployment. During this phase, feedback from users is gathered to identify any remaining issues or areas for improvement. The link for the User Acceptance Testing, google form: https://forms.gle/hNV9GzTSFWG4CXZQ6

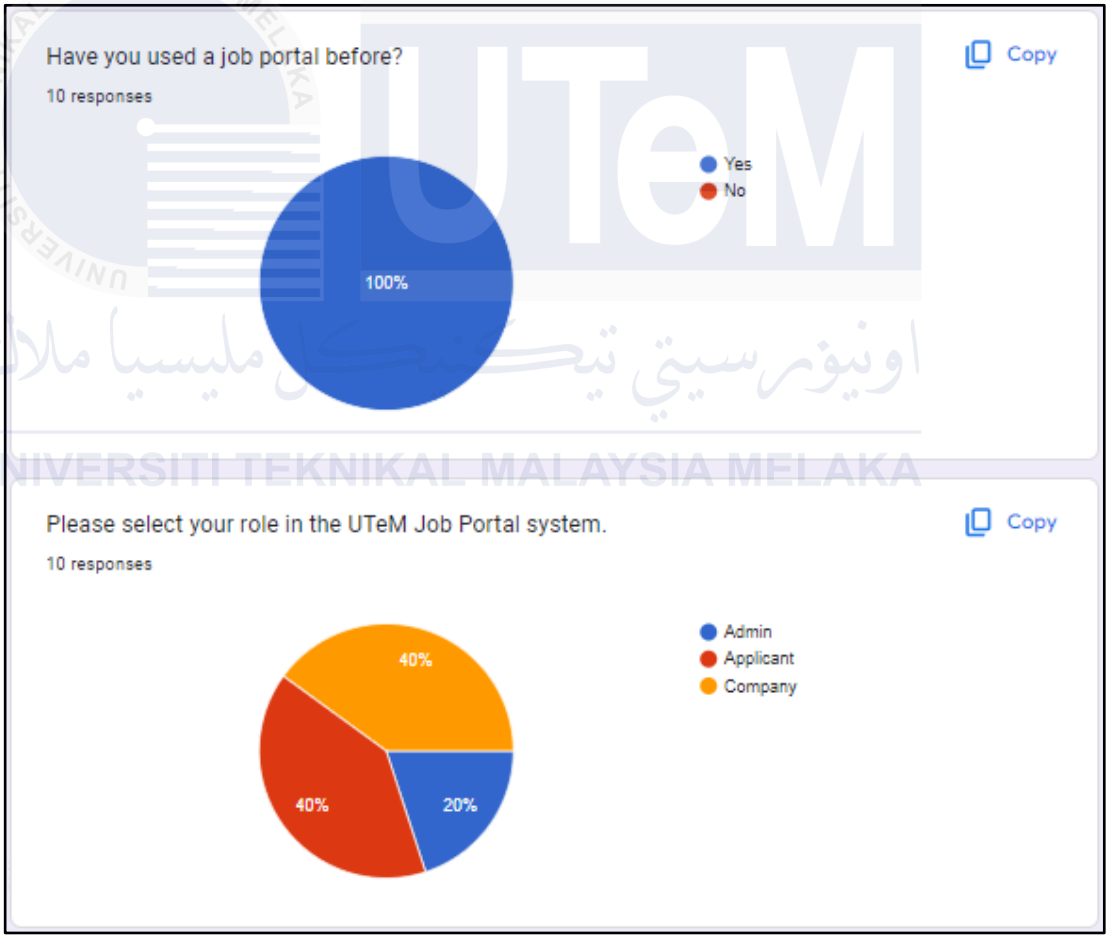


Figure 6.2: General Questions

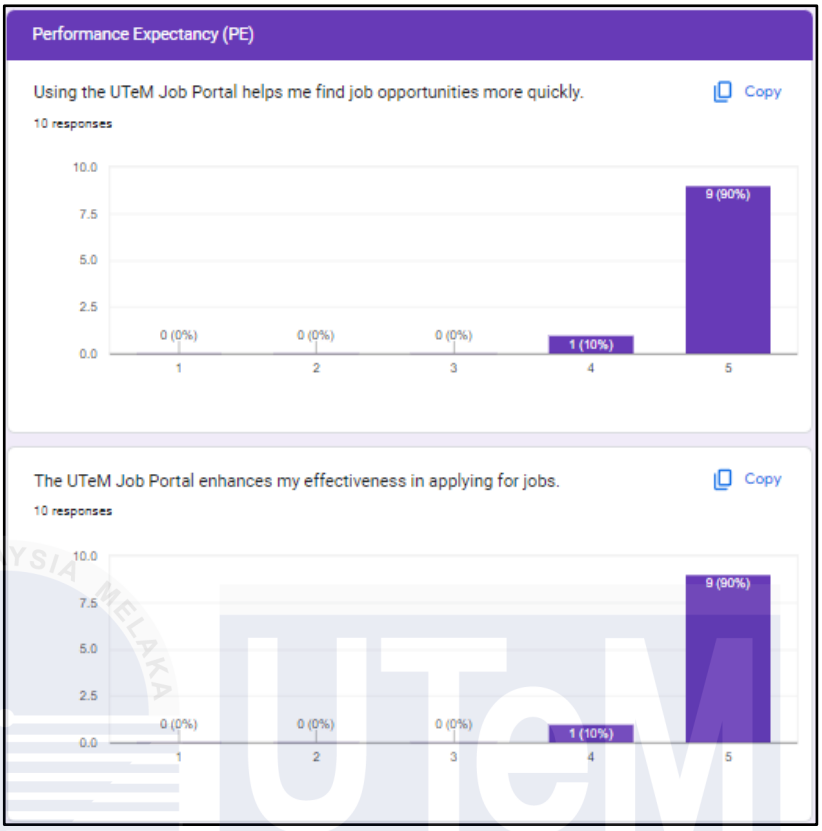


Figure 6.3: Performance Expectancy 1

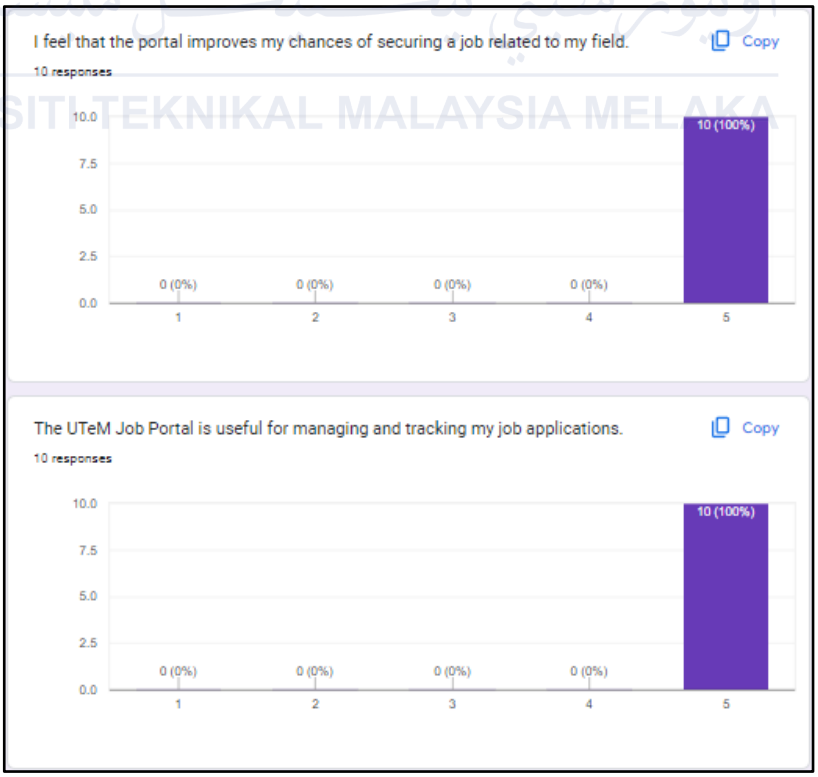


Figure 6.4: Performance Expectancy 2

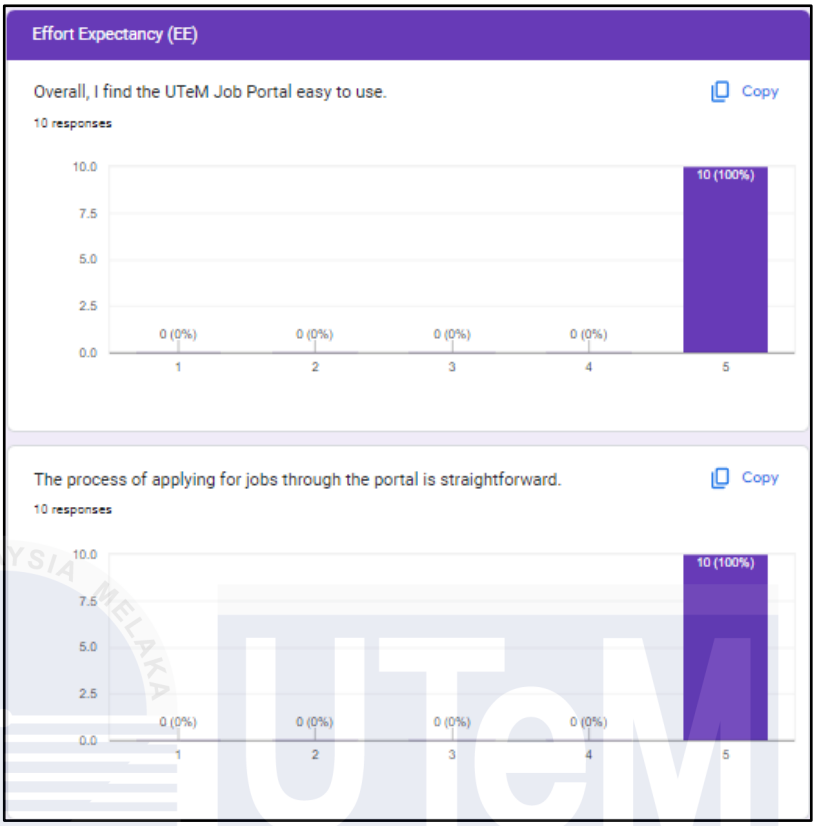


Figure 6.5: Effort Expectancy 1

Figure 6.6: Effort Expectancy 2

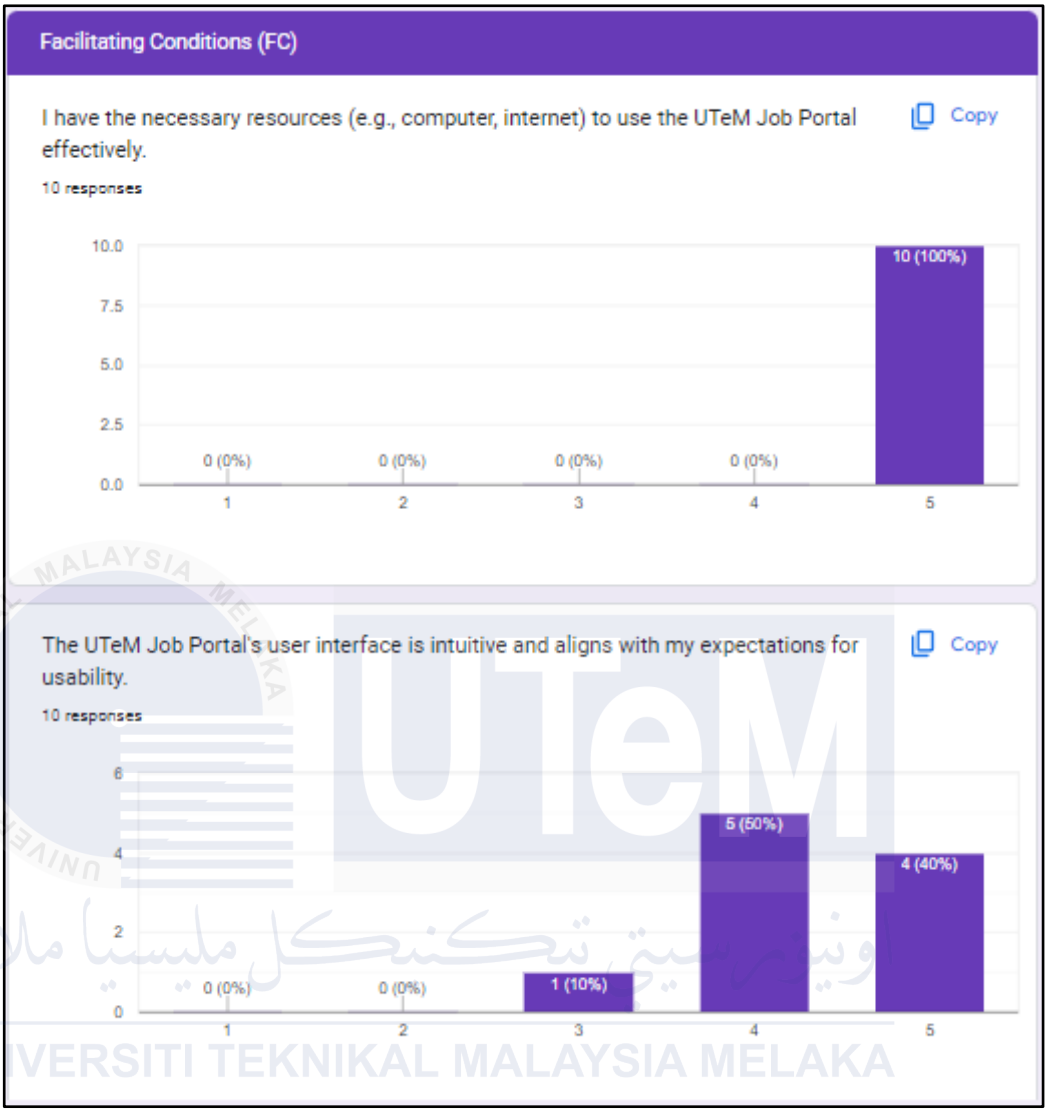


Figure 6.7: Facilitating Conditions 1

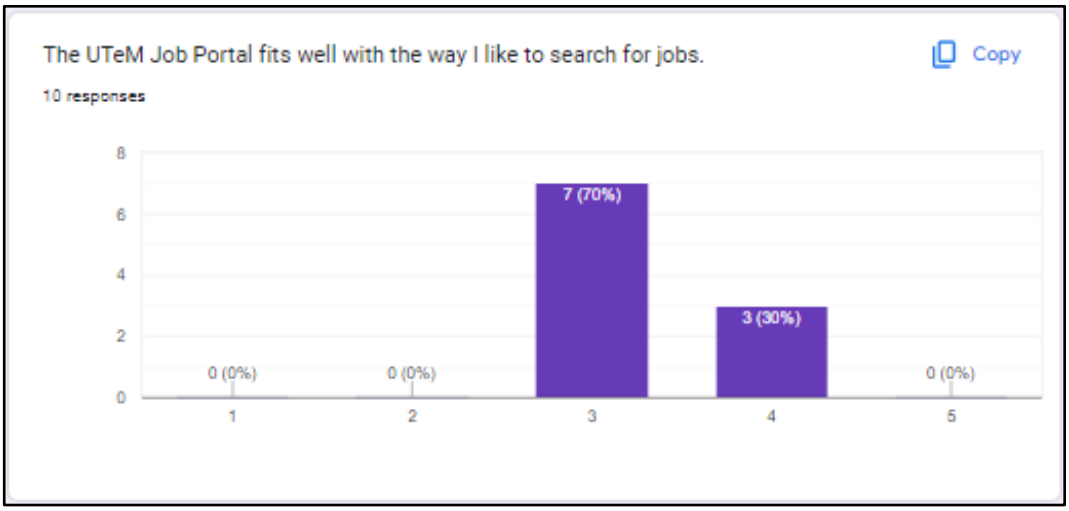


Figure 6.8: Facilitating Conditions 2

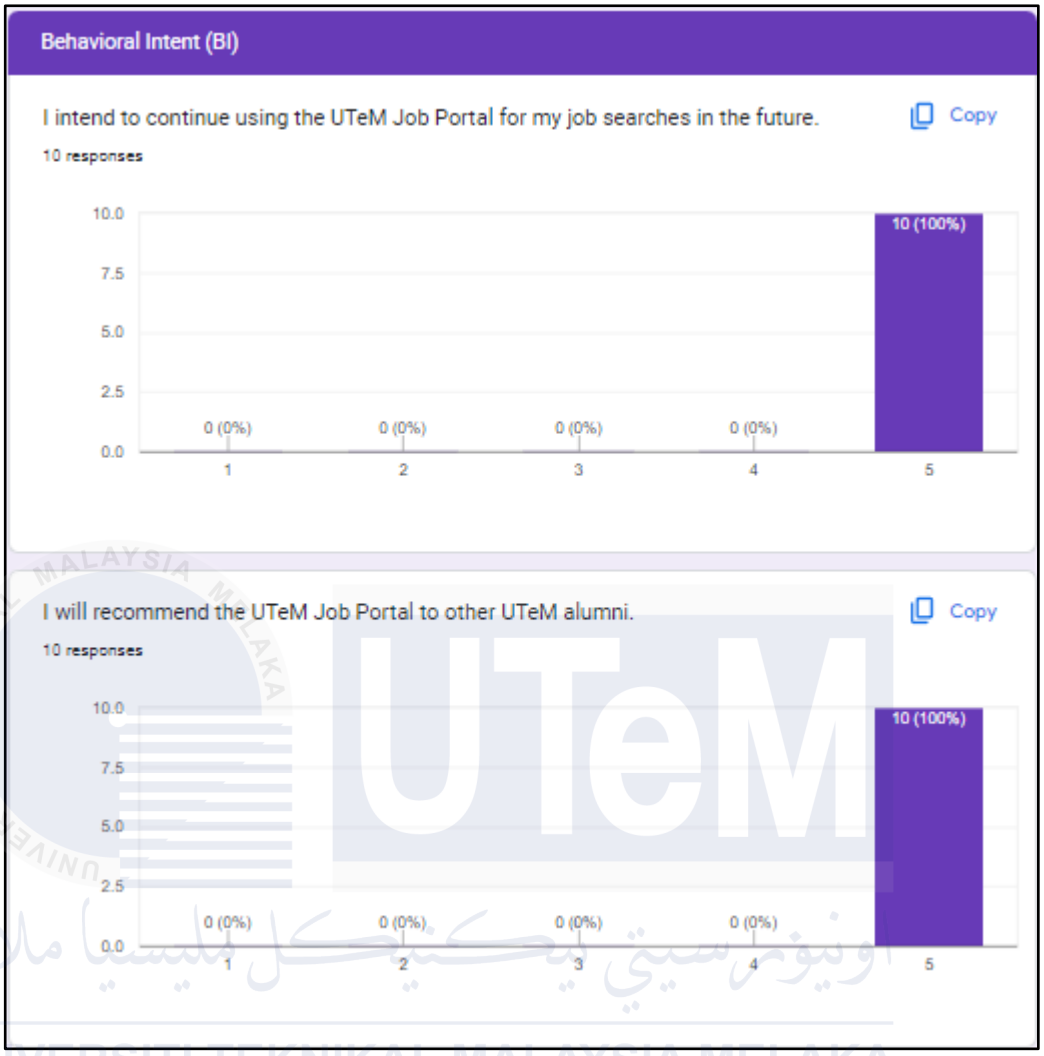


Figure 6.9: Behavioral Intent 1

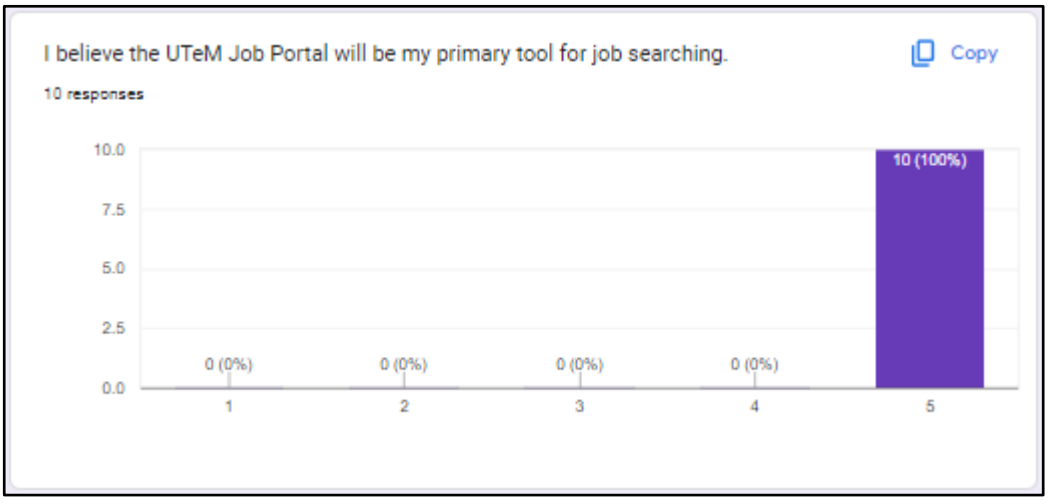


Figure 6.10: Behavioral Intent 2

6.6 Conclusion

In conclusion, system testing plays a vital role in the Software Development Life Cycle (SDLC) of the UTeM Job Portal System. This phase is essential for identifying and rectifying any bugs or defects to ensure that the final product is as error-free as possible for the end-users. The testing process helps to enhance the system's reliability and performance by allowing for the resolution of issues before the system goes live.

Moreover, the test plan serves as a critical tool for tracking the progress of the development process and ensuring that the project milestones are met. The test strategy, in turn, contributes to the smooth execution of the testing activities, reducing potential risks associated with the deployment. By identifying errors early, the testing process helps avoid costly last-minute fixes and maintenance issues that could arise if the system were deployed with unresolved defects.

Finally, the following chapter will discuss the overall strengths and areas for improvement of the UTeM Job Portal System. The findings from the testing phase will provide valuable insights for future enhancements, ensuring that the system meets the needs of its users and continues to deliver a high level of satisfaction.

CHAPTER 7: PROJECT CONCLUSION

7.1 Introduction

This chapter provides a comprehensive conclusion to the UTeM Job Portal project, summarizing the key findings and outcomes of the system's development and testing phases. The purpose of this chapter is to reflect on the strengths and weaknesses observed during the project, propose enhancements for future development, and highlight the contributions this project has made to UTeM, its alumni, and potential employers. By reviewing the entire project, this chapter will also determine whether the objectives set out at the beginning of the project have been successfully achieved and discuss the overall impact and future potential of the UTeM Job Portal.

7.2 Observation on Weaknesses and Strengths

Throughout the development and implementation of the UTeM Job Portal, several strengths and weaknesses were identified. These observations provide valuable insights into the system’s performance and the user experience, which are crucial for understanding how the project has met its objectives and where improvements can be made.

Table 7.1: Weaknesses and Strengths

NUMBER	WEAKNESSES
1	One of the significant weaknesses identified in the UTeM Job Portal is the limited functionality of the search filter. Currently, users can only filter their job searches by one criterion at a time. For example, a user can filter jobs by state, but they cannot simultaneously filter by both state and job category. This limitation makes it more challenging for users to narrow down their job search effectively, potentially leading to a less efficient and satisfying user experience. Enhancing the search filter to allow for multi-criteria filtering would significantly improve the usability of the system.
2	Another area that requires improvement is the notification system. As it stands, the portal does not provide notifications for updates related to job applications. This means that applicants might not be aware when there is a change in their application status, and companies may not immediately realize when they have received a new application. This lack of timely communication could result in missed opportunities and delays in the recruitment process. Implementing a notification system would ensure that all users stay informed about important updates.
3	The design and layout of the UTeM Job Portal have also been flagged as needing significant improvement. User feedback indicates that the current design is not visually appealing and can be cumbersome to navigate. A more modern, intuitive design would enhance the overall user experience, making the portal not only functional but also pleasant to use.

Number	STRENGTHS
1	Despite these weaknesses, the UTeM Job Portal also has several strengths that have been well-received by users. One of the most notable features is the mailbox function, which facilitates communication among applicants, companies, and administrators. This feature centralizes all interactions within the platform, making it easier for users to keep track of their conversations and ensuring that all communications are securely stored within the system.
2	Another strong point of the UTeM Job Portal is the "Forgot Password" function, which provides a straightforward way for users to recover access to their accounts if they forget their passwords. This function is essential for maintaining user accessibility and ensuring that the platform remains user-friendly, even for those who might struggle with remembering their login credentials.
3	Additionally, the portal's administrative capabilities offer significant benefits, particularly for UTeM. Administrators can monitor the progress of graduates' job applications, track which companies are engaging with their alumni, and identify reputable employers. This monitoring ability not only helps safeguard graduates from potential scams or dangerous employers but also allows UTeM to maintain a high standard of career support for its alumni, reinforcing the university's commitment to their success.

7.3 Propositions for Improvement

While the UTeM Job Portal has proven to be a valuable tool for both alumni and employers, there are several areas where the system could be improved to enhance its functionality and user experience. Below are some key propositions for improvement:

1. Enhanced Search Filter Functionality

Currently, the search filter allows users to filter job listings by only one criterion at a time, such as state or job category. To improve the efficiency of the job search process, the search functionality should be expanded to support multi-criteria filtering. For instance, users should be able to filter jobs by both state and job category simultaneously, as well as by other relevant criteria such as company size, job type (e.g., full-time, part-time), and salary range. This enhancement would enable users to refine their searches more effectively, helping them find the most relevant job opportunities faster.

2. Implementation of a Notification System

One of the critical gaps in the current system is the absence of a notification feature. Implementing a robust notification system would significantly enhance user engagement and ensure timely communication. For applicants, notifications could alert them to changes in the status of their job applications, such as when an application is reviewed, shortlisted, or rejected. For companies, notifications could inform them of new applications, upcoming deadlines, or other important updates. These notifications could be delivered via email, SMS, or in-platform alerts, ensuring that users stay informed and responsive.

3. Redesign of the User Interface and Layout

User feedback has indicated that the current design and layout of the UTeM Job Portal could benefit from a significant overhaul. To make the portal more visually appealing and easier to navigate, a redesign should focus on creating a clean, modern interface with intuitive navigation paths. This could include a more organized homepage, clearly labeled menus, and a consistent color scheme that aligns with UTeM's branding. Additionally, improving the layout to ensure that critical features are more accessible and reducing clutter would contribute to a smoother user experience.

4. Introduction of Job Recommendations

To further assist users in finding suitable job opportunities, a job recommendation feature could be introduced. This feature would use algorithms to analyze users' profiles, including their qualifications, past job searches, and application history, to suggest relevant job openings. Personalized job recommendations would not only save users time but also increase their chances of finding roles that match their skills and career goals.

5. Integration of Advanced Employer Tools

For employers, the addition of more advanced tools could make the UTeM Job Portal an even more attractive platform for posting jobs and managing candidates. Features such as an applicant tracking system (ATS), detailed analytics on job postings, and the ability to create and manage talent pools would provide employers with better control over their recruitment processes. These tools would help companies streamline their hiring workflows, making it easier to identify and recruit the best candidates from UTeM.

6. Implementation of Backup and Recovery Solutions

To ensure data integrity and protect against data loss, implementing a robust backup and recovery system is essential. The backup system should automatically save copies of the database and critical files at regular intervals to a secure location, allowing for quick restoration in the event of data corruption, accidental deletion, or system failure. A well-structured recovery plan would ensure that the system can be quickly restored to a previous state with minimal downtime, protecting both the data of users and the overall functionality of the UTeM Job Portal.

7.4 Project Contribution

The UTeM Job Portal project has made significant contributions to the university, its alumni, and the wider community by addressing critical needs in job placement and career development. Below are the key contributions of this project:

1. Contribution to the University

The UTeM Job Portal enhances the university's ability to support its graduates by providing a centralized platform for job searching and application management. Through this portal, UTeM can track the employment progress of its alumni, gaining valuable insights into their career paths and the effectiveness of the university's educational programs. Additionally, the portal allows UTeM to maintain strong relationships with employers, ensuring that graduates have access to quality job opportunities. The system's ability to monitor job application trends and outcomes also provides UTeM with data that can be used to improve future curricula and career services.

2. Contribution to Alumni

For UTeM alumni, the Job Portal serves as an essential resource for career advancement. By providing a streamlined, user-friendly platform for job searching, applying, and interacting with potential employers, the portal simplifies the job-seeking process. Alumni benefit from having access to a wide range of job listings tailored to their qualifications and career goals, as well as the ability to communicate directly with employers through the integrated mailbox function. The portal's security features also protect alumni from potential scams, ensuring a safe job-hunting environment.

3. Contribution to Employers

The UTeM Job Portal offers employers a direct channel to connect with qualified UTeM graduates. This contribution is particularly valuable for companies looking to hire fresh talent with specific educational backgrounds. Employers can easily post job vacancies, manage applications, and interact with candidates through the portal, streamlining their recruitment processes. The portal also allows employers to identify and build relationships with UTeM as a source of top-tier talent, potentially leading to partnerships, internships, and other collaborative opportunities.

7.5 Conclusion

In conclusion, the UTeM Job Portal project has successfully achieved its primary objectives, creating a centralized platform that significantly enhances the job-seeking experience for UTeM alumni while providing valuable tools for employers and university administrators. Through rigorous system testing and iterative improvements, the portal has been refined to meet the specific needs of its users, offering a secure, user-friendly, and efficient environment for job searching, application submission, and candidate recruitment.

The project has addressed critical issues identified during the analysis phase, such as the need for better communication between alumni and employers, the importance of data security, and the requirement for a system that UTeM can use to monitor and support its graduates' career progress. By providing solutions to these challenges, the UTeM Job Portal stands as a testament to the university's commitment to supporting its alumni and maintaining strong ties with employers. Looking forward, there are clear opportunities for further enhancement, as outlined in the propositions for improvement. By continuing to evolve and adapt the system to the changing needs of its users, the UTeM Job Portal has the potential to become an indispensable tool for career development and job placement within the university community.

Ultimately, this project not only meets its set objectives but also lays the groundwork for ongoing innovation and improvement, ensuring that UTeM remains at the forefront of supporting its graduates' professional success.

REFERENCES

Hamilton, T. (2024, March 23). *Database (Data) Testing Tutorial*. Guru99. https://www.guru99.com/data-testing.html

Pinjari, M., De, N., Kokne, R., Siddiqui, A., & Chitre, D. (2019). ONLINE JOB PORTAL. *International Research Journal of Engineering and Technology (IRJET)*, 156. https://d1wqtxts1xzle7.cloudfront.net/59736495/IRJET-V6I43320190615-44144-cgrnqq-libre.pdf?1560591161=&response-content-disposition=inline%3B+filename%3DIRJET_ONLINE_JOB_PORTAL.pdf&Expires=1725202853&Signature=SE~-ZaxhC07MfbFOFuvW85erCgExZB7BPCSmpFNaJBeV8vGQn~F6YEmgUpglM2w5BSx-SdhTrKeqkxbn6A6ZGx77k5xjzKrG-I66gTn6CYnxZGDeB3f91shZOJ9JHKdRYVdrnppnt1ZXLXZXjnz5QcAENos-USF~IZ9n5aH4dYfnHdHl8dxY2s3jcUTtQCUP8KvekamI7PIYgm9tQtczYEsIiNoUkVX19J-0UgYxlkfPshD6dYnTZQ558ZRp2N0fUsvOEYBatrXIFU0uMElwHvkLofBhJGSULaAXpDUhqCX2YZgBfQCWj95qDEXkwdvhOmvcAcVIBb~bIY2pt6Kh2A__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

Prodhan, M. M. R., & Saha, B. K. (2017, March 9). *ONLINE JOB PORTAL*. http://182.160.117.219:8080/xmlui/handle/123456789/205

Hamilton, T. (2024b, April 12). *What is Integration Testing? (Example)*. Guru99. https://www.guru99.com/integration-testing.html

Admin. (2023, July 27). *4 Levels of Software Testing and Difference Between them - Bút Chì Xanh*. Tài Chính - Kinh Tế. https://butchixanh.edu.vn/4-levels-of-software-testing-and-difference-between-them-k8mpsx2f/