Individual Assignment

Enterprise Programming for Distributed Application (052024-KGT)

Part 2

HAND-IN DATE: 31st August 2024

STUDENT NAME: Aishath Shafa Saudulla

STUDENT TP: TP066434

LECTURER NAME: Mr. Kau Guan Kiat

INTAKE: APD3F2405SE

Table of Contents

Introduction

This project consists of an employment system for the company APU. The primary objective of this project is to create a robust, scalable, and user-friendly employment management platform that can handle various tasks such as job postings, applications, user management, and feedback processing. The system is intended to streamline the employment process, making it easier for employers to post job vacancies, employees to apply for jobs, and administrators to manage the overall system efficiently. Some key features found in the system include:

- **User Management:** The system supports multiple user roles (Admin, Employer, Employee), each with specific functionalities and access levels. This allows for a tailored user experience depending on the user's role within the system.
- **Job Posting and Application:** Employers can post job vacancies, and employees can apply for these jobs through a streamlined interface. The system also tracks applications, allowing both employers and employees to manage and review job applications effectively.
- **Feedback and Notifications:** The platform enables employers to provide feedback on employees, which can be managed by administrators. Notifications are also sent to users to keep them informed about important events, such as job application status updates or feedback received.
- **Security and Data Integrity:** The system ensures that all user data is securely stored and managed. Role-based access control is implemented to ensure that sensitive information is accessible only to authorized users.

The system utilizes a three-tier architecture to ensure modularity and scalability. The system was made for 3 main user roles which are Admin, Employer and Employees. Each contain their own different functionalities and access privileges. The project is implemented using a three-tier architecture:

1. **Presentation Tier:**
   - **JSP Pages**: These pages serve as the user interface, allowing users to interact with the system. Different pages are designed for different roles (Admin, Employer, Employee), and include features like login, registration, and dashboard views.
   - **Servlets**: These act as controllers that process user requests from the JSP pages, invoke the appropriate business logic, and return responses back to the user.
2. **Business Tier:**
   - **Facade Classes**: These classes encapsulate the business logic of the application. They interact with the entity classes to perform CRUD (Create, Read, Update, Delete) operations.
   - **Entity Classes**: These classes represent the data model, corresponding to database tables. They hold the attributes and relationships that define how data is stored and managed.
3. **Database Tier:**
   - **Database**: This tier consists of the actual database where all the data is stored. Entity classes map directly to database tables, ensuring data is persisted and retrieved efficiently.

System Architecture

The diagram below provides a visual representation of the interaction between the three tiers of the system:
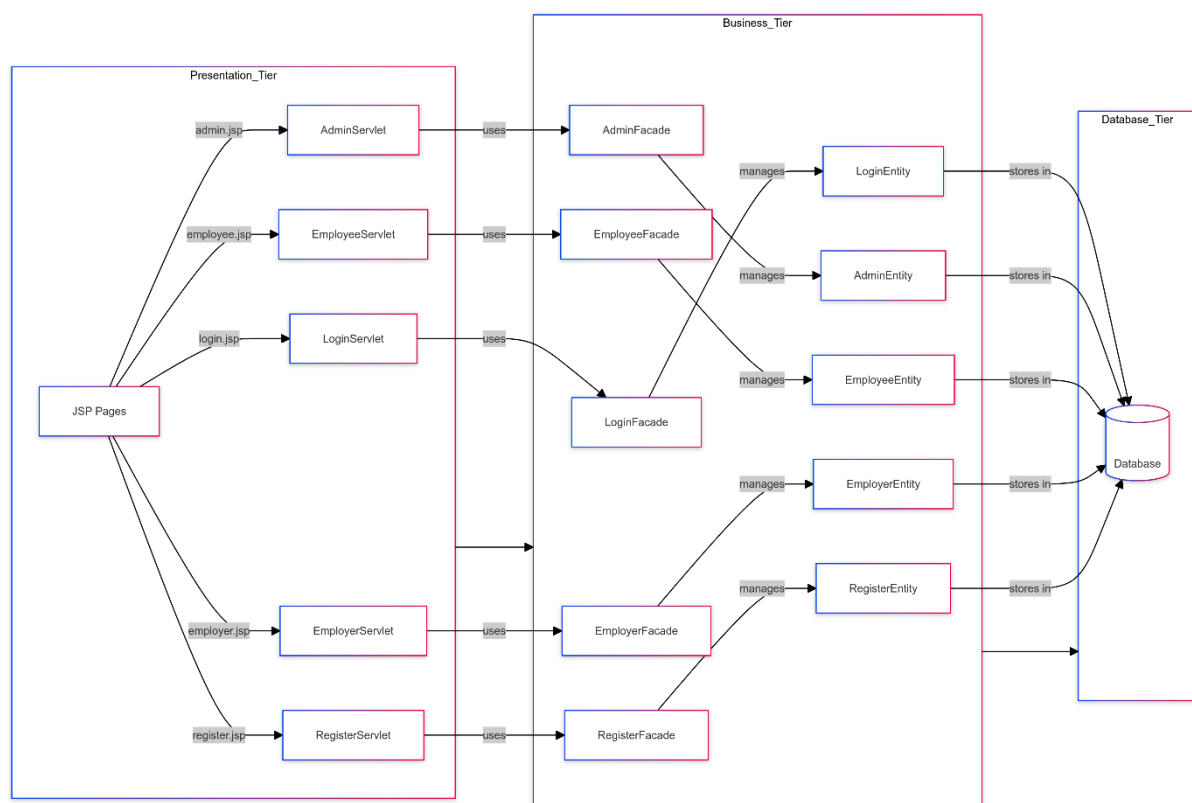
1. **Presentation Tier**:
   - The JSP pages are the entry point for user interactions. For example, a user might visit login.jsp to access the system.
   - The JSP pages communicate with the corresponding servlets (e.g., LoginServlet) that handle user requests.
2. **Business Tier**:
   - The servlets interact with facade classes in the Business Tier. These facades handle the business logic and interact with the entity classes.
   - Each facade class (e.g., LoginFacade) corresponds to a specific entity (e.g., LoginEntity), encapsulating operations like validating user credentials, retrieving user information, or managing sessions.
3. **Database Tier**:
   - The entity classes in the Business Tier interact with the Database Tier. They represent the structure of the database tables (e.g., AdminEntity, EmployerEntity), ensuring data is stored, updated, and retrieved as needed.
   - The diagram shows how these entities are connected to the database, ensuring data integrity and consistency across the application.
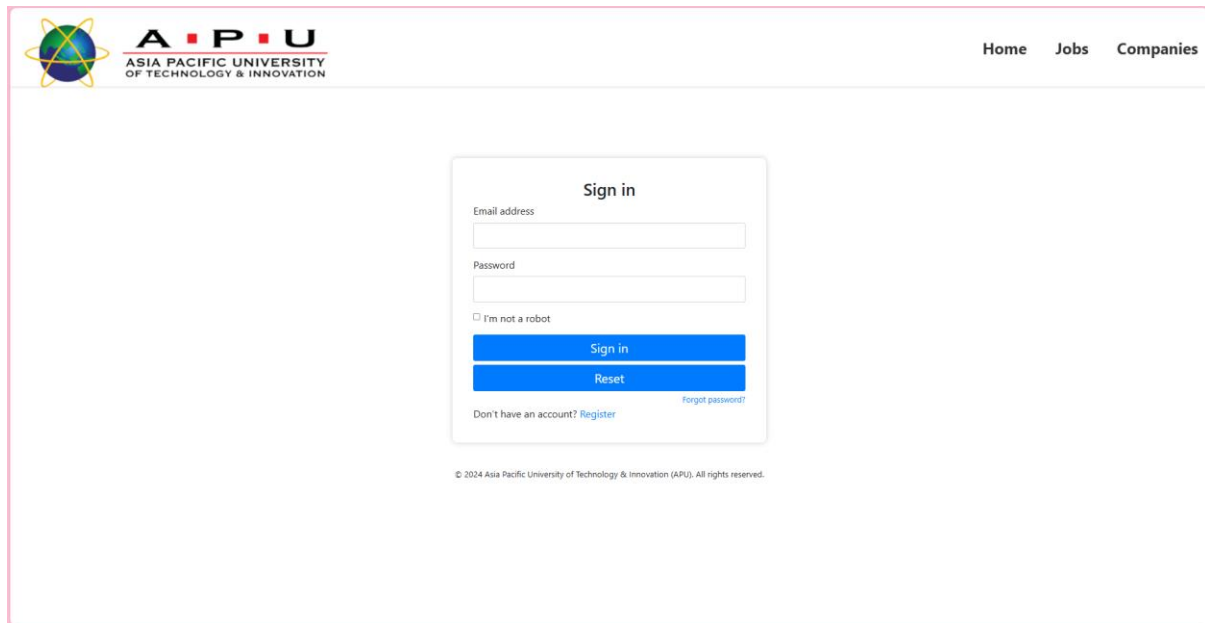
User Manual
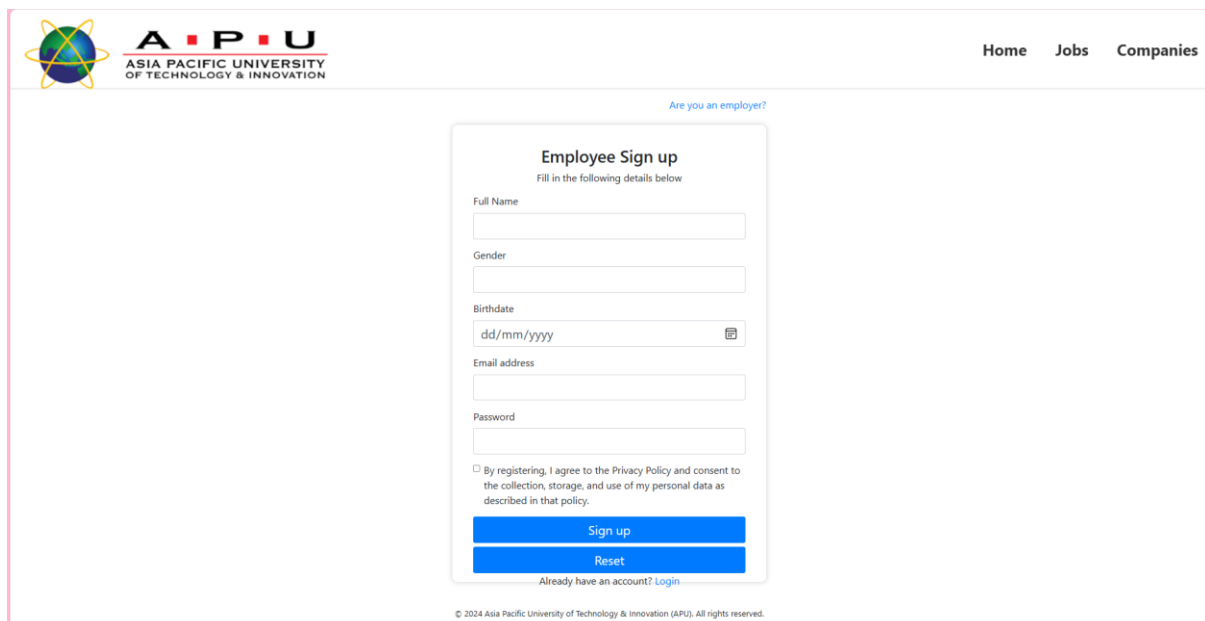
1. First Page – Login Page

The first page it loads is the login page where the user will be able to click on Register if they do not already have an existing account.

2. Registration for Employee

On the registration for employee page the user gets the option to either decide whether they want to continue as an employee or sign up to be an Employer or Corporate Customer.
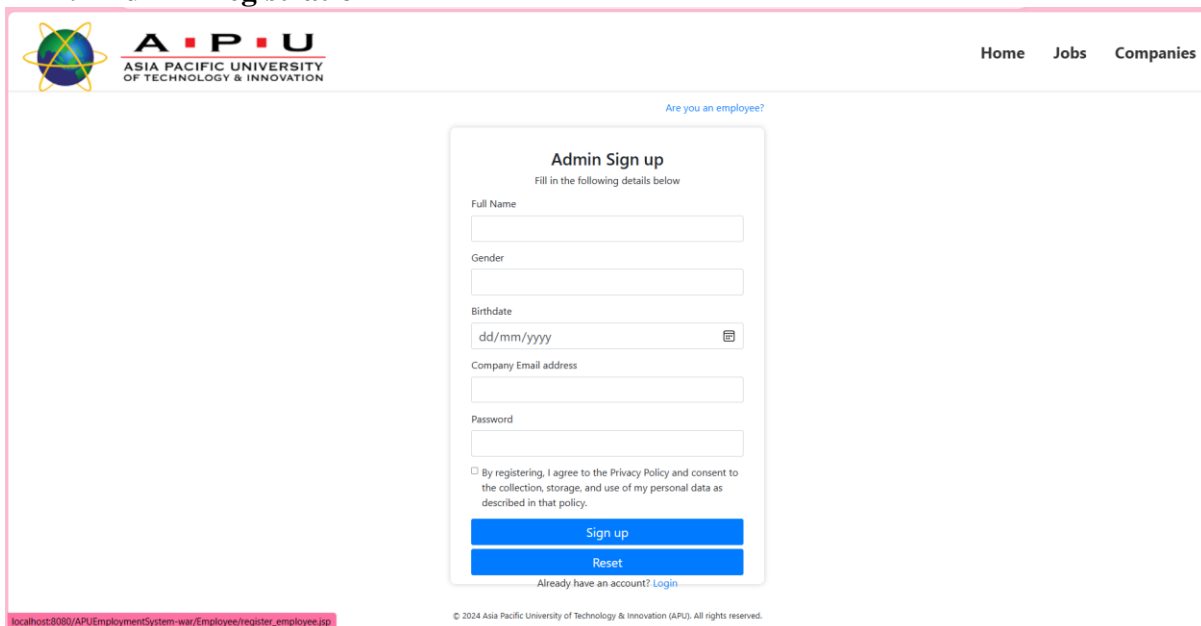
3. Employer Registration

Here the user can register to be an employee or decide whether they need an Admin account.

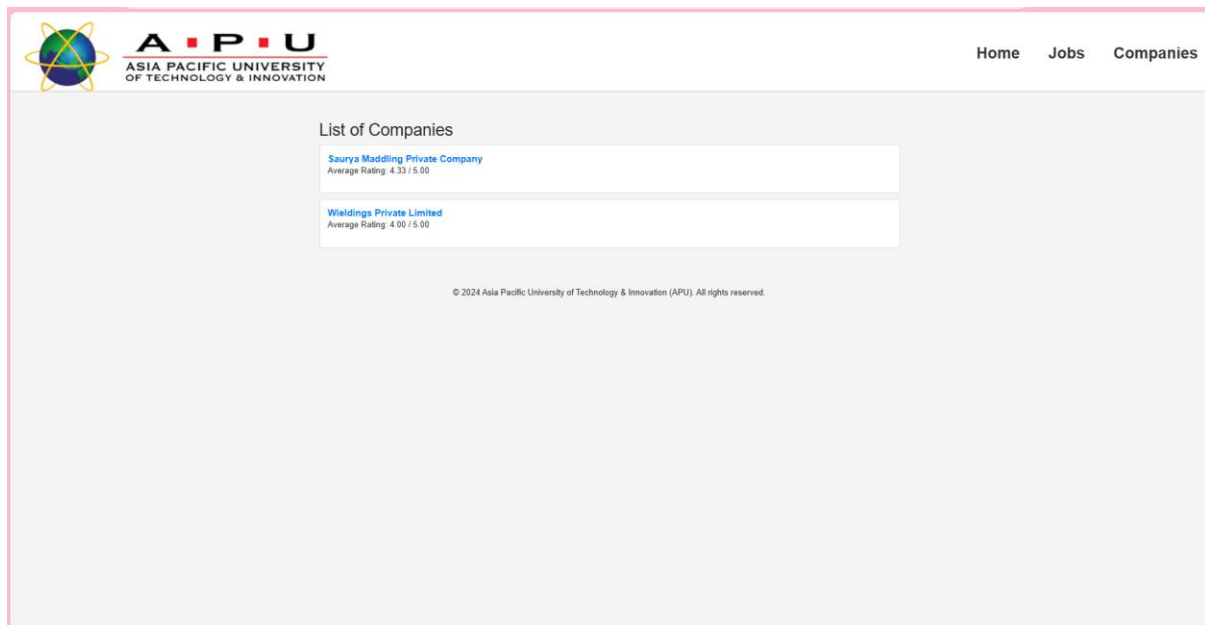### 4. Admin Registration

The admin can register for a new account here by filling in the blanks and submitting.
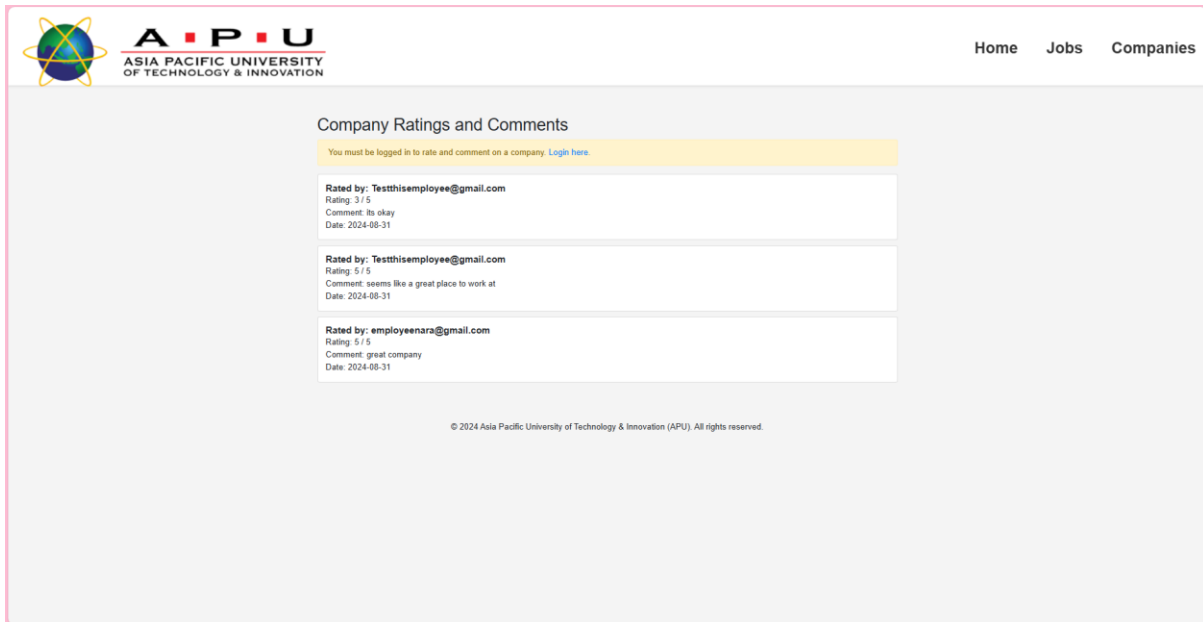
5. Jobs Page
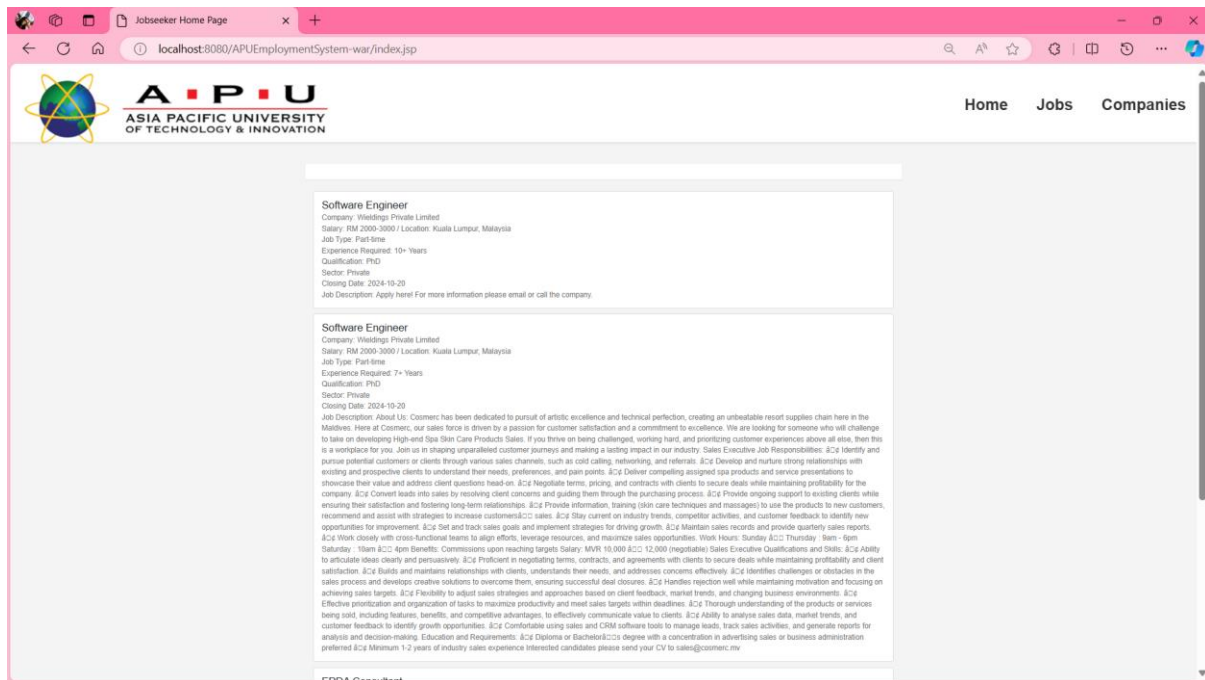
Before logging in this the view all users can see of the Job listings.

6. Companies Page

Here the user can view the list of companies who have posted vacancies and their general ratings.
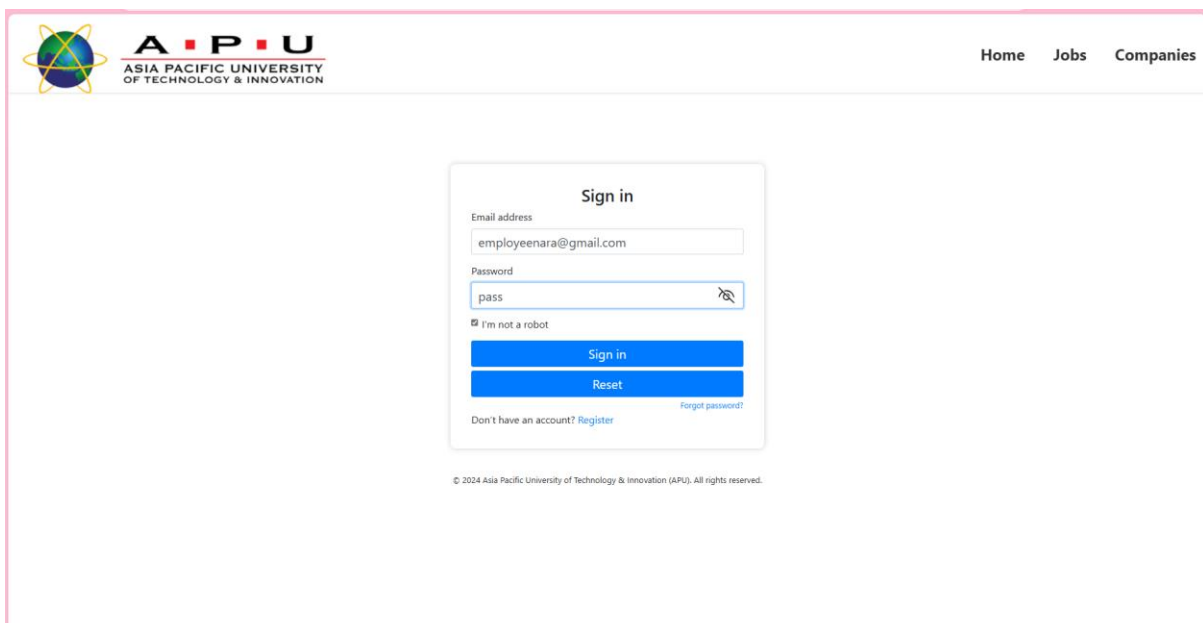
7. Company Ratings and Comments

Once the user clicks on a company name, it will show the previous comments and ratings left by other users. If the user wants to post their own ratings and comments they will have to first log in to proceed.

8. Sign in

The user has to sign in to their account through here to access the features.

9. Jobseeker Dashboard

Once the jobseeker is logged in they are able to see their dashboard consisting of the option to either manage their profile, view their application history or view their notifications. To apply for jobs they have to click on the Jobs of the header.
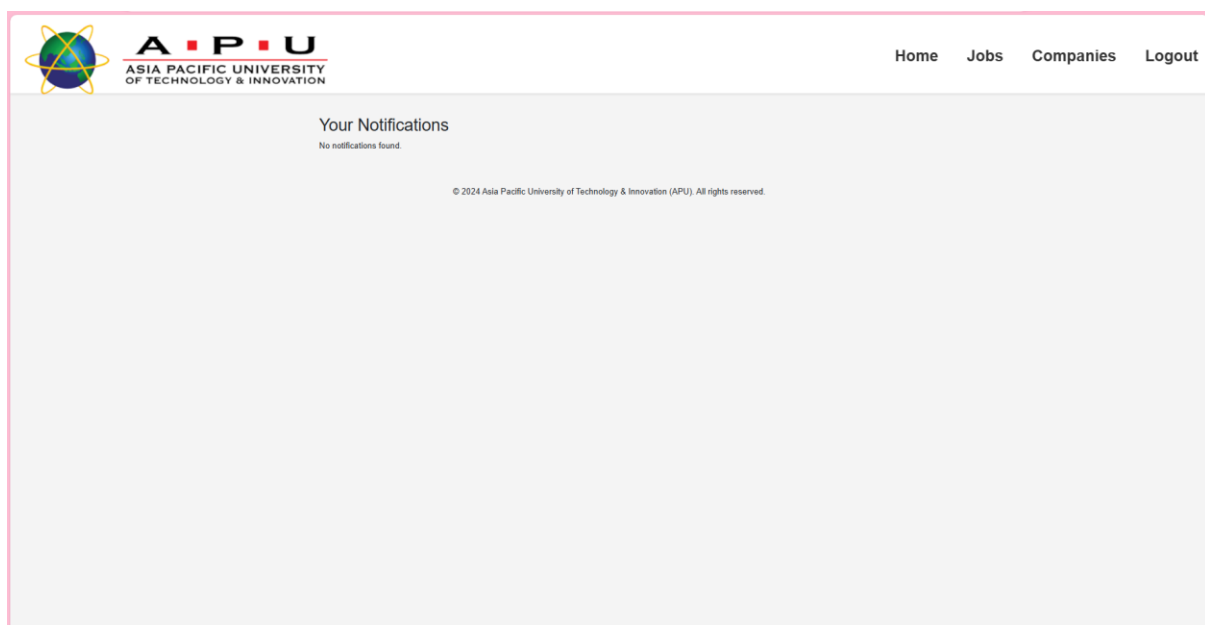
10. Profile Edit

All users have the options to edit their profile details here.

11. Application History
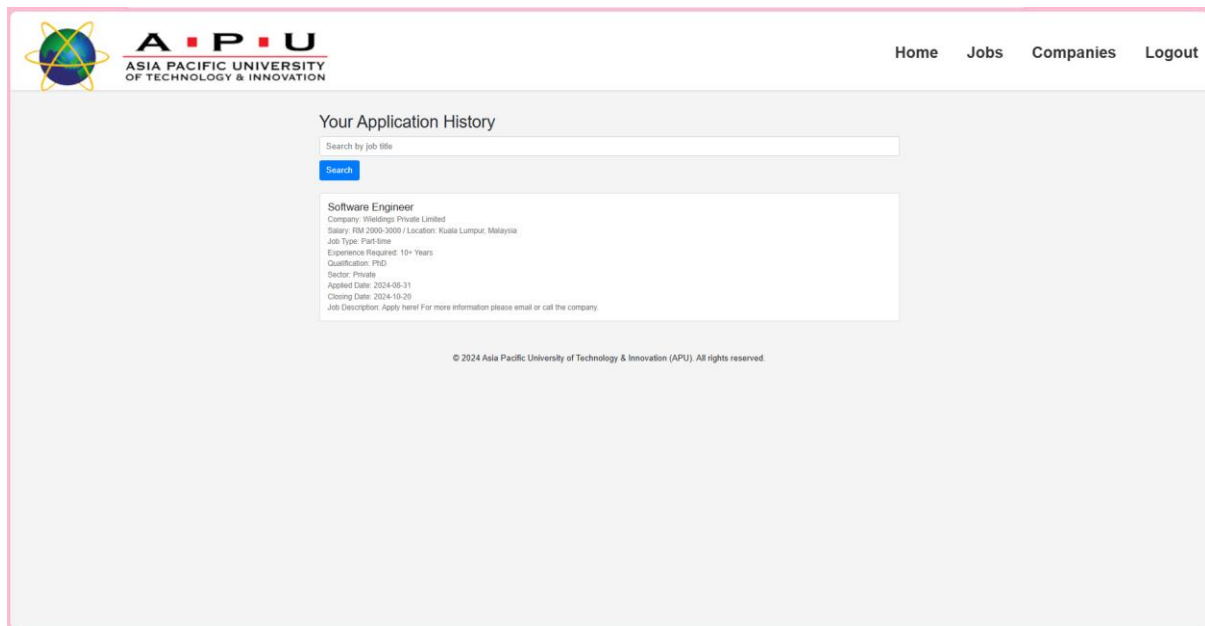
The jobseeker is able to view their past applied job listings here.

12. Notifications

The jobseeker will be able to view all of their notifications here such as warnings, when they get shortlisted or when they get the job confirmation as well.

13. Job Applications

The jobseeker is able to now have the option to apply for jobs and review them as well.

14. Rate the Job Listing

The jobseeker is able to submit ratings for both the job listings and the company as well after logging in.

15. Admin Dashboard
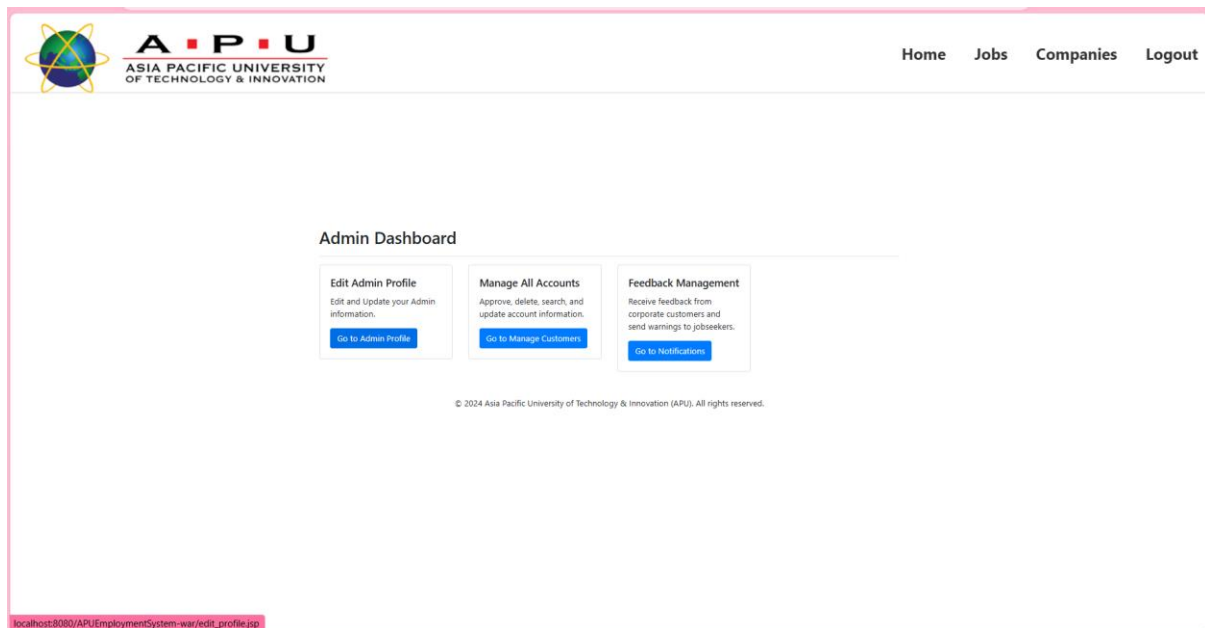
Once the admin logs into their account they are taken to the admin dashboard where they have the option to edit their admin profile, manage all accounts, and manage feedbacks and warnings.

16. Search & Manage Accounts

The admin has the ability to search for any account and edit, delete, or even add new information here. Once the admin is done with their work, they can go back home from the home button at the top.

17. Feedback Management

The admin is able to view all of the complaints made by the corporate customers here and issue warnings to the employee's. The admin is also able to view all of the warnings they have previously sent out. Once a warning has been issues 3 times, the employee account is automatically deleted.

18. Employer / Corporate Customer Dashboard

Once the corporate customer logs in from their account they are able to see their dashboard where it has the option to either manage their profile, post a new vacancy, shortlist applicants, view post histories and send feedback to admin.

19. Post New Vacancy

The customer is able to post a new job vacancy here by filling out the required details and submitting it.

20. Applicants List

The customer is able to view the list of applicants who applied to which job and shortlist them or confirm the job for them here.

21. Posting History

.

The customer is able to view their previous listings and edit or delete them here

22. Submit feedback or Complaints

The customer is able to submit feedbacks or complaints to the admin by filling out the form on this page.

Technical Explanation

1. Overview of Project Structure

1.1 **Project Modules:**

- **APUEmploymentSystem-ejb**: Contains the enterprise JavaBeans (EJB) module.
- **APUEmploymentSystem-war**: Contains the web application module.

1.2 **Main Folders:**

- **Source Packages**: Contains Java classes, including models and controllers.
- **Web Pages**: Contains JSP files for different user roles and other web-related resources.
- **Libraries and Configuration Files**: Contains dependencies and configuration settings for the project.

2. Design of Web Components

2.1 **JSPs (JavaServer Pages):**

2.1.1 **Admin Directory**:

- **admin_dashboard.jsp**: Renders the admin dashboard interface.
- **admin_feedback_management.jsp**: Manages feedback from users.

2.1.2 **Employee Directory**:

- **application_history.jsp**: Displays the job application history for an employee.
- **employee_dashboard.jsp**: Provides an overview of the employee's dashboard.

2.1.3 **Employer Directory**:

- **post_vacancy.jsp**: Allows employers to post new job vacancies.
- **employer_dashboard.jsp**: Provides employers with access to manage job postings and view applicants.

2.2 **Servlets (Source Packages > controller):**

2.2.1 **AdminFeedbackServlet.java**:

- Handles HTTP requests related to feedback management by the admin.

2.2.2 **ApplyJobServlet.java**:

- Manages the job application process for employees.

2.2.3 **PostVacancyServlet.java**:

- Allows employers to post new job vacancies.

2.3 **Other Components:**

2.3.1 **CSS**:

- **style.css**: Defines the styles and layouts used across the web pages.

2.3.2 **JavaScript (JS)**:

- **app.js**: Contains JavaScript code that adds interactivity to the web pages.

2.3.3 **Images**:

- **logo.png**: The logo image used in the application.

3. Design of Business Tier (EJBs)

3.1 **Entity Classes (model package):**

3.1.1 **Applicants.java**:

- Represents the Applicants table in the database.

3.1.2 **Vacancy.java**:

- Represents job vacancies, with fields like job title, description, and associated employer.

3.2 **Facade Classes (model package):**

3.2.1 **ApplicantsFacade.java**:

- Provides CRUD operations related to Applicants.

3.2.2 **VacancyFacade.java**:

- Manages operations related to job vacancies.

4. System Architecture and Interconnection Among Tiers

4.1 **Three-Tier Architecture:**

   4.1.1 **Presentation Tier**:

- **Components**: JSP files, CSS, and JavaScript.
- **Role**: Handles the user interface and presentation logic.

   4.1.2 **Business Tier**:

- **Components**: EJBs, Facade classes.
- **Role**: Manages business logic and processes data.

   4.1.3 **Database Tier**:

- **Components**: Entity classes representing database tables.
- **Role**: Stores and manages application data.

5. Database Design

5.1 Entities and Tables:

5.1.1 Applicants Table:

#	ID	NAME	BIRTHDATE	APPLIED_DATE	GENDER	EMAIL	JOBID
1	752	Test This Employee	2000-03-01	2024-08-31	Male	Testthisemployee@gmail.com	101
2	1301	Nara Smith Lucky	2002-02-01	2024-08-31	Female	employeenara@gmail.com	1
3	1401	Nara Smith Lucky	2002-02-01	2024-08-31	Female	employeenara@gmail.com	151
4	251	Test This Employee	2000-03-01	2024-08-30	Male	Testthisemployee@gmail.com	151
5	252	Test This Employee	2000-03-01	2024-08-30	Male	Testthisemployee@gmail.com	101

- **ID**: Primary key, unique identifier for each applicant.
- **NAME**: The full name of the applicant.
- **BIRTHDATE**: The applicant's date of birth.
- **APPLIED_DATE**: The date when the applicant submitted their application.
- **GENDER**: The gender of the applicant.
- **EMAIL**: The applicant's email address.
- **JOBID**: Foreign key linking to the `VACANCY` table, representing the job the applicant applied for.

5.1.2 Feedback Table:

#	ID	FROMEMPLOYER	ABOUTEMPLOYEE	FEEDBACKDATE	FEEDBACKTYPE	MESSAGE	STATUS
1	851	asiyathmalakaibrahimmanik@gmail.com	Testthisemployee@gmail.com	2024-08-31	Complaint	This user keeps adding false ratings.	Pending
2	901	asiyathmalakaibrahimmanik@gmail.com	Testthisemployee@gmail.com	2024-08-31	Complaint	This user won't stop their false ratings.	Pending

- **ID**: Primary key, unique identifier for each feedback entry.
- **FROMEMPLOYER**: The name or ID of the employer providing feedback.
- **ABOUTEMPLOYEE**: The name or ID of the employee the feedback is about.
- **FEEDBACKDATE**: The date when the feedback was given.
- **FEEDBACKTYPE**: The type or category of the feedback.
- **MESSAGE**: The feedback message or content.
- **STATUS**: The current status of the feedback (e.g., pending, resolved).
- **ADMINRESPONSE**: Any response or action taken by the admin regarding the feedback.

5.1.3 MyCustomer Table:

#	EMAIL	BIRTHDATE	GENDER	PASSWORD	ROLE	USERNAME
1	aishathshafa7@gmail.com	2001-03-07	Female	pass	Admin	Aishath Shafa Saudulla
2	mariyameafa@apu.com.my	1997-10-22 00:00:00.0	Female	pass	Admin	Mariyam Eafa Nasheedd
3	asiyathmalakaibrahimmanik@gmail.com	2012-10-14	Female	pass	Employer	Asiyath Malaka Ibrahim Mar
4	NaraSmith@gmail.com	2002-02-01	Female	defaultPassword	Staff	Nara Smith Lucky
5	BetaTestEmployee@gmail.com	2001-03-07	Male	pass	Employee	Beta Testing Unit
6	TesterEmployee@gmail.com	2001-05-02	Male	pass	Employee	Tester Employee
7	TesterStaff@gmail.com	1995-02-11	Male	defaultPassword	Staff	Tester Staff
8	Madilyn@gmail.com	1998-01-09	Female	defaultPassword	Employee	Employee Madilyn
9	IsmailHaisham@gmail.com	2002-09-01	Male	defaultPassword	Employer	Ismail Haisham
10	employeenara@gmail.com	2002-02-01 00:00:00.0	Female	pass	Employee	Nara Smith Lucky

- **EMAIL**: The email address of the customer.
- **BIRTHDATE**: The customer's date of birth.
- **GENDER**: The customer's gender.
- **PASSWORD**: The password for the customer's account.
- **ROLE**: The role of the user (e.g., admin, employer, employee).
- **USERNAME**: The username chosen by the customer.

5.1.4 **Notifications Table**:

#	ID	FROMWHO	MESSAGE	WHATDATE	TOWHO	
1	451	Employer	You have been shortlisted. You will receive an ...	2024-08-31 01:51:20.116	Testthisemployee@gmail.com	
2	452	Employer	Congratulations! You have been offered the jo...	2024-08-31 01:51:26.446	Testthisemployee@gmail.com	
3	453	Employer	You have been shortlisted. You will receive an ...	2024-08-31 01:51:51.319	Testthisemployee@gmail.com	
4	501	Employer	You have been shortlisted. You will receive an ...	2024-08-31 01:55:56.746	Testthisemployee@gmail.com	
5	502	Employer	Congratulations! You have been offered the jo...	2024-08-31 01:56:00.457	Testthisemployee@gmail.com	
6	503	Employer	Congratulations! You have been offered the jo...	2024-08-31 01:56:08.449	Testthisemployee@gmail.com	

- **ID**: Primary key, unique identifier for each notification.
- **FROMWHO**: The ID or name of the sender of the notification.
- **MESSAGE**: The content of the notification.
- **WHATDATE**: The date the notification was sent.
- **TOWHO**: The ID or name of the recipient of the notification.

5.1.5 **Ratings Table**:

#	COMMENT	TOWHO	FROMWHO	WHATDATE	RATING	JOBID	
1	its okay	Wieldings Private Limited	Testthisemployee@gmail.com	2024-08-31	4	101	
2	its average	Wieldings Private Limited	Testthisemployee@gmail.com	2024-08-31	3	1	
3	perfect listing	Wieldings Private Limited	Testthisemployee@gmail.com	2024-08-31	3	1	
4	seems like a great place to work at	Saurya Maddling Private Company	Testthisemployee@gmail.com	2024-08-31	5		
5	great!	Wieldings Private Limited	employeenara@gmail.com	2024-08-31	5	1	
6	great company	Saurya Maddling Private Company	employeenara@gmail.com	2024-08-31	5		

- **ID**: Primary key, unique identifier for each rating.
- **COMMENT**: Any comments associated with the rating.
- **TOWHO**: The ID or name of the person the rating is for.
- **FROMWHO**: The ID or name of the person giving the rating.
- **WHATDATE**: The date the rating was given.
- **RATING**: The rating value (e.g., 1-5 stars).
- **JOBID**: Foreign key linking to the VACANCY table, indicating the job related to the rating.

5.1.6 **Vacancy Table**:

#	ID	CLOSINGDATE	COMPANYNAME	EXPERIENCE	JOBTITLE	JOBTYPE	
1	1	2024-10-20	Wieldings Private Limited	10+ Years	Software Engineer	Part-time	
2	101	2024-10-20	Wieldings Private Limited	7+ Years	Software Engineer	Part-time	
3	151	2024-10-20	Saurya Maddling Private Company	8+ Years	EPDA Consultant	Full-time	

#	LOCATION	QUALIFICATION	SALARYRANGE	SECTOR	JOBDESCRIPTION	EMPLOYEREMAIL	
1	Kuala Lumpur, Malaysia	PhD	RM 2000-3000	Private	Apply here! For more information please...	asiyathmalakaibrahimmanik	
2	Kuala Lumpur, Malaysia	PhD	RM 2000-3000	Private	About Us:Cosmerc has been dedicated t...	asiyathmalakaibrahimmanik	
3	Kuala Lumpur, Malaysia	PhD	RM 8000-9000	Private	MOODHU Holidays Maldives, winner of S...	asiyathmalakaibrahimmanik	

- **ID**: Primary key, unique identifier for each job vacancy.
- **CLOSINGDATE**: The date when the job application closes.
- **COMPANYNAME**: The name of the company offering the job.
- **EXPERIENCE**: The required experience level for the job.
- **JOBTITLE**: The title of the job position.
- **JOBTYPE**: The type of job (e.g., full-time, part-time).
- **LOCATION**: The location of the job.
- **QUALIFICATION**: The required qualifications for the job.
- **SALARYRANGE**: The salary range offered for the job.
- **SECTOR**: The sector or industry of the job.
- **JOBDESCRIPTION**: A detailed description of the job responsibilities.
- **EMPLOYEREMAIL**: The email address of the employer offering the job.

5.2 **Relationships:**

  5.2.1 **One-to-Many Relationship**:

- **Between EMPLOYER and VACANCY**: One employer can post multiple vacancies.
- **Between VACANCY and APPLICANTS**: Each job can have multiple applicants.

  5.2.2 **Many to-Many Relationship**:

- **Between APPLICANTS and VACANCY**: Managed through a join table (though not directly visible here), where multiple applicants can apply for multiple jobs.

5.3 **Indexes and Foreign Keys:**

  5.3.1 **Indexes**:

- These are created on key columns (e.g., ID, JOBID) to improve the speed of data retrieval operations.

  5.3.2 **Foreign Keys**:

- Ensures referential integrity between tables (e.g., JOBID in APPLICANTS table linking to VACANCY table).

References

admin. (2020, October 24). *Jsp Mysql Search Records*. Tutusfunny - Software Development

Tutorials. https://www.tutussfunny.com/jsp-mysql-search-records/

Tutus Funny. (2020, October 23). *Jsp Mysql Search Records*. YouTube.

https://www.youtube.com/watch?v=-XhHlWV8ymw

Yuli Vasiliev. (2009). *Beginning Database-Driven Application Development in Java EE*.

Apress.