PLACEMENT PORTAL

UCS503 Software Engineering Project Report

Mid-Semester Evaluation

Submitted by:

101903694 Meenal Mehrotra

101903708 Maulika Luthra

101903710 Rashita

101953007 Ankur Singh

BE Third Year, CoE

Group Number: 3COE27

Submitted to:

Ms. Arwinder Dhillon

Computer Science and Engineering Department

TIET, Patiala

October 2021

TABLE OF CONTENTS

S.No. **Assignment** **Page No.**

1. Project Selection Phase

1.1 Software Bid

Group : 3COE27 Dated: 26-08-2021

Team Name: Team Juan

Team ID (will be assigned by Instructor): 2

Please enter the names of your Preferred Team Members.

1. You are required to form **a three to four person** teams
2. Choose your team members wisely. You will not be allowed to change teams.

Name	Roll No	Project Experience	Programming Language used
Rashita	101903710	A ReactJs component Library containing pre-built react components and custom hooks pushed to npm, contains documentation site.	JavaScript
Meenal Mehrotra	101903694	Desktop based voice assistant which can be used to do operations easily.	Python
Maulika Luthra	101903708	Portal for easy and efficient communication among teacher, student and admin.	HTML, CSS, JavaScript, NodeJs.
Ankur	101953007	Databash – portal for collection of data to train ML models.	Python

Programming Language / Environment Experience

List the languages you are most comfortable developing in, **as a team**, in your order of preference. Many of the projects involve Java or C/C++ programming.

1. JavaScript

2. HTML

3. Python

Choices of Projects:

Please select **4 projects** your team would like to work on, by order of preference: *[Write at-least one paragraph for each choice (motivation, reason for choice, feasibility analysis, etc.)]*

First Choice	**Placement Portal** – A single platform for managing internships, placements through each user's profile to manage applications, offer acceptance, checking eligibility, etc. Two portals – Admin (for placement cell) and user's portal.
Second Choice	Portfolio Website Builder – An online platform that will build portfolio website though Github API, which can be customized though various available templates and themes. Helpful for non-web developers.
Third Choice	Face Recognition attendance system – Marking attendance through scanner present inside the class. Helps avoid false attendance.
Fourth Choice	Emotion based music player – An online portal detects mood through webcam and plays relevant song. After song ends, checks your mood again and recommends a new one.

Additional Remarks/ Inputs

Please tell us about any other factors that we should take into consideration (e.g., if you really would like to work on a project for some particularly convincing reason).

This portal combats various existing problems, like manually checking selected interns should not apply again, false CGPA entries, etc. This is a one-place for students, placement cell and the companies' campus HR's to manage student's data efficiently.

2. Project Selection Phase

2.1 Use Cases

2.1.1 Use Case Diagram

Below is the use case diagram for how the actors will perform operations in the placement portal.

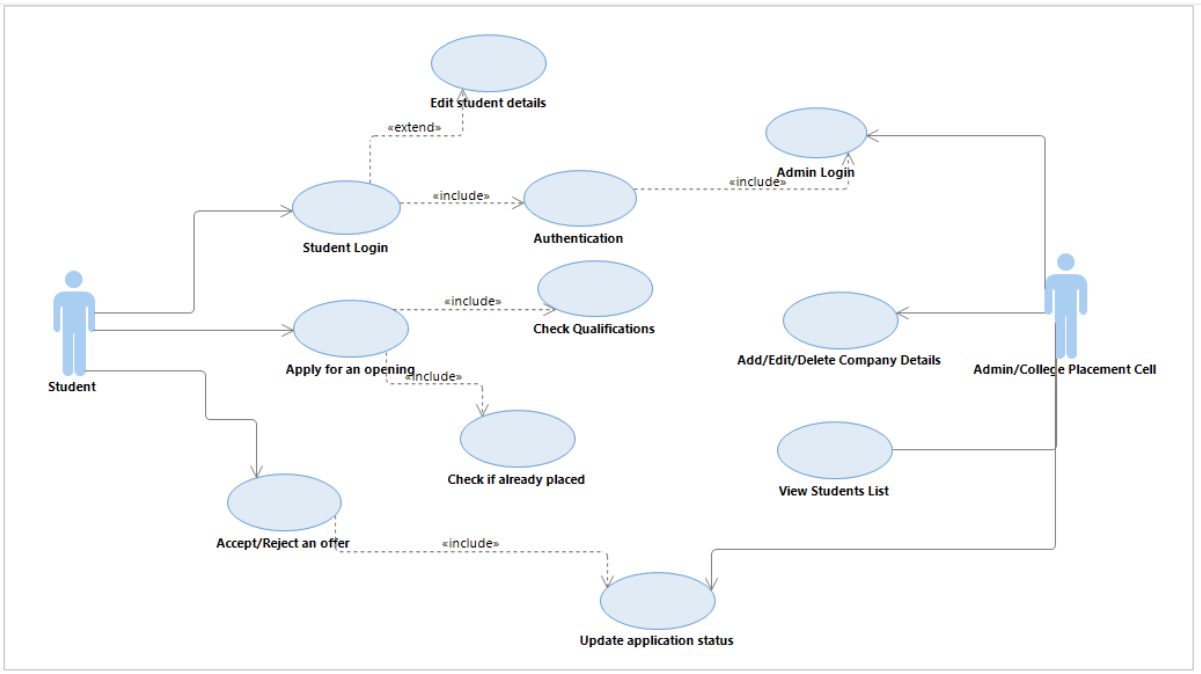

Figure 1: Use Case Diagram

Figure 1 shows the Activity diagram for the placement portal. Students can apply for a job/internship opening and can accept/reject an offer offered to the student after logging in to the software. Authentication and qualification checks are necessary criteria before a student can apply on the portal. ‘Edit student details’ stands as an optional activity, so it is extended with student entity.

Admin can add/edit/delete job/internship details after logging in to the system. They can view student lists of applicants, who have applied to a particular job/internship posting and can update students' application status on successful selection by the company interviews.

2.2.2 Use Case Template

Use Case Title	Placement Portal
Abbreviated Title	CGP
Actors	User, Admin
Description	Login is used to give access to the website to the user and the admin. Once authorized, the admin can post, update, or delete job/internship descriptions. Users can see the dashboard and their application status or can apply to the jobs/internships if they fulfill the necessary criteria required for the job/internship.
Pre-Conditions	Users must have their login credentials before logging in to the system.
Task Sequence	1. Users and Admin enter their user-id. 2. Users and Admin enter passwords. 3. Entered user-id and password are authenticated by the system. 4. The system shows an error message if authentication fails. 5. Users can check job/internship postings and can apply to them if they fulfill the necessary criteria set by the company. 6. Admin can post, update and delete the job openings, and can update the application status.
Post Condition	The system directs to the corresponding dashboard for the users, where they can see their application status. Admin and user can finally log out from the web application.
Author	Team Juan

2.2 Activity Diagram and Swimlane Diagrams

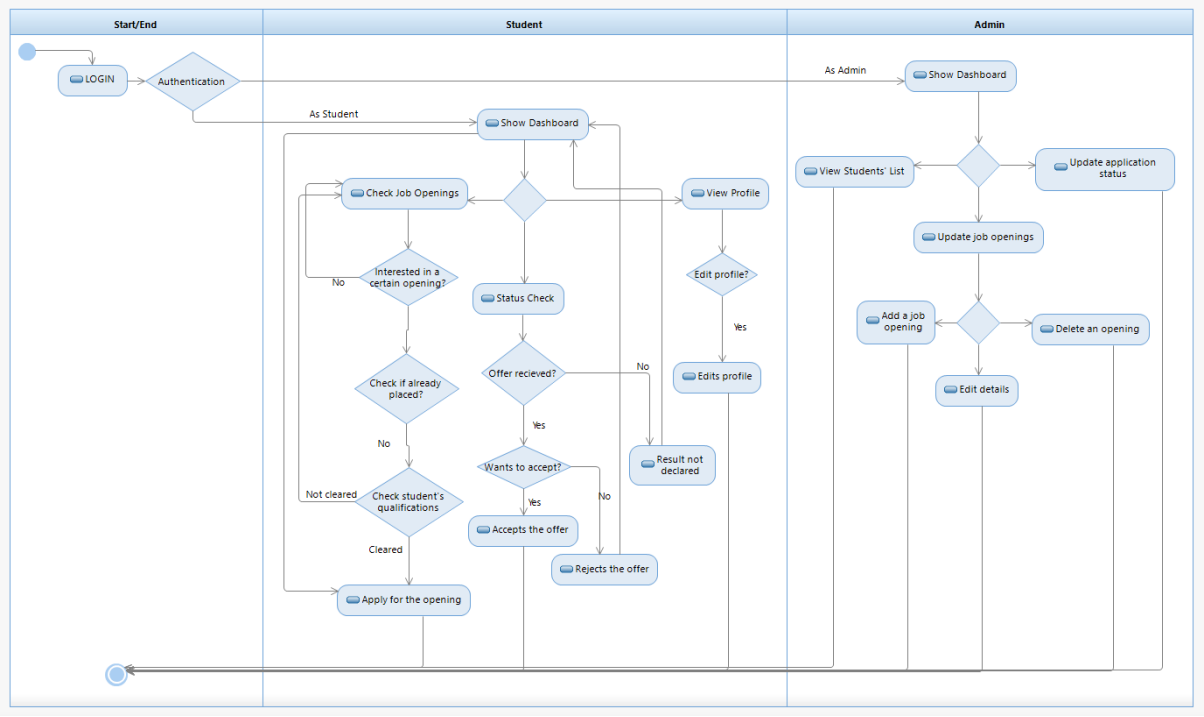

Figure 2: Swimlane Diagram

Figure 2 shows the Activity diagram for our placement portal. Activity diagram shows the flow of activities that goes while accessing and using the software. After authentication, the first thing that comes our way is the dashboard, which has different possible operations for both Admin and student. After successful operations by the student and the admin, they are either redirected to the dashboard where they can check the status or are allowed to logout.

2.3 Data Flow Diagrams

2.3.1 DFD Level 0

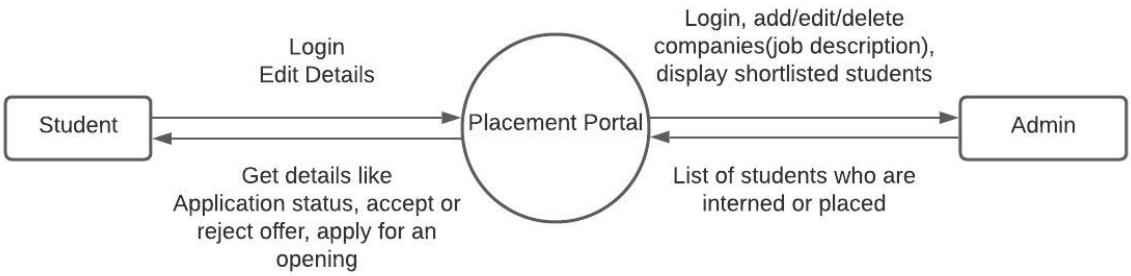

Figure 3.1: DFD Level 0

Figure 3.1 shows the Data Flow Diagram of the placement portal. This is the 0 level Data Flow Diagram which gives an overview about the information provided by the external entities, i.e., students and the admin to the portal, and what information they receive back.

2.3.2 DFD Level 1

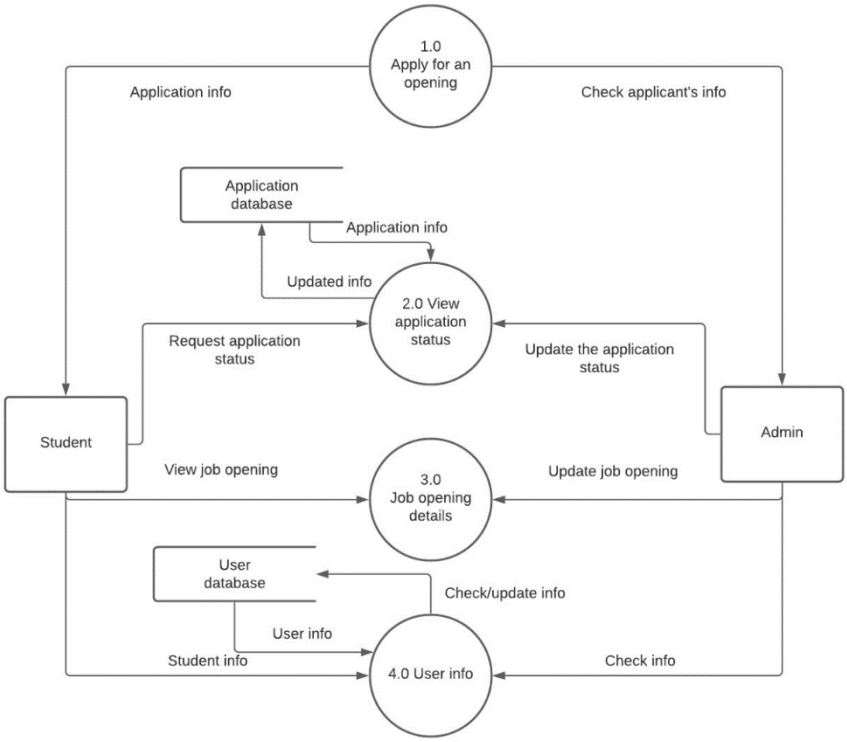

Figure 3.2: DFD Level 1

Figure 3.2 is the level 1 Data flow diagram. It basically represents the flow of data in the portal. It explains all the processes happening in the portal to that together form the complete system. We can think of a level 1 DFD as an "exploded view" of the context diagram. This figure also shows the databases where the data is stored.

2.3.3 DFD Level 2

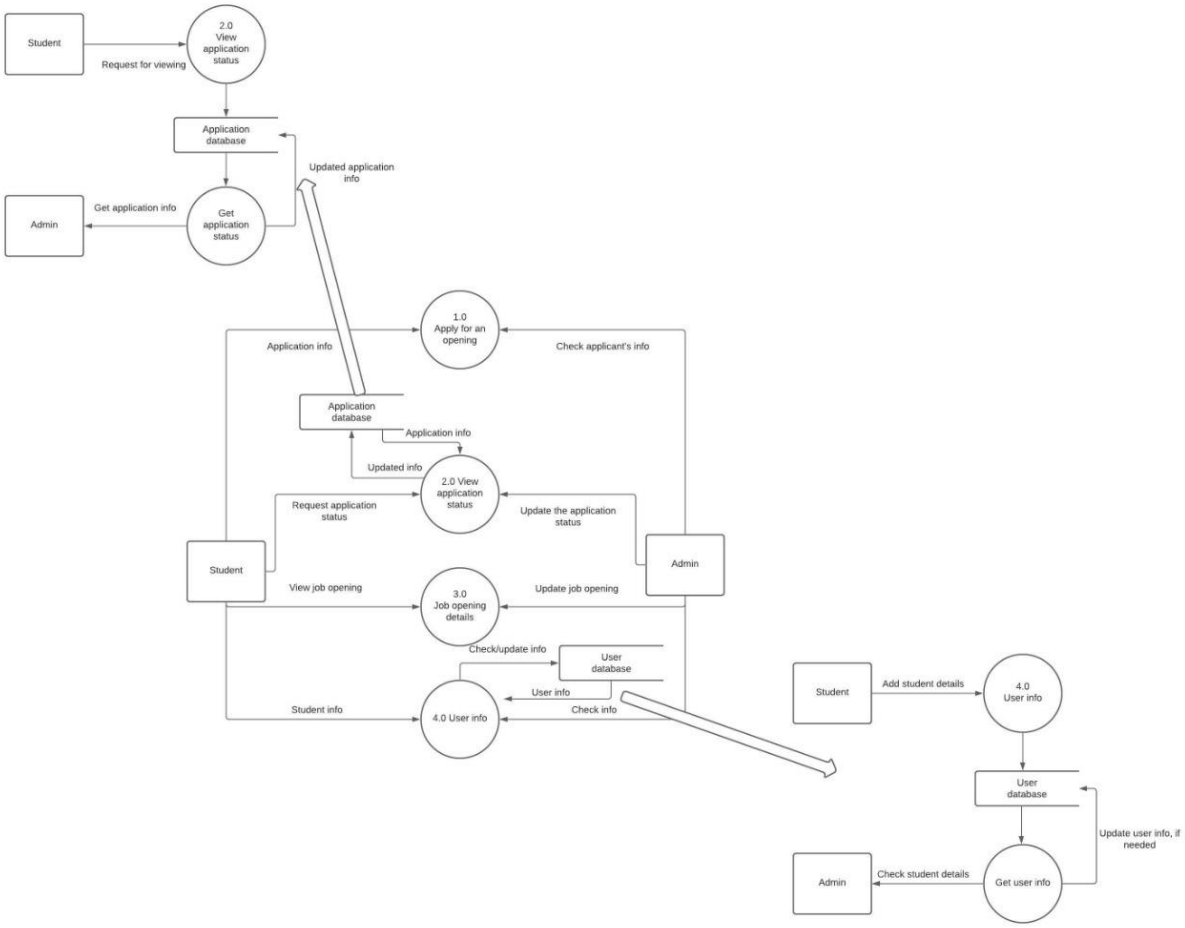

Figure 3.3: DFD Level 2

2.4 Software Requirement Specification in IEEE Format

Software Requirements Specification

for

CGP

Version 1.0 approved

Prepared by

Meenal Mehrotra

Maulika Luthra

Rashita

Ankur Singh

Thapar Institute of Engineering and Technology

26th August 2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
First Edit	14/10/21		
Final Edit	7/12/21		

1. Introduction

1.1 Purpose:

This Software Requirements Specification provides a description of all the functions and constraints of the Placement management System, developed for our colleges' placement cell. The project focuses on designing a Placement Management System for graduating students to a pool company from various domains.

1.2 Document Conventions:

Following typing, Conventions are followed.

- · 1.5 line spacing should be used for typing the general text.
- · The general text shall be justified and typed in the Font style Time New Roman and Font size 12.
- Heading shall be typed in the Font style 'Times New Roman' and Font size 21 and bold. Subheadings shall be typed in the Font style 'Times New Roman' and Font size 12 and bold.
- · Fancy fonts should be avoided for text writing.

1.3 Intended Audience

This project is mainly created for pre final and final year students of any college. It will help the students to smoothen out their whole placement process. It is useful for the placement cell, the admin, as well to reduce their manual work.

1.4 Scope and Overview

The Placement Management System is for the students which maintains the database for the students where all the students' records are entered including their academic details and their personal details. This software is intended to build an efficient system for the record of their placement statistics and deliver the best services and placement opportunities to their students. It will also manage the data of the Company, eligibility criteria and the facilities or the package it provides etc which will be handled by the admin. The System would provide the

facility of viewing both the personal and academic information of the student; it would also search for eligible students . It will search for eligible students based on the eligibility criteria.

2. Overall Description

2.1 Product Perspective

This project is aimed at developing a web application for the Placement Department of tour. The training and placement officers have to manage the students' profiles and the documents of students for their training and placement manually. Also, Placement Coordinators have to collect the information of various companies who want to recruit students and notify students from time to time about the placements.

Placement Coordinators also have to arrange profiles of students according to various streams and notify them according to company requirements. If any modifications or updates are required in the profile of the students or the Company, it has to be searched and done manually. Hence the Placement Management System would maintain a huge database for the complete details of the students as well as the Companies in the Placement and Internship process which would help to save time and effort.

2.2 Product Features

- The Placement Management System is to be developed as an attempt to take a record of Companies and students by restricting a large database that would be used for each.
- The System would provide the facility of viewing both the personal and academic information of the students and also the company.
- The System would also be able to search for eligible students and company with respect to their specifications and requirements.

2.3 Operating Environment

The project would be developed using Python framework, and django . Various system configurations are as follows-

- Operating system: windows 10
- Database: sqlite3
- Scripting Language: Django, javascript,HTML,CSS
- Web Browser:Any

2.4 User Documentation

The software product will contain a user manual that would be written to help people understand the working methodology and usage of the developed prototype system. It would be written from user point of view and thus for non-technical individuals and the level of content or terminology would differ considerably from, for example, a Developer's Manual, which is more detailed and complex. The user manual would follow common user documentation styles capturing purpose and scope of the product along with key system features and operations; step-by- step instructions for using the system including conventions, messaging structures, quick references, tips for errors and malfunctions; pointers to reference documents; and glossary of terms.

3. Interface Requirements

3.1 User Interfaces

The user interface will be html pages on a web browser through which all user queries will be processed and required information will be displayed. The website would contain following forms-

- Login Page
- New user Registration Form
- A page displaying job postings
- Company Profile
- Student Profile
- Employer Profile

3.2 Hardware Interfaces

The software heavily depends on the database retrieval system and web server. Thus a high speed processor and a RAID hard disk is recommended to avoid loss of data and faster processing of queries. Network connection is required in order to access the web portal.

3.3 Software Interfaces

The project has been set up using LAMP (Linux, Apache, MySQL and PHP). The product will host a local Apache web server where the user interface will be displayed via the web browser. The scripting language PHP will define image placement, size and overall set-up. PHP will also be used to create background colors, border colors and text display. JavaScript will be used to control client-side interactions. The MySQL database will store the images to be displayed at the output. All images are in jpeg format.

3.4 Communications Interfaces

The requirements associated with any communications functions required by this product, including email, web browser, network server communications protocols, electronic forms, and so on. Communication standards that will be used, such as FTP or HTTP. Communication security or encryption issues will be handled by using javascripts.

4. Non Functional Requirements

4.1 Performance Requirements

1. Display information about Job Openings– Display information should be prominent and fast. If a student wants to see available job openings the system should respond quickly. Also there is a requirement for sorting data according to the preferences specified by the user.

2. Usage of the result in the list - The results displayed in the list view should have various options available for students, for example- to apply for the job opening, to email related company profiles etc.

3. Response Time - The queries should be processed fast enough as slow responsiveness of the website makes users look for other available alternatives. The response time of a query response is tested by performing various queries at the time of development.

4. Tracking the student's status – The students who have not been yet placed would be given preferences and frequent emails about the available job posting.

5. Tracking User Id and/or Password - The user may lose credentials for login, so recovery option should be provided.

4.2 Security Requirements

There would be a session maintained and the user would be logged out automatically after a certain period of inactivity. The site will use a self-signed certificate for hosting on https i.e. secured port. For transferring documents TLS would be used and files will be transferred over secure FTP.

4.3 Software Quality Attributes

- **Availability-**
  The system should be available at all times, meaning the user can access all the time, only restricted by the down time of the server on which the system runs. In case of a of a hardware failure or database corruption, a replacement page will be shown. Also in case of a hardware failure or database corruption, backups of the database should be retrieved from the server and saved by the administrator. Then the service will be restarted. It means 24 X 7 availability.
- **Reliability-**
  The system provides storage of all databases on redundant computers with automatic switchover. The reliability of the overall program depends on the reliability of the separate components. The main pillar of reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes. Thus the overall stability of the system depends on the stability of container and its underlying operating system.
- **Maintainability-**
  A commercial database is used for maintaining the database and the application server takes care of the site. In case of a failure, a re-initialization of the program will be done. Also the software design is being done with modularity in mind so that maintainability can be done efficiently.
- **Portability-**
  The application is Java and php scripting language based. So the end-user part is fully portable and any system using windows and linux should be able to use the features of

the system, including any hardware platform that is available or will be available in the future.

An end-user may use this system on any OS, either Windows or Linux. The system shall run on PC, Laptops and PDA etc.

3. Design Phase

3.1 Class Diagram

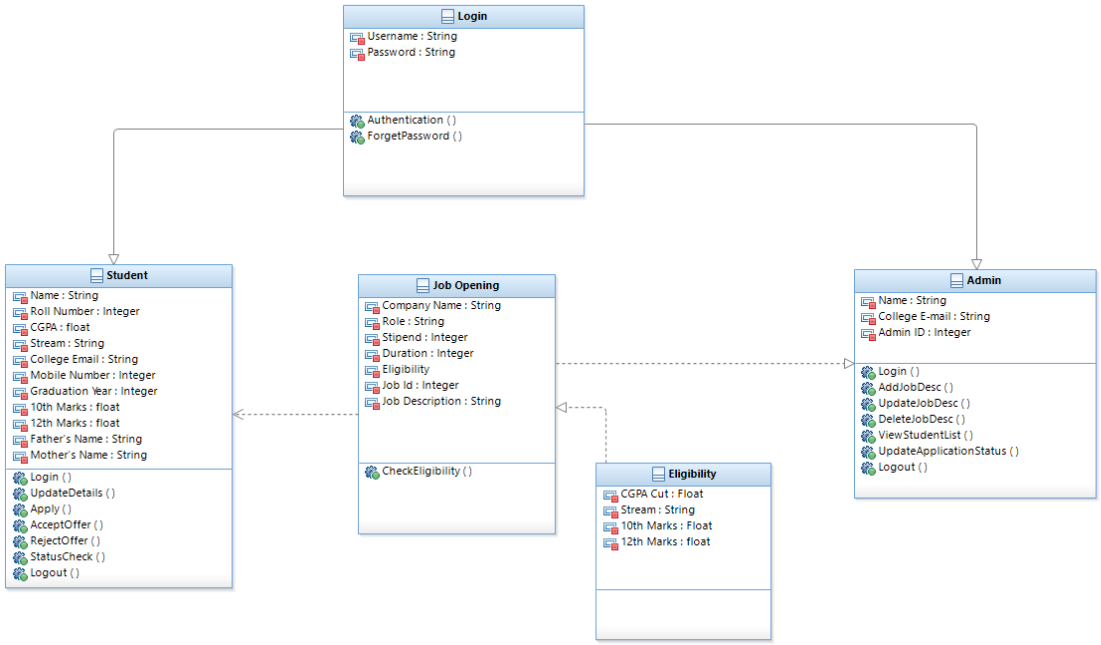

Figure 4: Class Diagram

Figure 4 shows the class diagram for our placement portal. Class diagram is the static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects. Class diagrams for login, student, admin, job opening and eligibility are made, and different operations corresponding to them are displayed along with the attributes these classes possess.

3.2 Sequence Diagram

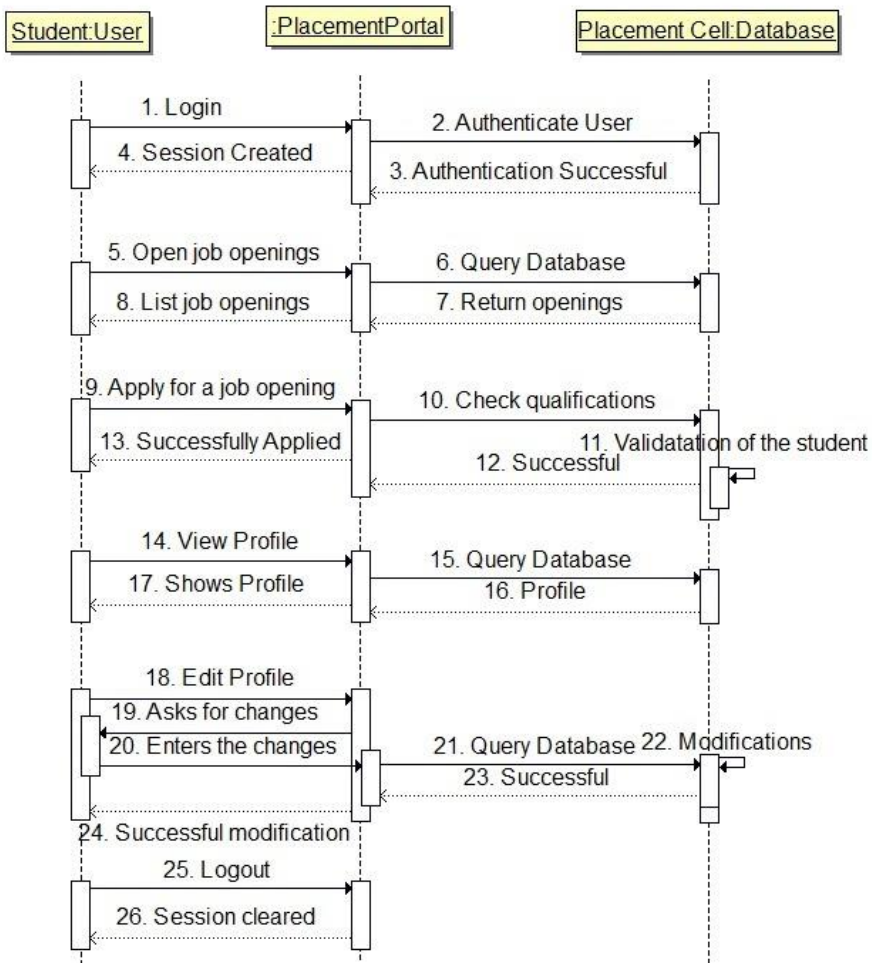

Figure 5.1: Sequence Diagram for User

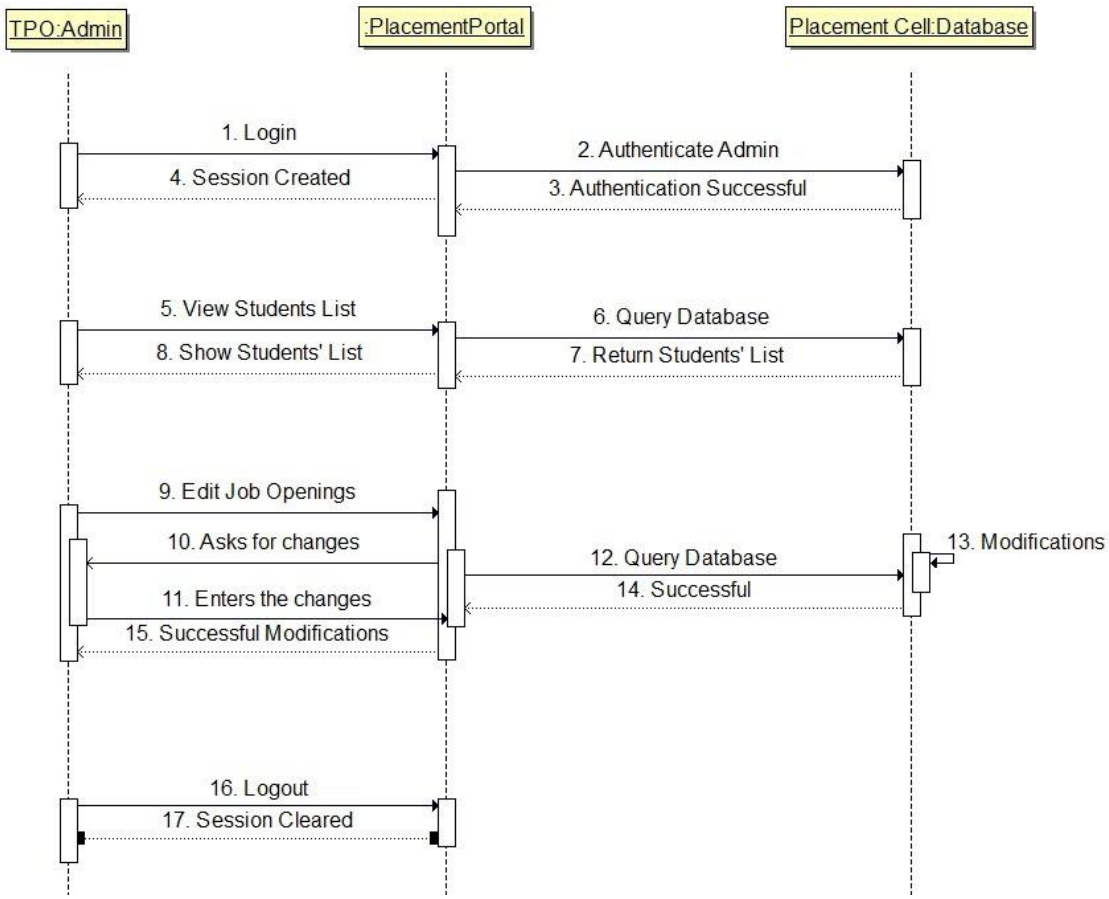

Figure 5.2: Sequence Diagram for Admin

Figure 5.1 and 5.2 shows the sequence diagram for the placement portal. Sequence diagram shows the interactions between the objects in a timely sequence manner. In both cases, the first operation is login. When a user/admin logs in on the portal, a request goes to the server for authentication, which is approved by verifying the credentials in the database. The same needs to be done by the admin to access the portal.

The message flow between the various objects is shown in a timely sequence using this diagram.

3.3 Collaboration Diagram

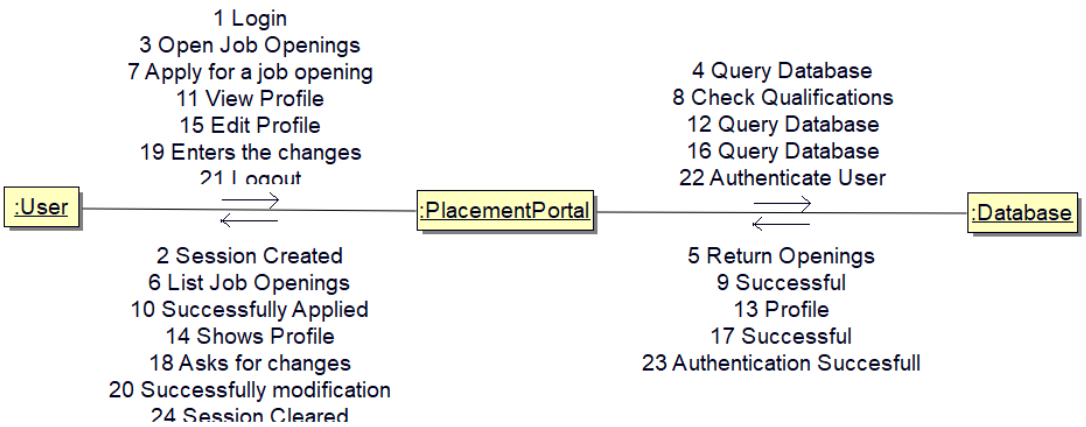

Figure 6.1: Collaboration diagram for User

Figure 6.1 shows the collaboration diagram for our placement portal. A collaboration diagram is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML).

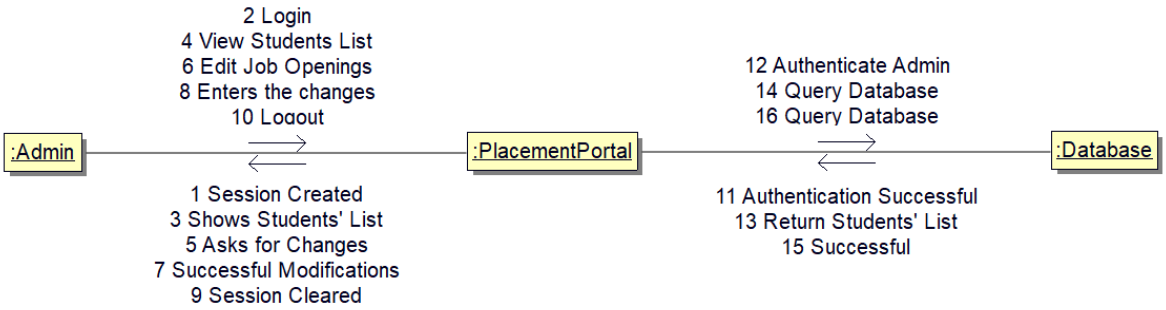

Figure 6.2: Collaboration diagram for Admin

3.4 Database Design: ER Diagram

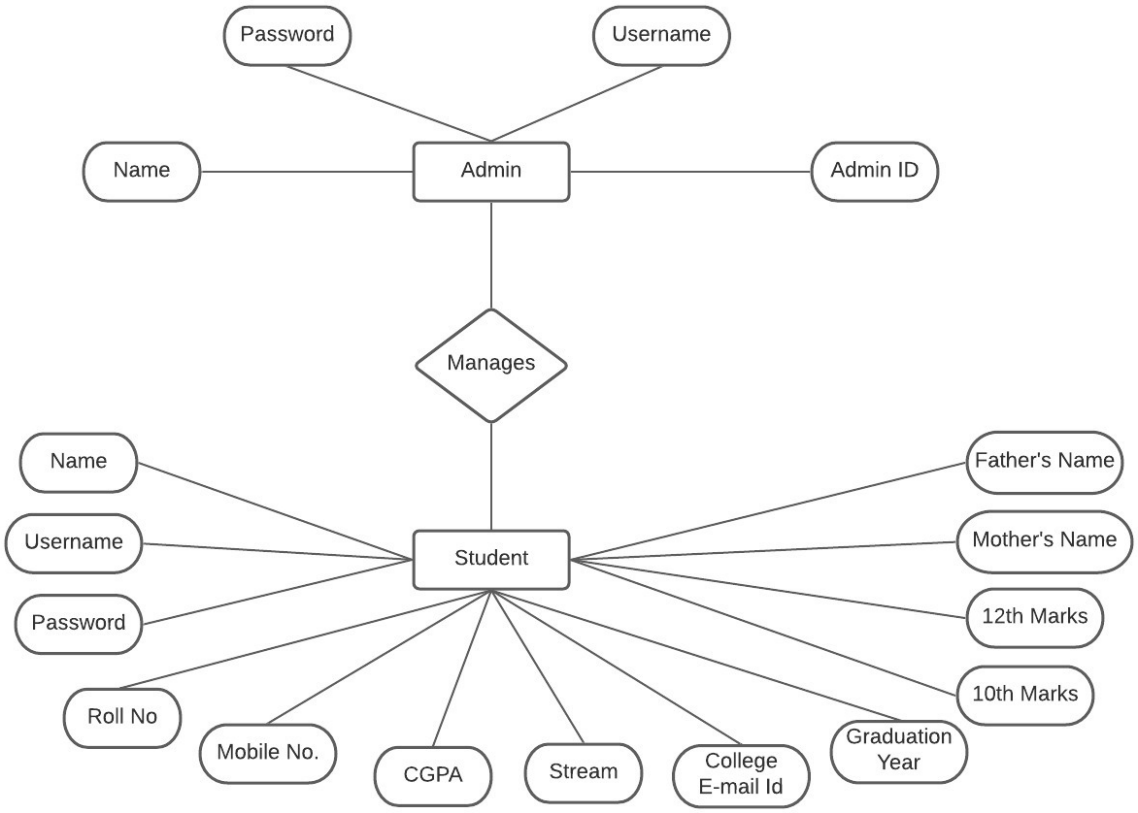

Figure 7: ER Diagram

Figure 7 shows the ER Diagram for our placement portal. We have two actors - Student and Admin. Admin manages the student. Various attributes corresponding to the student and Admin are shown in the figure above.

3.5 State Chart Diagram

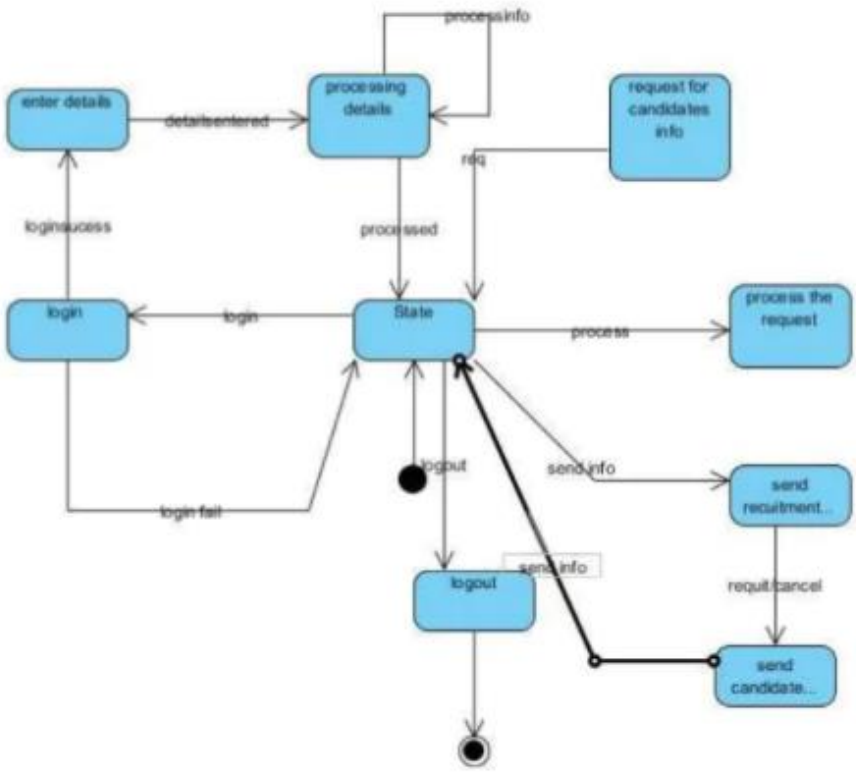

Figure 8: State Chart Diagram

Figure 8 models the dynamic behaviour of individual classes or any other kind of objects. They show the sequence of states that an object goes through, the events that causes a transition from one state to another and the actions that result from a state change. It is typically used to model the discrete stages of an object's lifetime. It typically contains one state chart and multiple end states and transitions connected to various states on the diagram

4. Implementation

4.1 Component Diagrams

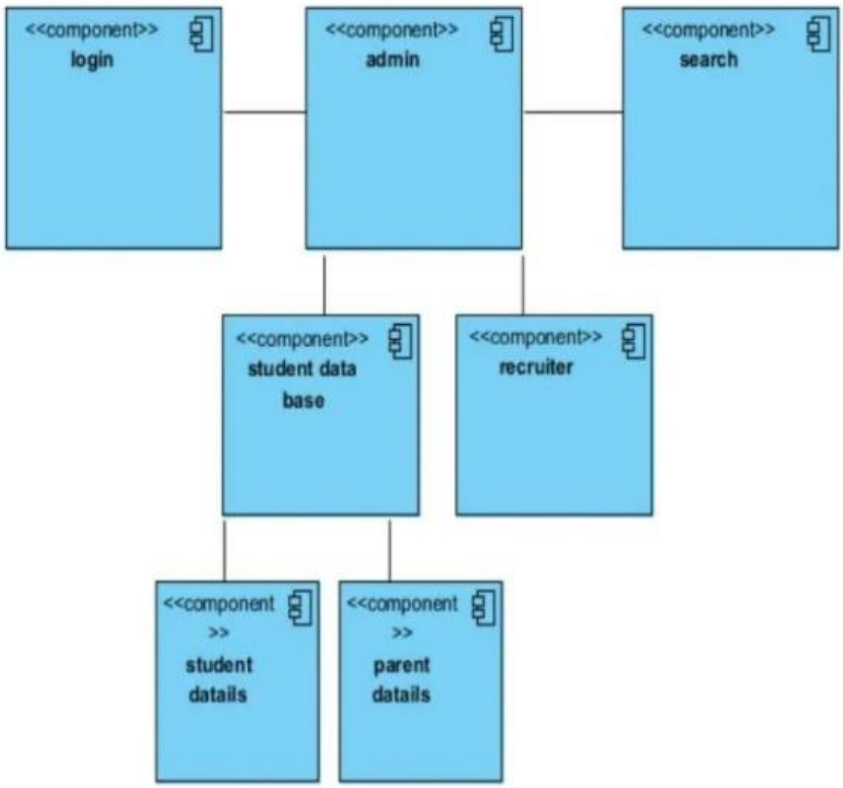

Figure 9: Component Diagram

Figure 9 breaks down the complete object-oriented system into components and represents the components used to make functionalities. It is a special kind of UML diagram with the main purpose of envisioning all the components used in the system and depicting the relationship and organization among all the components of the system.

4.2 Deployment Diagrams

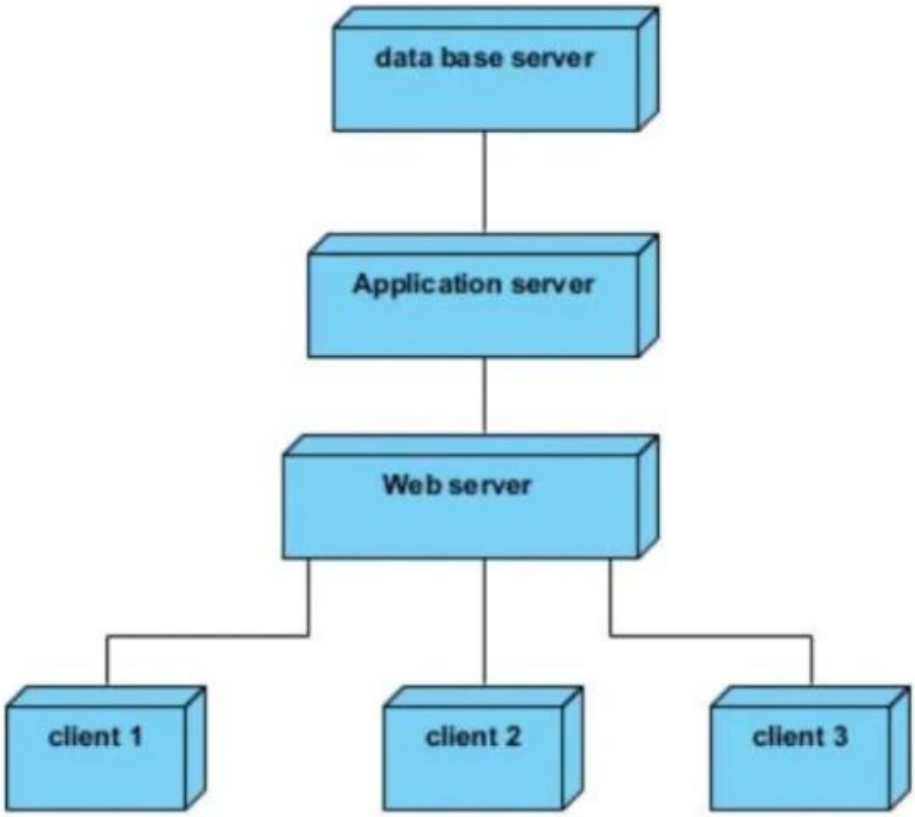

Figure 10: Deployment Diagram

Figure 10 is a deployment diagram which is a special kind of UML model, which shows the configuration of run-time processing nodes and the components used in the system. The main purpose of the deployment diagram is to understand the run-time configuration and see how the hardware used is linked to each other and the communication paths between them.

4.3 Screenshots

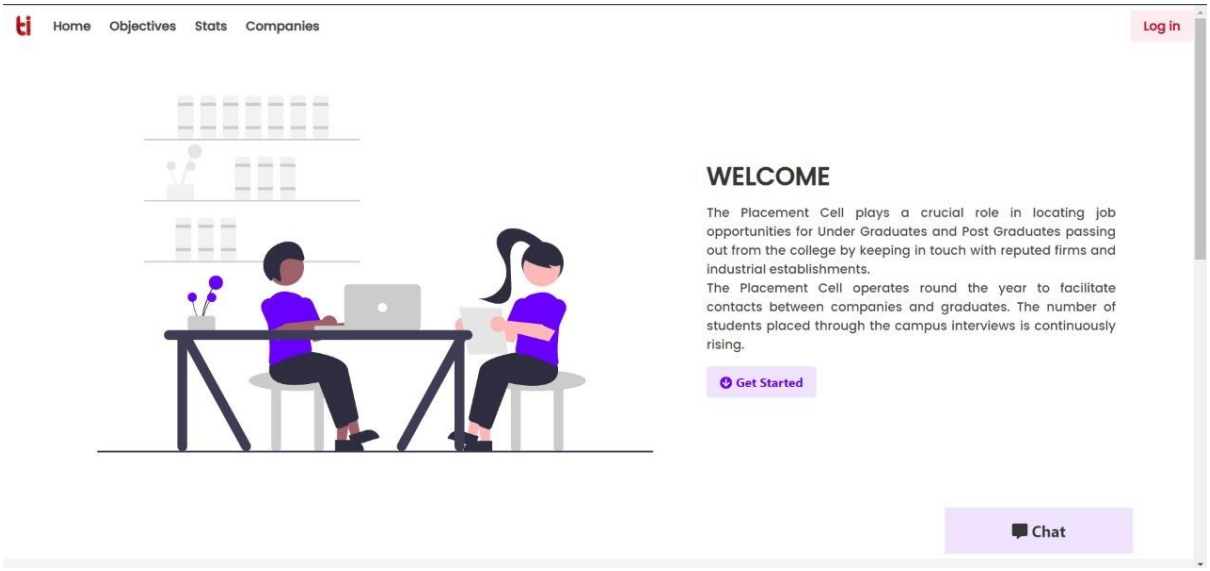

Figure 11.1: Welcome Page

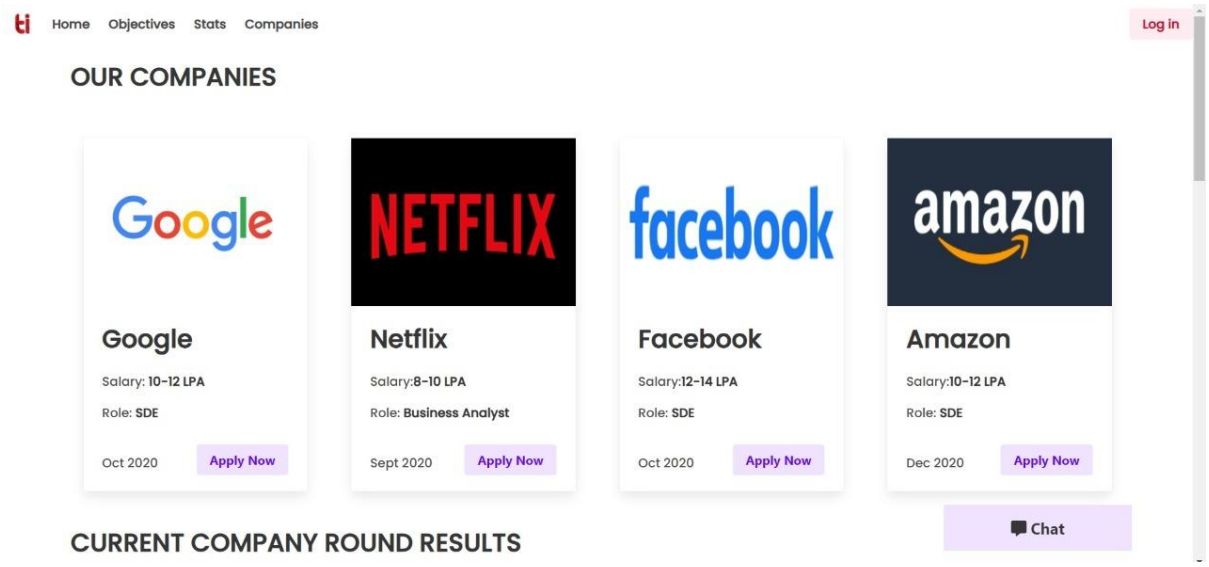

Figure 11.2: Display of all Recruiters

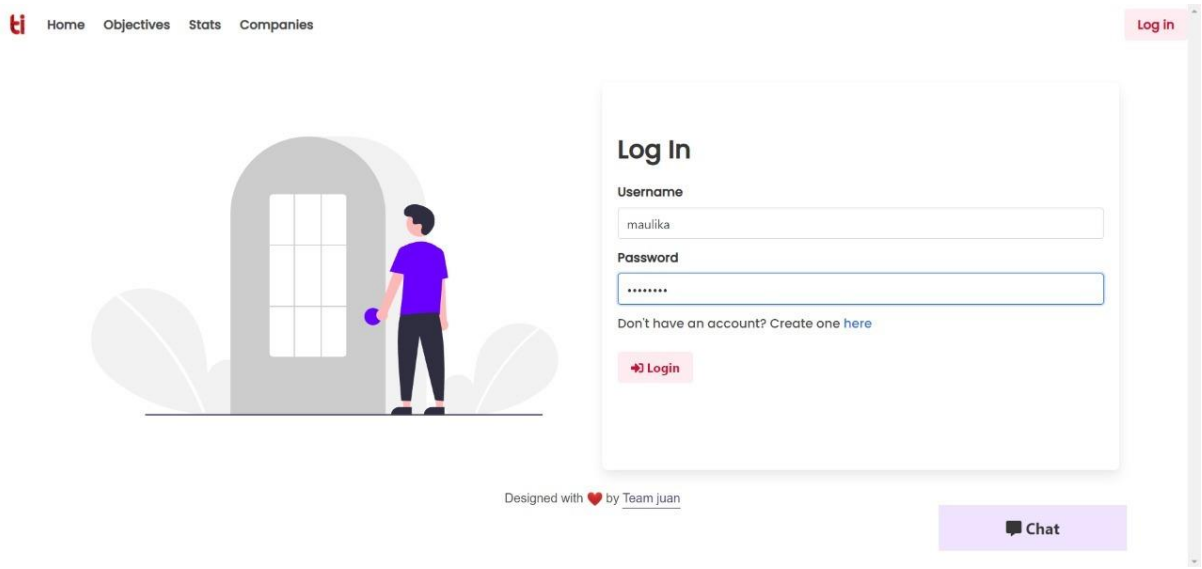

Figure 11.3: Login Page

5. Testing

5.1 Test Plan

Test environment:

OS: Linux, Windows, Android, iOS

Browsers: Mozilla Firefox 75+, Google Chrome 80+, Microsoft Edge, Apple Safari

Hardware: Accelerator: GPU

Test Resources:

Continuous Integration: We plan to use GitHub Actions to check the quality of the code. It can run different types of tests and if all the tests succeed then only the code gets merged in the codebase. With this approach we can maintain code quality. Common tests that run on GitHub Actions are Linting checks where it makes sure the code written is according to the standardized style guide. If linting is not up to mark, then it automatically formats the documents. We also run unit tests, integration tests and api tests.

Type of tests:

- **Unit Testing:** Unit testing technique using which individual modules are tested to

determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules. Python has a unit test library which will be used to test Django functions.

- Api testing: We plan to perform using python request library, so that we can get http code of api request and compare with all valid codes.
- Quality Assurance: We plan to perform tests according to test cases given below. These tests will be performed by developer and alpha, beta users and domain experts to ensure the software is practical in real world scenarios.

5.2 Test Cases

Test Case - 1

Test Case	1
Test Case Name	Student Login
System	User
Subsystem	Student Authentication
Designed by	Ankur Singh
Design Date	01-10-2021
Executed By	Rashita
Executed Date	08-10-2021
Short Description	Test User Login

Pre-conditions:

1. The User can access a website by typing the url in the browser that has been updated recently.
2. The user should already exist in the database.
3. The Username is ‘johndoe’ and the password is ‘ItisaStr0ngPas33w0rDTrustMe’.
4. The system authenticates the user and redirects it to the dashboard.

Step	Action	Expected System Response	Pass/Fail
1.	Click the ‘Login Button’ on navbar	Login page should open	Pass
2.	Enter ‘johndoe’ in username textfield	The text should be displayed on the textfield.	Pass
3.	Enter ‘ItisaStr0ngPas33w0rDTrustMe’ in password textfield	The text should be displayed on the textfield and should be hidden.	Pass
4.	Click on Login Button	The system should send data to the backend to authenticate and the user should be redirected to the dashboard.	Pass

Post-conditions:

1. The username ‘johndoe’ should appear on the dashboard.

Test Case - 2

Test Case	2
Test Case Name	Apply for test
System	User
Subsystem	Student
Designed by	Ankur Singh
Design Date	01-10-2021
Executed By	Rashita
Executed Date	08-10-2021
Short Description	Test to apply in various companies

Pre-conditions:

1. **User must log in before applying for any job.**

Step	**Action**	**Expected System Response**	**Pass/Fail**
1.	Click on companies on navigation bar	Page with different available to apply will open	Pass
2.	Choose any company and click on apply	Form will open to be filled with details	Pass
3.	Fill the form and Click on submit	Applied for job	Pass

Post-conditions:

1. A new instance of application will start appearing in database.

5.3 Test Reports

Executed	Passed	5
	Failed	0
	Total Test Executed	5
Pending		0
In Progress		0
Blocked		0

Test Planned		0
Total		5

Functions	Description	% TCs Executed	% TCc Passed	% TCs pending	Priority
Student Login	Test User Login	100 %	100 %	0 %	High
Apply for job	Test the functionality of the form	100 %	100 %	0 %	High