Campus Connect : Bridging Opportunities

A

Project Report

Submitted in partial fulfillment of the requirement for the award of degree of

Bachelor of Technology

In

Information Technology

Submitted to

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA,
BHOPAL (M.P.)

Guided By
Prof. Prashant Lakkadwala

Submitted By
Jahnavi Mandloi(0827IT211048)
Jayshree Dave(0827IT211050)
Mahee Dubey(0827IT211065)
Minakshi Soni(0827IT211069)

DEPARTMENT OF INFORMATION TECHNOLOGY
ACROPOLIS INSTITUTE OF TECHNOLOGY & RESEARCH,
INDORE (M.P.) 452020
2024-2025

Declaration

We hereby declared that the work, which is being presented in the project entitled **Campus Connect: Bridging Opportunities** partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology**, submitted in the department of Information Technology at **Acropolis Institute of Technology and Research, Indore** is an authentic record of our own work carried under the supervision of **Prof. Prashant Lakkadwala**. We have not submitted the matter embodied in this report for the award of any other degree.

Jahnavi Mandloi(0827IT211048)
Jayshree Dave(0827IT211050)
Mahee Dubey(0827IT211065)
Minakshi Soni(0827IT211069)

Prof. Prashant Lakkadwala

Supervisor

Project Approval Form

We hereby recommend that the project **Campus Connect: Bridging Opportunities** prepared under our supervision by Jahnavi Mandloi(0827IT211048), Jayshree Dave(0827IT211050), Mahee Dubey(0827IT211065), Minakshi Soni(0827IT211069) be accepted in partial fulfillment of the requirement for the degree of Bachelor of Technology in Information Technology.

Prof. Prashant Lakkadwala

Supervisor

Recommendation concurred in 2024-2025

Prof. Varsha Choudhary

Project Incharge

Prof. Deepak Singh Chouhan

Project Coordinator

Acropolis Institute of Technology & Research

Department of Information Technology

Certificate

The project work entitled **Campus Connect:Bridging Opportunities** submitted by Jahnavi Mandloi(0827IT211048), Jayshree Dave(0827IT211050), Mahee Dubey(0827IT211065), Minakshi Soni(0827IT211069) is approved as partial fulfillment for the award of the degree of Bachelor of Technology in Information Technology by Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.).

Internal Examiner

Name:…….

Date: …./…/……..

External Examiner

Name: ……..

Date: …./…/……..

Acknowledgement

With boundless love and appreciation, we would like to extend our heartfelt gratitude and appreciation to the people who helped us to bring this work to reality. We would like to have some space of acknowledgement for them.

Foremost, our would like to express our sincere gratitude to our supervisor,

Prof. **Prashant Lakkadwala** whose expertise, consistent guidance, ample time spent and consistent advice that helped us to bring this study into success.

To the project in-charge **Prof. Varsha Choudhary** and project coordinator **Prof. Deepak Singh Chouhan** for their constructive comments, suggestions, and critiquing even in hardship.

To the honorable **Prof. Prashant Lakkadwala**, Head, Department of Information Technology for his favorable responses regarding the study and providing necessary facilities.

To the honorable **Dr. S.C. Sharma**, Director, AITR, Indore for his unending support, advice and effort to make it possible.

Finally,we would like to pay our thanks to faculty members and staff of the Department of Information Technology for their timely help and support.

We also like to pay thanks to our **parents** for their eternal love, support and prayers without them it is not possible.

Jahnavi Mandloi(0827IT211048)

Jayshree Dave(0827IT211050)

Mahee Dubey(0827IT211065)

Minakshi Soni(0827IT211069)

Abstract

The Campus Connect: Bridging Opportunities platform is an advanced recruitment management system designed to streamline campus placements by enhancing efficiency, transparency, and communication among students, college administrators, and recruiters. Traditional campus recruitment methods are often slow and disorganized, relying on manual processes, extensive paperwork, and scattered communication, which leads to inefficiencies, delays, and missed opportunities. Campus Connect addresses these issues by providing an automated, centralized platform where companies can request recruitment drives, administrators can review and approve them, and students can apply for opportunities that match their qualifications.

The platform features real-time notifications, eligibility-based filtering, and seamless tracking of applications and recruitment statuses, ensuring a structured and smooth hiring process. Built using a robust technology stack, including Java, MySQL, HTML, CSS, and JavaScript, it ensures scalability, security, and ease of use. By integrating modern web technologies and structured workflows, Campus Connect bridges the gap between students and employers, making campus recruitment more efficient, data-driven, and transparent for all stakeholders.

Table of Content

List of Figures

List of Tables

INTRODUCTION

1.1 Rationale

The increasing complexity and inefficiency of the traditional campus recruitment process has created a demand for a more streamlined, digital solution. The existing process relies on manual procedures, emails, and physical interactions, leading to delays, confusion, and missed opportunities for students, colleges, and companies alike. To address these issues, Campus Connect: Bridging Opportunities is designed to provide an efficient, automated recruitment platform that simplifies communication between students, the college administration, and companies. By leveraging technology, the platform will create a more organized and transparent recruitment process, improving efficiency and reducing the time spent on administrative tasks.

1.2 Existing System

Traditional campus recruitment process are inefficient due to manual procedures, reliances on email, and physical interactions, making it difficult for students to track schedules, eligibility, and application statuses. Administrators face challenges coordinating activities and ensuring smooth communication, while companies struggle target the right candidates without access to comprehensive student data. This disorganized leads to delays, confusion, and missed opportunities for all parties. Here are the features and limitations of existing platforms:-

Linkedin

Features:Networking platform for professionals, different fields available, allows job postings.

Limitations:Designed for both freshers and experienced professionals, but we are focusing on freshers.

Naukri

Features:Job listings and applications.

Limitations:Limited access to student data, not campus-specific.

Remotive-

Features: Remote job listings, community support, and resources for remote work.
Limitations:Limited to remote jobs, fewer entry-level opportunities.

Indeed

Features:Extensive job listings, resume uploads, employer reviews. Limitations:High competition, generic job recommendations, limited campus-specific filtering.

1.3 Problem Formulation

The traditional campus recruitment process is plagued with various challenges:

- Inefficiency in Application Tracking: Students often struggle to track their application status across multiple companies, while recruiters have difficulty managing large volumes of applicants.
- Communication Gaps: There is often a lack of seamless communication between students, recruiters, and the college administration, leading to missed deadlines, confusion, and delays in the recruitment process.
- Limited Visibility for Recruiters: Companies do not have access to a comprehensive student database, making it difficult to filter out candidates based on specific qualifications.
- Manual Administrative Work: The administrative burden on college staff is significant, as they must manage recruitment schedules, student eligibility, and ensure that both students and companies are informed about the process.

1.4 Proposed System

- Centralized Platform: Create a single interface for students, administrators, and companies to interact and manage recruitment activities.
- Automated Eligibility and Notification System: Notify only eligible students about relevant job opportunities based on their profiles, ensuring that companies have access to the right candidates.
- Real-time Tracking: Allow students to track their application status, interview schedules, and selection results in real-time.
- Simplified Process for Companies: Enable companies to post recruitment drives and job openings, which are then reviewed and approved by the college admin before being made available to students. This ensures that the companies requirements are met and that only eligible students apply.

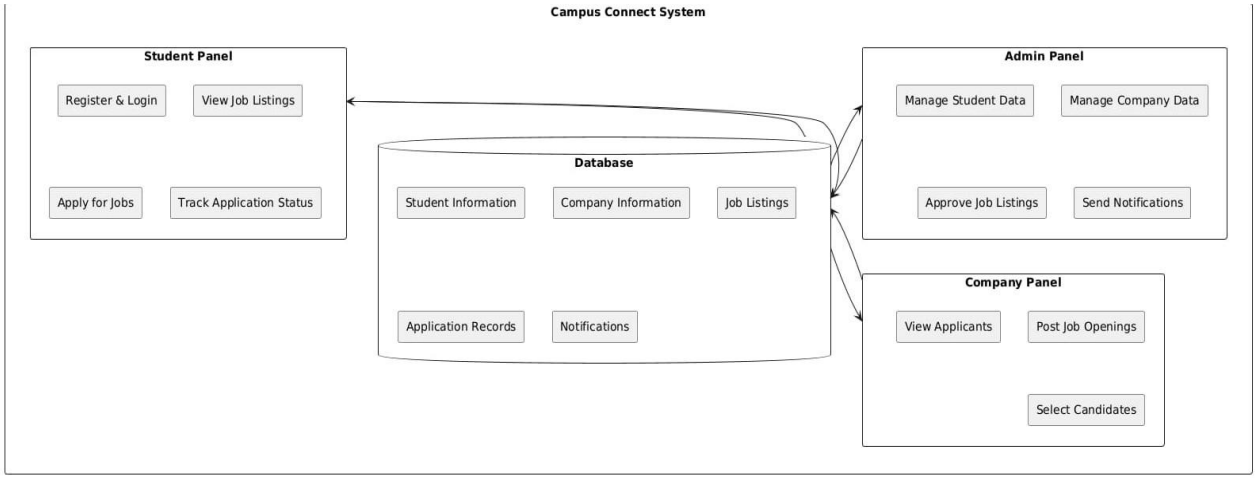

Fig.1 Block Diagram of Campus Connect:Bridging Opportunities

1.5 Objectives

- Streamlining the Recruitment Process: To create a digital platform that simplifies the recruitment process for colleges, students, and companies. This platform will automate and organize the procedures involved in campus recruitment, reducing manual intervention.
- Ensuring Efficient Communication: To establish a seamless communication system between students, college administration, and recruiters. The platform will facilitate timely notifications regarding job openings, deadlines, interviews, and other important updates.
- Student Eligibility Tracking: To implement an automatic eligibility tracking system that filters students based on their qualifications and ensures that only eligible candidates are notified about relevant job opportunities. This will save recruiters time by narrowing down the candidate pool.
- Centralized Management for Admin: To provide college administrators with a comprehensive dashboard to manage recruitment drives, track student progress, and oversee the entire recruitment process. This will enhance administrative control and coordination between all parties involved.
- User-Friendly Interface for Students: To create an easy-to-use interface for students, allowing them to browse job opportunities, track application statuses, and receive timely updates on their applications. This will help students stay informed and organized during the recruitment process.

- Company Engagement: To provide a dedicated platform for companies to post job opportunities, view student profiles, and connect directly with colleges to conduct structured recruitment drives. The system will help companies identify and recruit the right candidates quickly and efficiently.
- Transparency and Real-Time Updates: To ensure transparency in the recruitment process by providing real-time updates to all users (students, companies, and admins), keeping everyone informed about application statuses, interview results, and selection outcomes.

1.6 Contribution of the Project

1.6.1 Market Potential

With the increasing demand for efficient recruitment solutions in educational institutions and the rise of digital platforms, Campus Connect has significant market potential. It serves an untapped niche by focusing specifically on campus-based recruitment, which is a critical need for colleges, students, and companies. As the trend towards digitization in recruitment grows, this platform could attract widespread adoption across educational institutions and companies looking to streamline their hiring processes, creating immense growth opportunities in the market.

1.6.2 Innovativeness

The Campus Connect: Bridging Opportunities platform offers an innovative solution by combining automation with a campus-specific focus. Unlike general job platforms, it tailors the recruitment process to the unique needs of colleges, students, and companies, providing personalized notifications and eligibility-based filtering. Its integration of real-time tracking, transparent communication, and centralized management distinguishes it as a modern, forward-thinking platform for campus recruitment.

1.6.3 Usefulness

This platform is highly useful for all stakeholders involved in the campus recruitment process. Colleges can efficiently manage and track recruitment activities, while students gain easy access to relevant job opportunities and real-time updates. Companies benefit from streamlined recruitment and better access to qualified

candidates, saving time and effort. The system's focus on eligibility filtering ensures that only suitable candidates are notified, enhancing the quality of the recruitment process.

1.7 Report Organization

The report is structured into the following chapters:

1. Chapter 1: Introduction – Overview of the project, its objectives, and the rationale.

2. Chapter 2: Requirement Engineering – Discusses the feasibility, requirement gathering, and analysis.

3. Chapter 3: Analysis & Conceptual Design – Details the system architecture, design, and technical components.

4. Chapter 4: Implementation & Testing – Describes the development methodology, tools used, and testing strategies.

5. Chapter 5: Results & Discussion – Presents the results of the implementation and testing, followed by a discussion of findings.

6. Chapter 6: Conclusion & Future Scope – Concludes the report and outlines possible future developments.

REQUIREMENT ENGINEERING

2.1 Feasibility Study

Technical Feasibility: The Campus Connect platform is technically feasible, leveraging Java for the backend, HTML, CSS, and JavaScript for the frontend, and MySQL 8.0 for database management. Developed using Apache NetBeans on Windows, it follows a scalable three-tier architecture with encryption and HTTPS for security. It supports integration with external systems and ensures secure login and data protection. With version control, thorough testing, and potential cloud deployment, the platform is built for scalability, security, and ease of maintenance.

Economic Feasibility: The Campus Connect platform is economically feasible, with low development costs due to open-source technologies and free development tools. Operational expenses, such as cloud hosting and maintenance, scale with growth.

Operational Feasibility : The Campus Connect platform is operationally feasible due to its user-friendly design, automation of key processes, and scalability. It reduces manual workload for admins and users, ensuring smooth operation. The platform can easily scale with growing users and requires minimal maintenance and support. Overall, it is simple to manage and operate

2.2 Requirement Collection

2.2.1 Discussion

Campus Connect is an automated recruitment platform designed to enhance efficiency, transparency, and communication in the hiring process. By automating tasks, it saves time and streamlines operations, making recruitment faster and more organized. The platform ensures transparency by providing real-time updates to users, keeping all stakeholders informed. Its scalability allows it to adapt to growing demands, making it a reliable solution for expanding organizations. With a user-friendly interface and strong security measures, Campus Connect has significant potential. However, its success depends on widespread adoption and continuous improvements in security to ensure a seamless and secure recruitment experience.

2.2.2 Requirement Analysis

The requirement analysis for the Campus Connect: Bridging Opportunities platform focuses on creating a user-friendly, scalable, and secure system to streamline the campus recruitment process. It must enable seamless communication between students, admins, and companies, facilitate efficient job application management, and ensure eligibility-based notifications. The platform should be highly reliable, with real-time updates, easy navigation, and strong data security measures to protect sensitive user information. Additionally, it should be adaptable to growing user demands and comply with data privacy regulations.

2.3 Requirements

2.3.1 Functional Requirements

User Management: Registration and login for students, recruiters, and admin.

Job Posting: Companies can post job openings.

Eligibility Filtering: Automatically filters students based on qualifications.

Notifications: Alerts for job opportunities and application status.

Application Tracking: Students track their application progress.

Admin Dashboard: Admin can manage drives and student data.

Job Search: Students can search and filter job listings.

2.3.2 Non-Functional Requirements

Performance: Fast load times and high concurrency support.

Scalability: Supports increasing users without performance loss.

Security: Data encryption and secure logins.

Availability: 99.9% uptime.

Usability: Easy, intuitive interface.

Compatibility: Works across major browsers and mobile devices.

Maintainability: Easy updates and clear documentation.

2.4 Hardware & Software Requirements

2.4.1 Hardware Requirement (Developer & End User)

- Processor: Intel Core i5 or higher
- RAM: 8GB or more
- Storage: 100GB minimum
- Operating System: Windows/Linux

2.4.2 Software Requirement (Developer & End User)

- Programming Languages: Java, HTML, CSS, JavaScript
- Database: MySQL 8.0
- Operating System: Windows 10
- IDE: Apache NetBeans 23

2.5 Use-case Diagrams

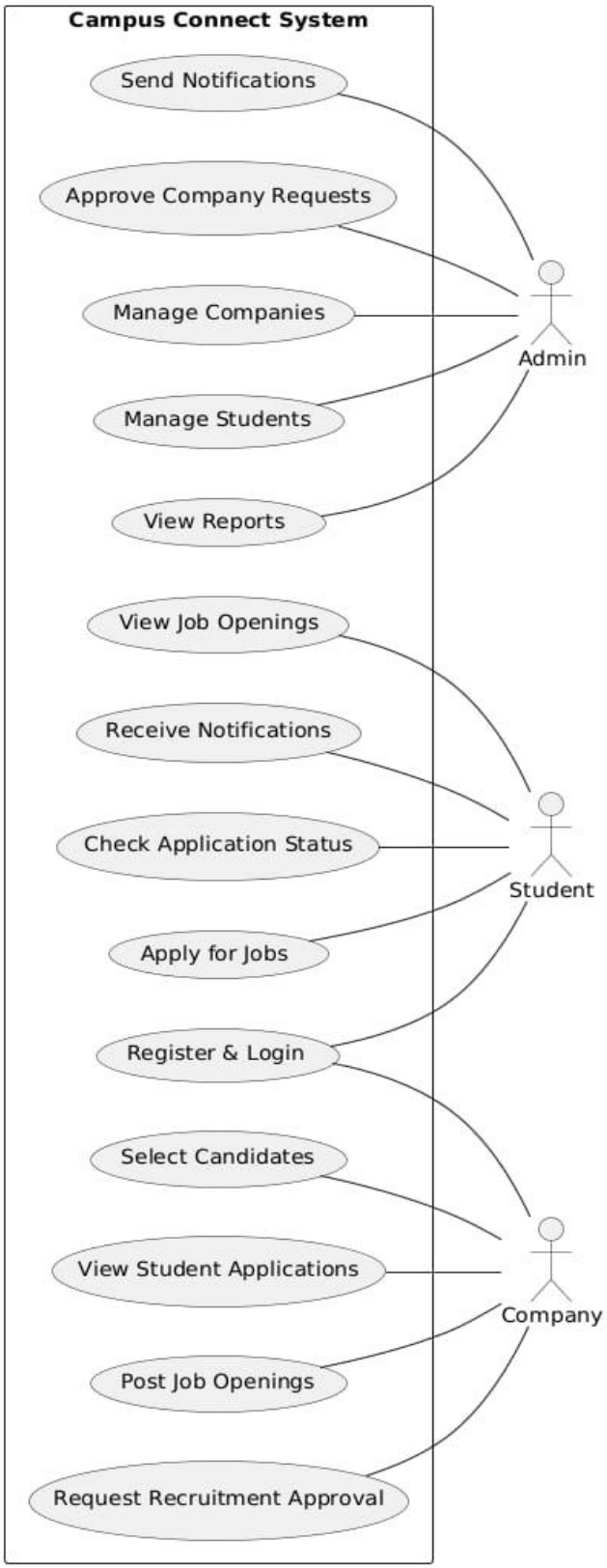

Fig 2.5 Use-Case Diagram

2.5.1 Use Case Description

The Campus Connect System is designed to facilitate job placements by connecting students, companies, and administrators. The system allows companies to post job openings, students to apply for jobs, and administrators to manage the overall recruitment process. Below are the detailed use case descriptions

1. Send Notifications

Actor:Admin
Description: The admin sends important notifications to students and companies regarding job openings, deadlines, and system updates.

2. Approve Company Requests

Actor:Admin
Description: The admin reviews requests from companies wanting to recruit students and either approves or rejects them.

3. Manage Companies

Actor:Admin
Description: The admin can add, update, or remove company profiles from the system. This ensures that only verified companies can participate in recruitment.

4. Manage Students

Actor:Admin
Description: The admin maintains student records, verifies their eligibility, and updates their status regarding placement activities.

5. View Reports

Actor:Admin
Description: The admin can access various reports related to student placements, job postings, and application statistics for analysis.

6. View Job Openings

Actor:Student
Description: Students can browse job opportunities posted by companies, filtering based on criteria like location, role, and salary.

7. Receive Notifications

Actor:Student
Description: Students receive notifications regarding new job openings, selection updates, and recruitment deadlines.

8. Check Application Status

Actor:Student
Description: Students can track the status of their job applications, whether pending, shortlisted, or rejected.

9. Apply for Jobs

Actor:Student
Description: Students can submit applications for job openings by uploading resumes and filling out necessary details.

10. Register & Login

Actors:Student,Company
Description: Students and companies must register and log in to access system functionalities.

11. Select Candidates

Actor:Company
Description: Companies can shortlist and select candidates based on their applications and qualifications.

12. View Student Applications

Actor:Company
Description: Companies can review submitted applications and filter candidates based on job requirements.

13. Post Job Openings

Actor:Company
Description: Companies can create and post job openings for students to apply.

14. Request Recruitment Approval

Actor:Company
Description: Companies must request admin approval before conducting recruitment drives on the platform.

ANALYSIS AND CONCEPTUAL DESIGN AND TECHNICAL ARCHITECTURE

3.1 Technical Architecture

The system follows a client-server architecture:
Client-side: The web interface allows users (students, companies, and admins) to interact with the system for registration, job applications, posting jobs, approvals, and feedback.
Server-side: The backend processes user inputs, manages business logic, and communicates with the database to send real-time updates.
Database: A MySQL database stores all user data, job postings, applications, approvals, and feedback.

3.2 Sequence Diagrams

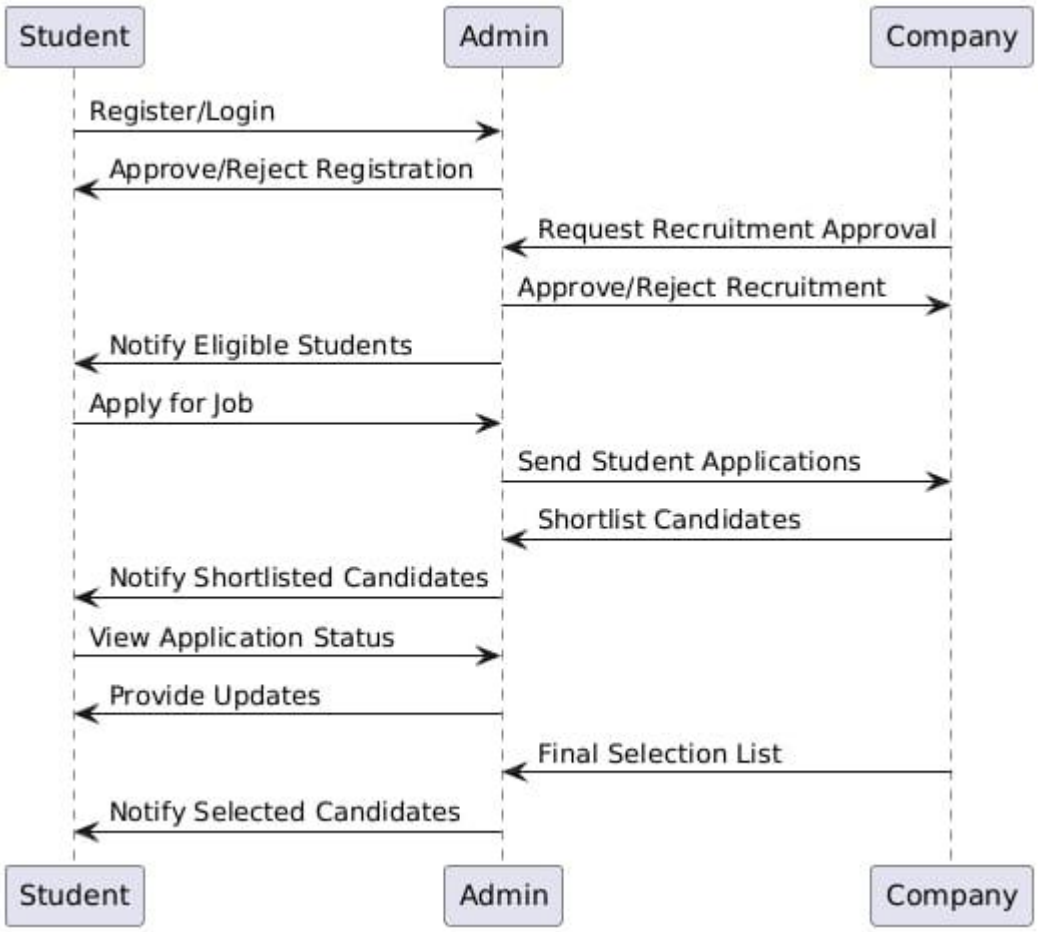

Fig. 3.2 Sequence diagram

3.3 Class Diagrams

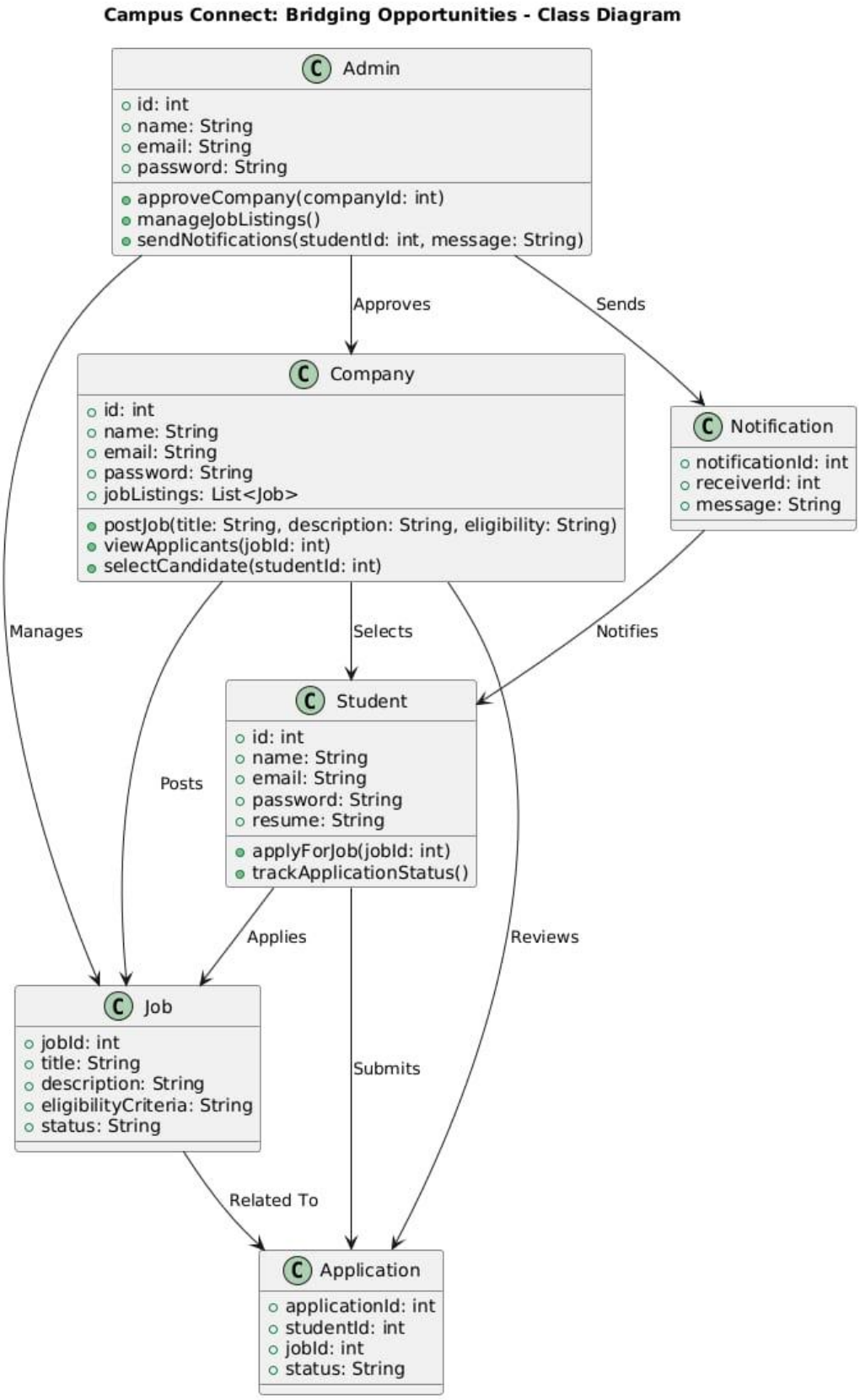

Fig. 3.3 Class diagram

3.4 Data Flow Diagrams (DFD)

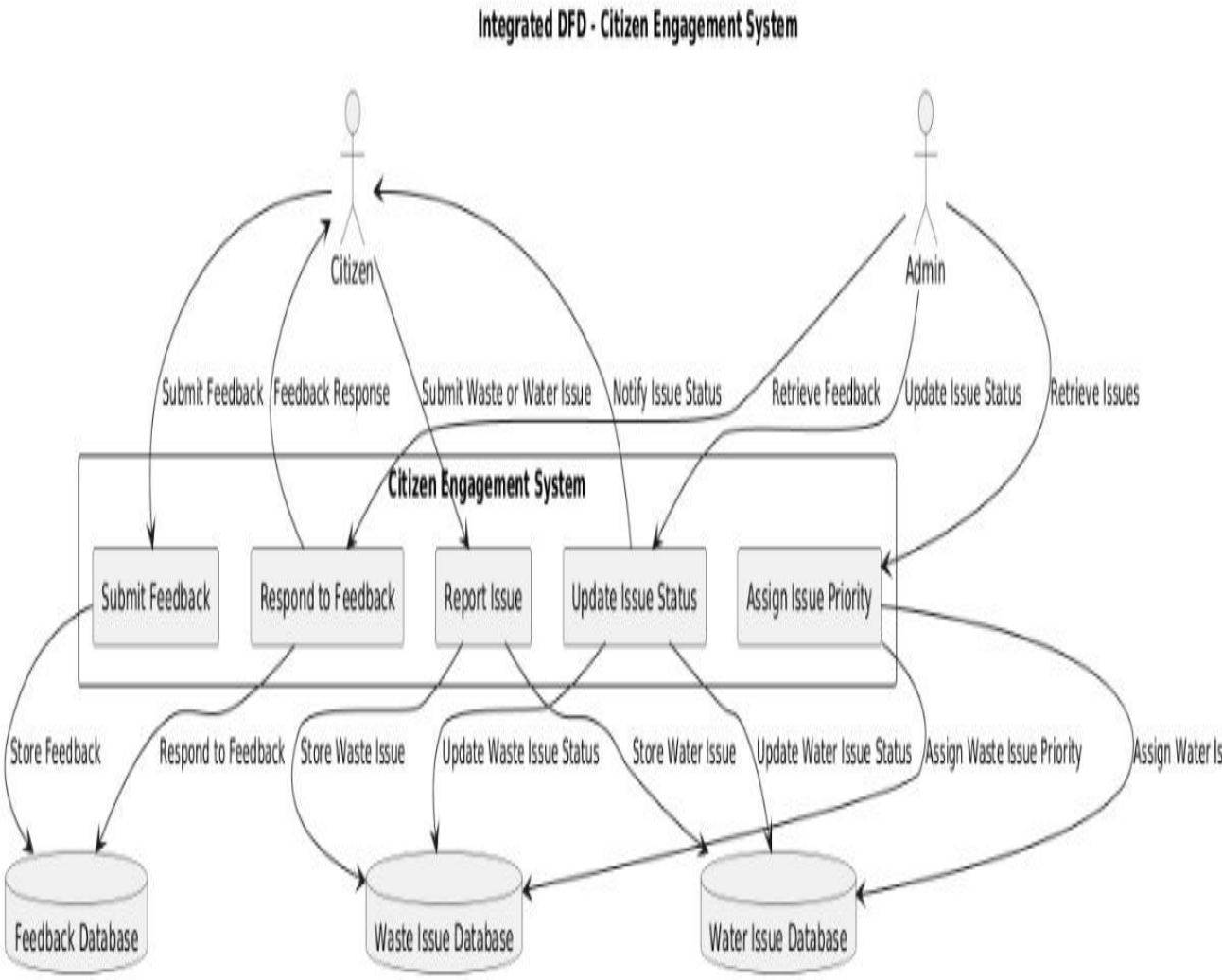

Fig.3.4 Data flow diagram

3.5 User Interface Design

The UI is designed with simplicity and accessibility in mind. It includes: Responsive Design: The interface adjusts to different screen sizes. Simple UI: Users can process for job applications and feedback.

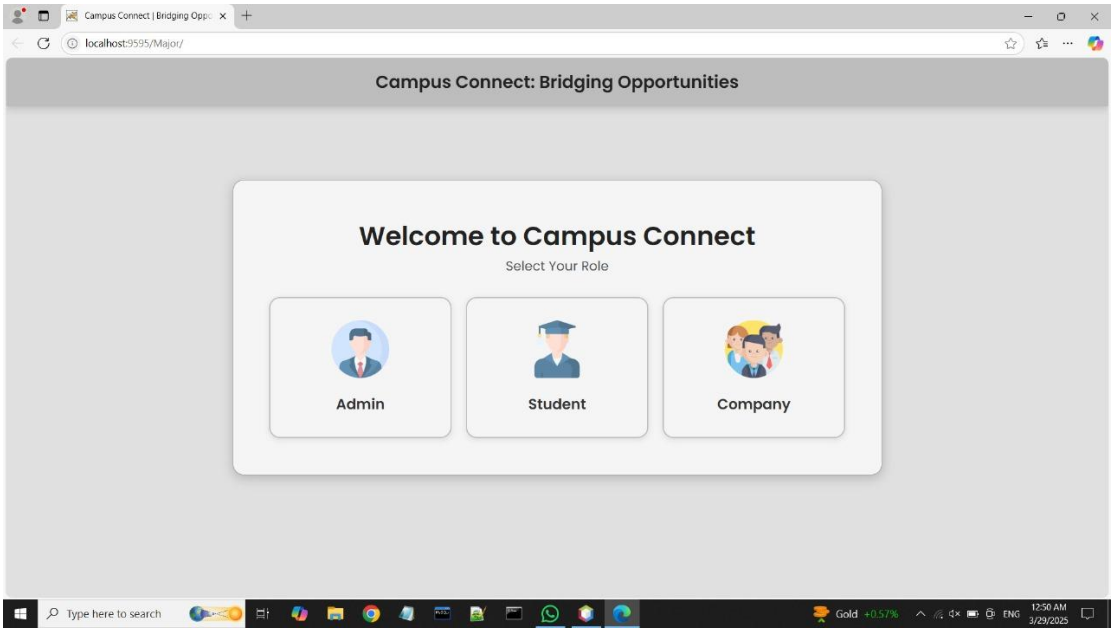

Fig.3.5 Campus Connect : Dashboard

3.6 Data Design

3.6.1 Schema Definitions

1. Student Registration Table (regis)

SQL Query:

```
CREATE TABLE student (
id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(100) NOT NULL,
email VARCHAR(100) UNIQUE NOT NULL,
contact VARCHAR(15) NOT NULL,
password VARCHAR(255) NOT NULL,
status ENUM('PENDING', 'APPROVED', 'REJECTED') DEFAULT 'PENDING'
);
```

Explanation:

- **Purpose**: Stores information for each registered user.

- **Fields**:
    - id: Unique identifier for each user, automatically incremented.
    - name: User's full name, up to 50 characters.
    - password: Hashed password, ensuring security.
    - email: User's email, unique to prevent duplicates.
    - contact: Optional phone contact information.
    - address: Optional address field for user location.

2. Admin Registration Table (admin)

SQL Query:

```
CREATE TABLE admin (
id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(100) NOT NULL,
email VARCHAR(100) NOT NULL UNIQUE,
contact VARCHAR(10) NOT NULL UNIQUE,
password VARCHAR(255) NOT NULL,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Explanation:

- **Purpose**: Manages details of registered administrators.
- **Fields**:
    - id: Unique identifier for each admin, with auto-increment.
    - name: Admin's name, up to 50 characters.
    - password: Hashed password for secure login.
    - email: Admin's email, unique for login purposes.
    - contact: Optional contact number for the admin.

- address: Optional address.

3.Company Registration Table (company)

SQL Query:

```
CREATE TABLE company (
id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(100) NOT NULL,
email VARCHAR(100) UNIQUE NOT NULL,
contact VARCHAR(15) NOT NULL,
password VARCHAR(255) NOT NULL,
status ENUM('Pending', 'Approved', 'Rejected') DEFAULT 'Pending'
);
```

Explanation:

- **Purpose**: Manage Details of registered company.
- **Fields**:
  - id: Unique issue identifier.
  - name: Name of the company registering.
  - contact: Contact number for follow-up.
  - address: Location details.
  - location: Specific location for the issue within the service area.
  - status: Tracks the status.
  - timestamp: Auto-populates with date and time.

4. Job Table (jobs)

SQL Query:

```
CREATE TABLE jobs (
id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    title VARCHAR(255) NOT NULL,
    description TEXT NOT NULL,
    location VARCHAR(255) NOT NULL,
    salary VARCHAR(50) NOT NULL,
    status ENUM('Pending', 'Approved', 'Rejected') DEFAULT 'Pending'
);
```

Explanation:

- **Purpose**: Stores jobs provided by company.
- **Fields**:
    - id: unique id.
    - name: Name of the company posting jobs.
    - email: Contact email for potential follow-up.
    - submitted_at: Submission date and time for tracking.

5. Jobs applied Table (applied_jobs)

SQL Query

```
    CREATE TABLE applied_jobs (id INT AUTO_INCREMENT PRIMARY KEY
    student_id INT NOT NULL,
    job_id INT NOT NULL,
    status ENUM('Pending', 'Approved', 'Rejected') DEFAULT 'Pending',
    applied_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (student_id) REFERENCES student(id) ON DELETE CASCADE,
    FOREIGN KEY (job_id) REFERENCES jobs(id) ON DELETE CASCADE
);
```

Explanation:

- **Purpose**: Manages data related to applied jobs.

- **Fields**:
    - id: Unique student id.
    - name: Name of the student.
    - contact: Contact number.
    - address: Address of the person reporting.
    - location: Location details for resolving the issue.
    - status: Current status, with a default of “Open.
    - timestamp: Automatically records when the issue is reported.

6. Job application Table (job_applications)

SQL Query:

```
CREATE TABLE job_applications (
id INT PRIMARY KEY AUTO_INCREMENT,
job_id INT,
student_name VARCHAR(255),
student_email VARCHAR(255),
resume TEXT,
FOREIGN KEY (job_id) REFERENCES jobs(id) ON DELETE CASCADE
);
ALTER TABLE job_applications CHANGE job_id id INT;
DROP TABLE job_applications;
```

Explanation:

- **Purpose**: Stores jobs applied by the students.
- **Fields**:
    - id: Unique ID.
    - name: Applicant’s name.
    - email: Email for follow-up.
    - timestamp: Records the submission time.

7. Job applications Tabke (job_applications)

SQL Query:

```
ALTER TABLE job_applications CHANGE job_id id INT;

DROP TABLE job_applications;

ALTER TABLE job_applications

ADD CONSTRAINT fk_job FOREIGN KEY (job_id) REFERENCES jobs(id) ON DELETE CASCADE;

CREATE TABLE job_applications (
   id INT PRIMARY KEY AUTO_INCREMENT,
   job_id INT,
   student_name VARCHAR(255),
   student_email VARCHAR(255),
   resume TEXT,
   status VARCHAR(20) DEFAULT 'Pending',
   FOREIGN KEY (job_id) REFERENCES jobs(id) ON DELETE CASCADE
);
```

Explanation:

- **Purpose**: Change job applications.
- **Fields**:
    - id: Unique identifier .
    - name: Changer name.
    - contact: Contact number for follow-up.
    - address: Address for reference.
    - created_at: Submission date.

3.6.2 ER Diagram

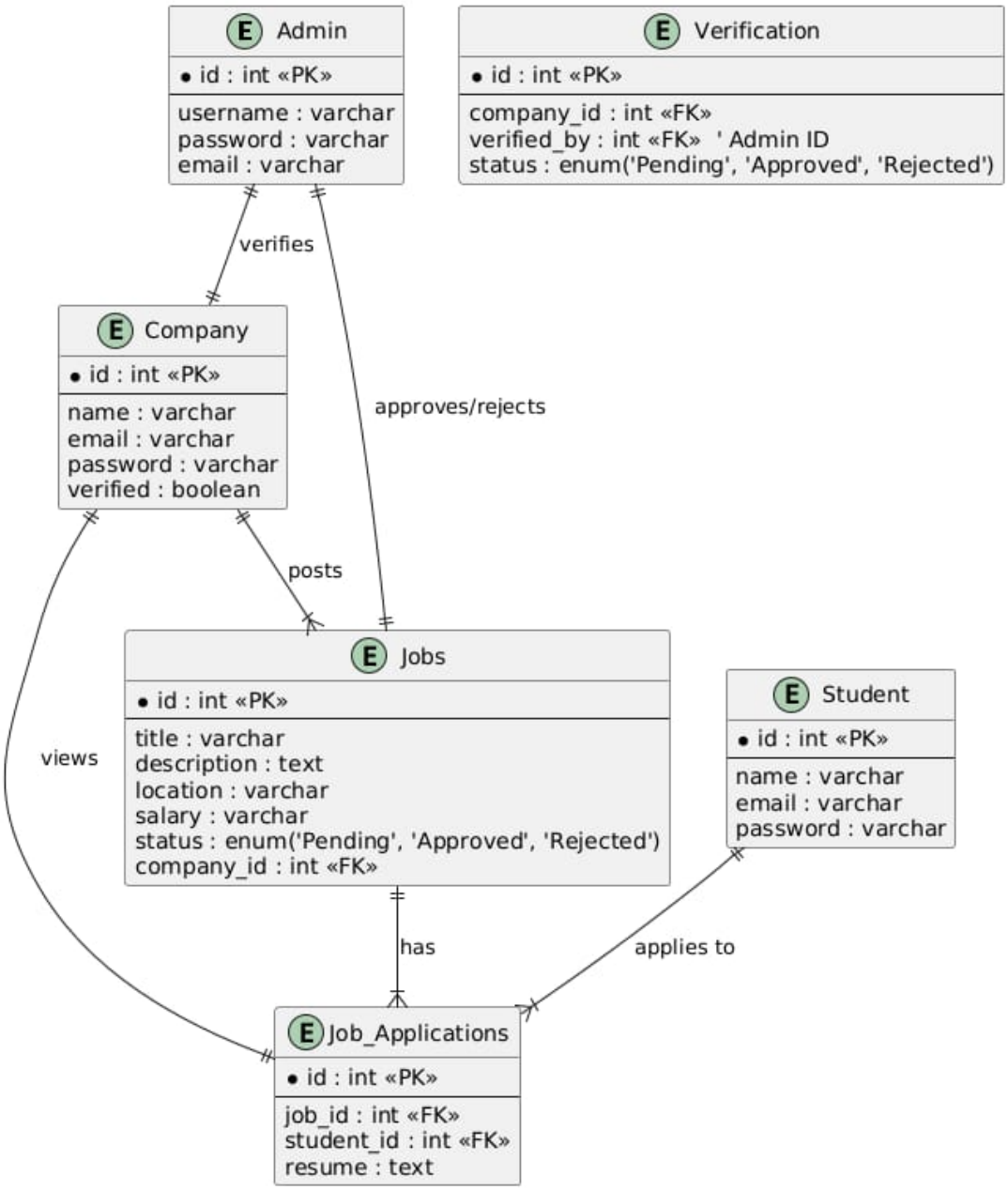

Fig.3.6.2 ER diagram

IMPLEMENTATAION AND TESTING

4.1 Methodology

The development of the Campus Connect platform follows a structured methodology to ensure efficiency, scalability, and security. It begins with requirement analysis, where functional and technical needs are gathered from stakeholders, including colleges, companies, and students. Next, in the system design phase, the three-tier architecture (frontend, backend, and database) is defined, technologies such as Java, MySQL, HTML, CSS, and JavaScript are selected, and security measures are planned. The development phase follows an Agile approach, allowing for iterative improvements and continuous integration. Thorough testing is conducted, including unit, integration, and user acceptance testing (UAT), to ensure reliability and performance.Finally, ongoing maintenance and updates ensure bug fixes, feature enhancements, and scalability to meet growing demands. This structured approach ensures a secure, efficient, and user-friendly recruitment platform.

4.1.1 Proposed Algorithm

Proposed Algorithm for Campus Recruitment System

1. User Registration and Login
    - User registers by providing details; system validates and stores in Users table.
    - User logs in; credentials are verified, granting access to the dashboard.
2. Job Posting
    - Company fills job details (title, description, location, salary); system validates and saves to Jobs table with "Open" status.
3. Job Application
    - Student applies for a job by submitting their resume; system saves in Applications table with "Pending" status.
4. Status Tracking
    - Student views all applied jobs and their statuses on the dashboard.
    - System sends notifications of any status updates.

5. Application Shortlisting
    - Admin or Company shortlists applications; system updates Applications table with new status ("Shortlisted").
6. Notification System
    - Notifications sent to users for job updates, application status changes, and feedback via in-app or email, ensuring transparency.

4.2 Implementation Approach

The development used the following technologies:

The project uses HTML, CSS, JavaScript, and Java Servlets, with MySQL for the database. Validation is handled via JavaScript.

4.2.1 Introduction to Languages, IDES, tools and Technology

Java: Java is a widely-used, high-level programming language known for its portability, security, and robustness. It follows the object-oriented programming (OOP) paradigm, making it easy to organize and structure code through objects and classes. Java programs are compiled into bytecode, which can be executed on any device with a Java Virtual Machine (JVM), enabling "write once, run anywhere" functionality. This platform independence is a key advantage, allowing Java applications to run on diverse systems without modification. Java prioritizes security with features like the Java sandbox and security manager, which protect against unauthorized code execution and ensure safe program operation. It includes strong memory management, exception handling, and multithreading support, making it suitable for building scalable and reliable applications. Java's rich standard library offers a vast array of utilities and APIs for common tasks, from basic I/O operations to advanced networking and GUI development. Its combination of simplicity, performance optimization through JIT compilation, and a robust development ecosystem makes Java a preferred choice for enterprise applications, Android development, and web services.

2.HTML: HTML (Hypertext Markup Language) is the standard language used to create and design web pages. It provides a set of elements or tags that structure the content of a webpage. Each HTML tag represents a different part of the content, such

as headings, paragraphs, links, images, and forms. Tags are enclosed in angle brackets, < >, and usually come in pairs: an opening tag and a closing tag, with content nested between them. For example, and are tags used to define a paragraph. HTML is not a programming language but rather a markup language that defines the structure and layout of a webpage, including text, images, and other elements. Modern web development often combines HTML with CSS (Cascading Style Sheets) for styling and JavaScript for interactivity, forming the core technologies of the World Wide Web.

3.CSS: CSS (Cascading Style Sheets) is a fundamental technology used for styling and formatting web pages written in HTML and XML. It allows developers to control the appearance of web pages by defining styles for various HTML elements. CSS separates the content of a webpage from its design, enabling a more flexible and efficient approach to web development. Styles are applied to HTML elements using selectors, which can target elements based on their type, class, ID, or other attributes. Each CSS rule consists of a selector and a declaration block, where properties and values define how the selected elements should be styled. For example, CSS can define fonts, colors, margins, padding, and layout, making it possible to create visually appealing and responsive web pages.

4.NetBeans: NetBeans is an open-source Integrated Development Environment (IDE) primarily used for Java development, though it supports other languages like PHP, HTML5, C++, and JavaScript. Known for its intuitive user interface, NetBeans offers a comprehensive suite of tools that streamline coding, debugging, and project management. One of its key features is the drag-and-drop GUI builder, which makes designing user interfaces much easier. Additionally, NetBeans supports numerous plugins, allowing developers to extend its functionality based on specific needs. The IDE also provides real-time error detection and syntax highlighting, which improves coding accuracy and efficiency. Its robust code refactoring tools enable quick and safe modifications across large projects. NetBeans integrates seamlessly with version control systems like Git, allowing collaborative work and efficient code management. Furthermore, it’s cross-platform, running on Windows, macOS, and Linux, which increases its accessibility.

5.JDBC: JDBC (Java Database Connectivity) is an essential Java API that enables Java applications to interact with relational databases. It provides a uniform interface for accessing and managing data stored in databases, regardless of the specific database

management system (DBMS) being used. JDBC allows developers to perform database operations such as querying data, inserting or updating records, and executing stored procedures directly from Java code.

The core components of JDBC include the DriverManager, which handles the loading and registering of database drivers, and the Connection interface, which represents a connection to a specific database. Developers use the Connection interface to create Statement objects for executing SQL queries and updates, or PreparedStatement objects for precompiled queries that improve performance and security.

6.Servlet: Servlets are Java programming language components that dynamically process and respond to requests from web clients, usually browsers, by generating web content. They are server-side programs running inside a web server that handle client requests and provide responses. Servlets are part of the Java Enterprise Edition (Java EE) technology stack and are typically used to implement web applications.

4.3Testing Approaches

4.3.1 Unit Testing

a. Test Cases

TABLE I

TEST CASES FOR CAMPUS CONNECT

Test Case Id	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result
UU01	Create Admin Registration Page	1. Open website and click on Admin Registration	Admin details (name, email, password, contact)	Data is successfully added in admin table	As expected
UU02	Create Student Registration Page	1. Open website and click on Student Registration	Student details (name, email, password, contact, resume)	Data is successfully added in student table	As expected
UU03	Create Company Registration Page	1. Open website and click on Company Registration	Company details (name, email, password, contact)	Data is successfully added in company table	As expected

UU04	Create Admin Login Page	1. Open website and click on Admin Login 2. Enter username and password	Admin enters valid credentials	Data is fetched from admin table	As expected
UU05	Create Student Login Page	1. Open website and click on Student Login 2. Enter username and password	Student enters valid credentials	Data is fetched from student table	As expected
UU06	Create Company Login Page	1. Open website and click on Company Login 2. Enter username and password	Company enters valid credentials	Data is fetched from company table	As expected
UU07	Admin Approval for Student	1. Admin logs in 2. Admin views pending students 3. Admin approves/rejects a student	Student ID for approval/rejection	Student status updates in student table	As expected
UU08	Admin Approval for Company	1. Admin logs in 2. Admin views pending companies 3. Admin approves/rejects a company	Company ID for approval/rejection	Company status updates in company table	As expected
UU09	Student Applies for Job	1. Student logs in 2. Student selects a job 3. Clicks on 'Apply'	Student selects job ID	Application is added in application table	As expected
UU10	Company Posts a Job	1. Company logs in 2. Company fills job details	Job details (title, description, location, salary, type)	Data is successfully added in job table	As expected

		3. Clicks on 'Post Job'			
UU11	Admin Views Job Applications	1. Admin logs in 2. Admin selects a job 3. Views list of applicants	Job ID	List of applicants fetched from application table	As expected
UU12	Company Views Job Applications	1. Company logs in 2. Company selects a job 3. Views list of applicants	Job ID	List of applicants fetched from application table	As expected
UU13	Admin/Company Shortlists Candidates	1. Admin/Company logs in 2. Selects an application 3. Updates application status	Application ID	Status updated in application table	As expected
UU14	Student Gives Feedback for Company	1. Student logs in 2. Selects company 3. Provides rating and comment	Feedback details (rating, comment, company ID)	Data is successfully added in feedback table	As expected
UU15	Admin Views Feedback for Companies	1. Admin logs in 2. Selects a company 3. Views feedback	Company ID	Feedback fetched from feedback table	As expected
UU16	Company Gives Feedback for Student	1. Company logs in 2. Selects student 3. Provides feedback	Feedback details (rating, comment, student ID)	Data is successfully added in feedback table	As expected
UU17	Search Jobs Feature	1. User logs in 2. Uses search bar to find jobs	Keywords for job title/location	Relevant job results displayed	As expected

				from job table	
UU18	Search Candidates Feature	1. Company logs in 2. Uses search bar to find candidates	Keywords for student skills	Relevant candidates displayed from student table	As expected
UU19	Admin Views All Data	1. Admin logs in 2. Clicks on View All Data	-	Admin can see Students, Companies, Jobs, Applications, Feedback	As expected

RESULTS AND CONCLUSION

5.1 User Interface Representation

5.1.1 Brief Description of Various Modules

1. User Management Module

This module allows different types of users to register and log in to the system.

Features:

Admin Registration & Login

Student Registration & Login

Company Registration & Login

Description:

New users (admin, student, company) can create an account by providing details like name, email, password, and contact.

Once registered, they can log in using their credentials.

Upon login, users can access the respective functionalities assigned to their role.

2. Admin Management Module

This module allows the admin to manage users and oversee platform activities.

Features:

Admin Approval for Student Accounts

Admin Approval for Company Accounts

Admin Views Job Applications

Admin Shortlists Candidates

Admin Views Feedback for Companies

Admin Views All Data

Description:

Admins can approve/reject students and companies after reviewing their profiles.

Admins can view and manage job applications.

Admins can check feedback given by students for companies.

They have full access to all system data, including registered users, job listings, applications, and feedback.

3. Student Management Module

This module allows students to apply for jobs and give feedback.

Features:

Student Applies for a Job

Student Gives Feedback for Company

Description:

Registered students can search and apply for jobs.

Students can provide ratings and comments for companies they interact with.

4. Company Management Module

This module enables companies to post jobs, manage applications, and give feedback.

Features:

Company Posts a Job

Company Views Job Applications

Company Shortlists Candidates

Company Gives Feedback for Students

Description:

Companies can create and post job listings.

They can view job applications and shortlist candidates.

They can provide ratings and comments for students.

5. Job Search & Application Module

This module allows users to search for jobs and companies to find suitable candidates.

Features:

Search Jobs Feature

Search Candidates Feature

Description:

Students can use keywords to search for jobs based on title, location, or company.

Companies can search for candidates based on skills and qualifications.

This structure ensures that different users can interact efficiently within the system, making it a complete job placement and recruitment platform.

5.2 Snapshot of System with Brief Description

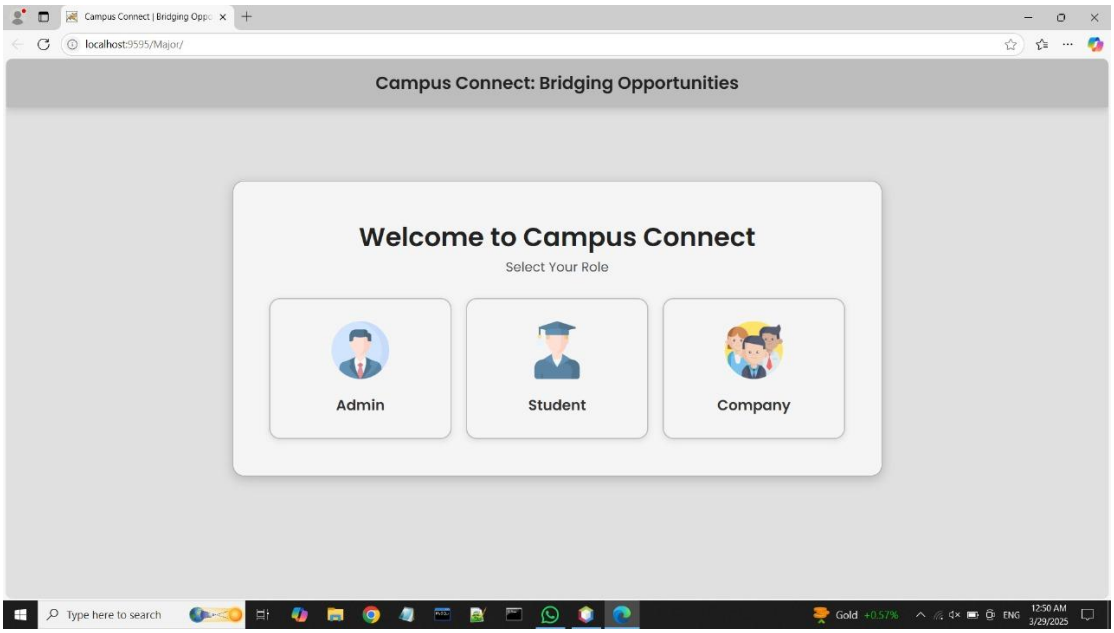

Fig 5.2(a) Campus Connect : Role Selection Page

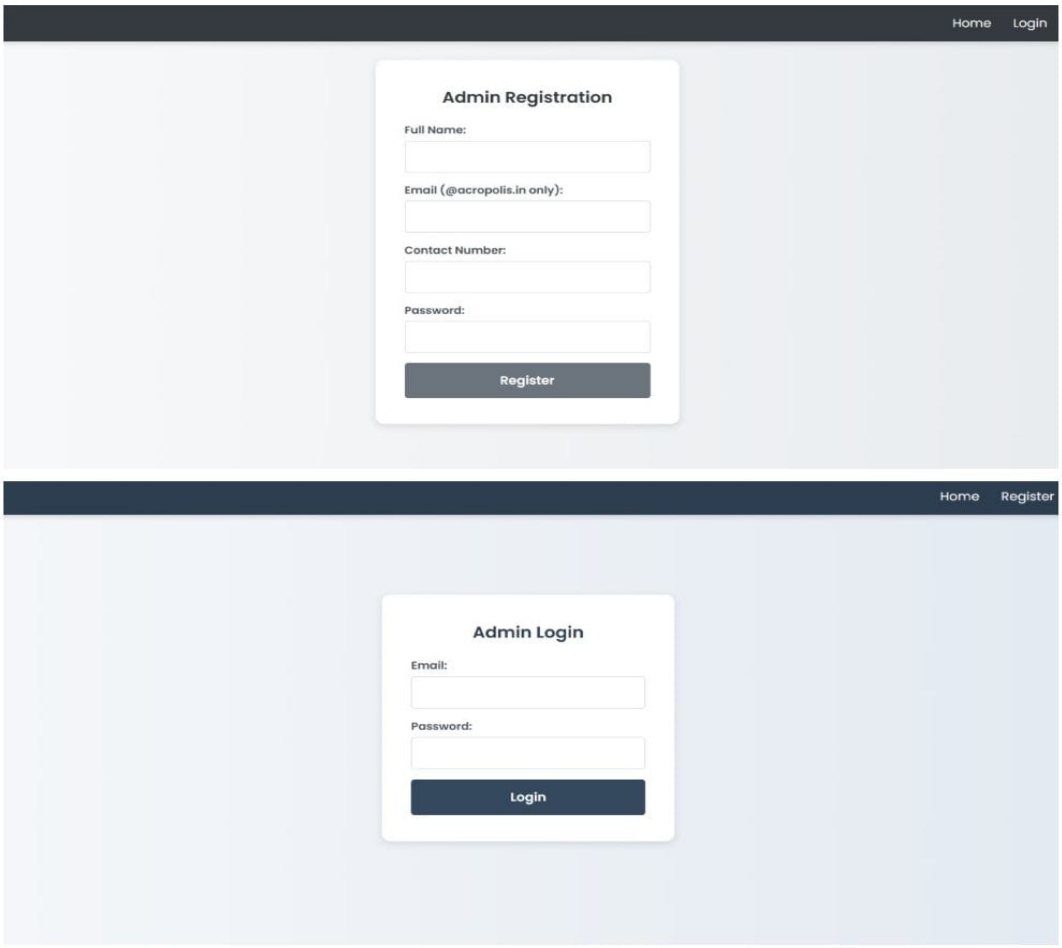

Fig.5.2(b) Admin Registration and Login Page

Fig.5.2(c) Admin Dashboard: Pending Approvals

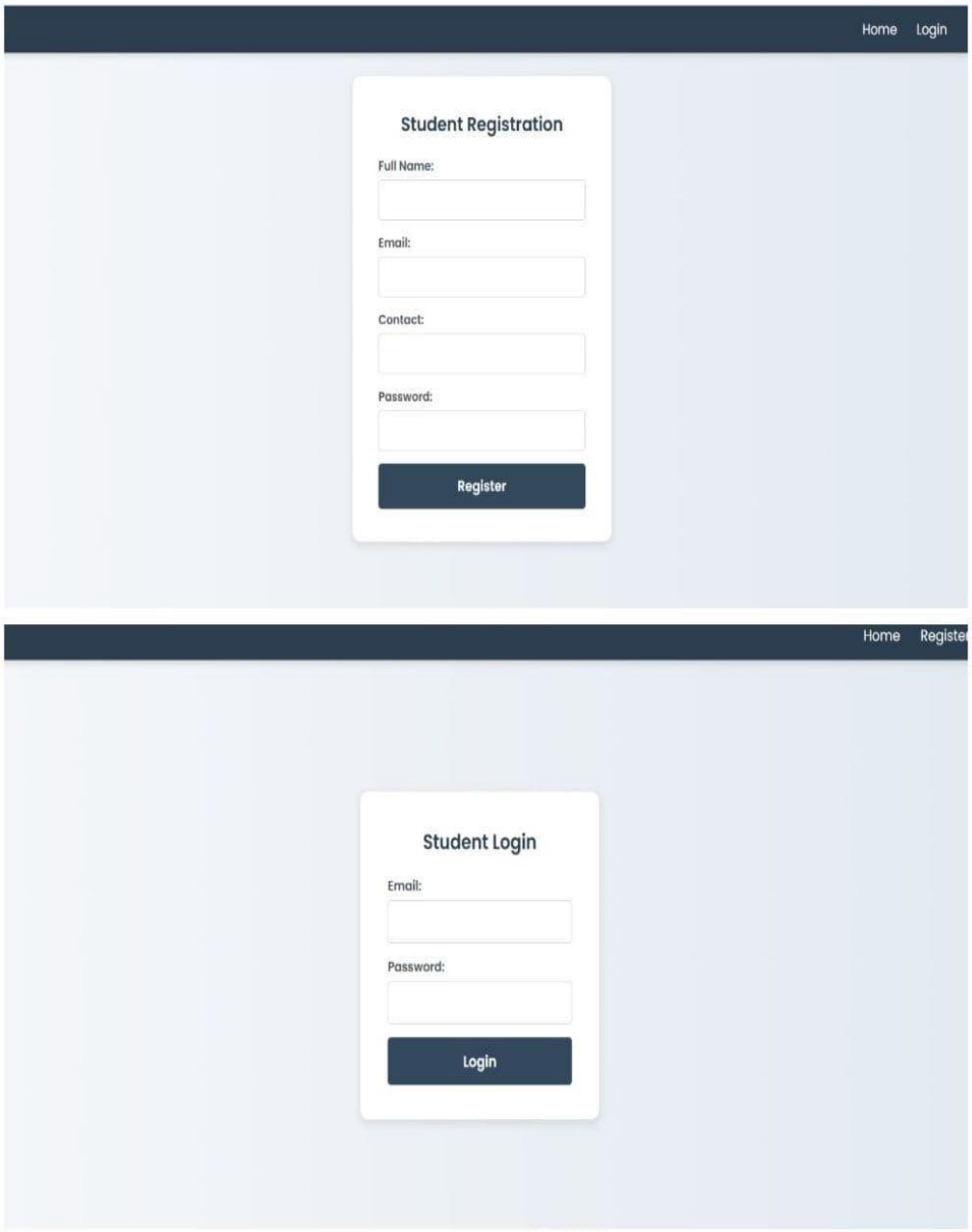

Fig 5.2(d) Student Registration and Login Page

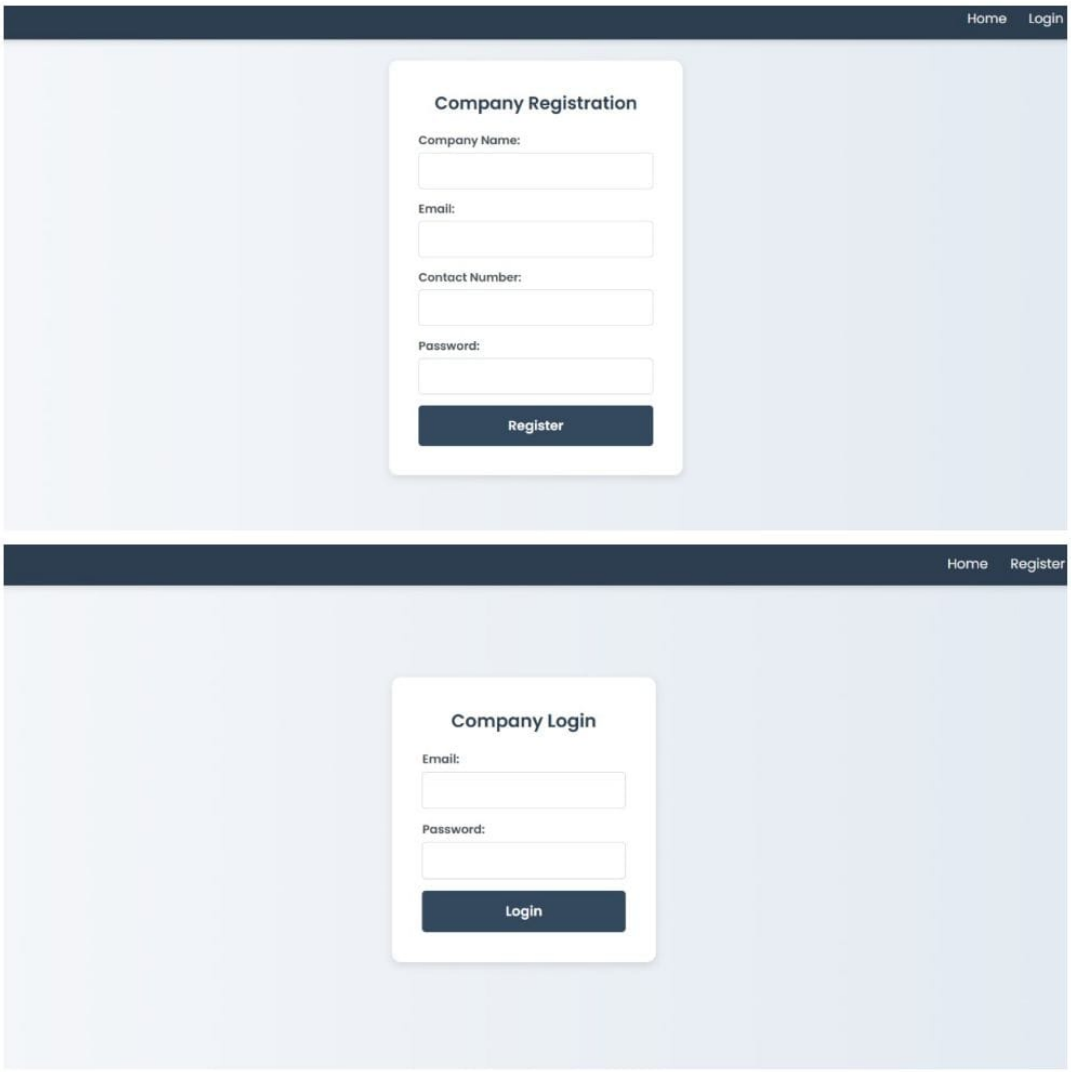

Fig 5.2(e) Company Registration and Login Page

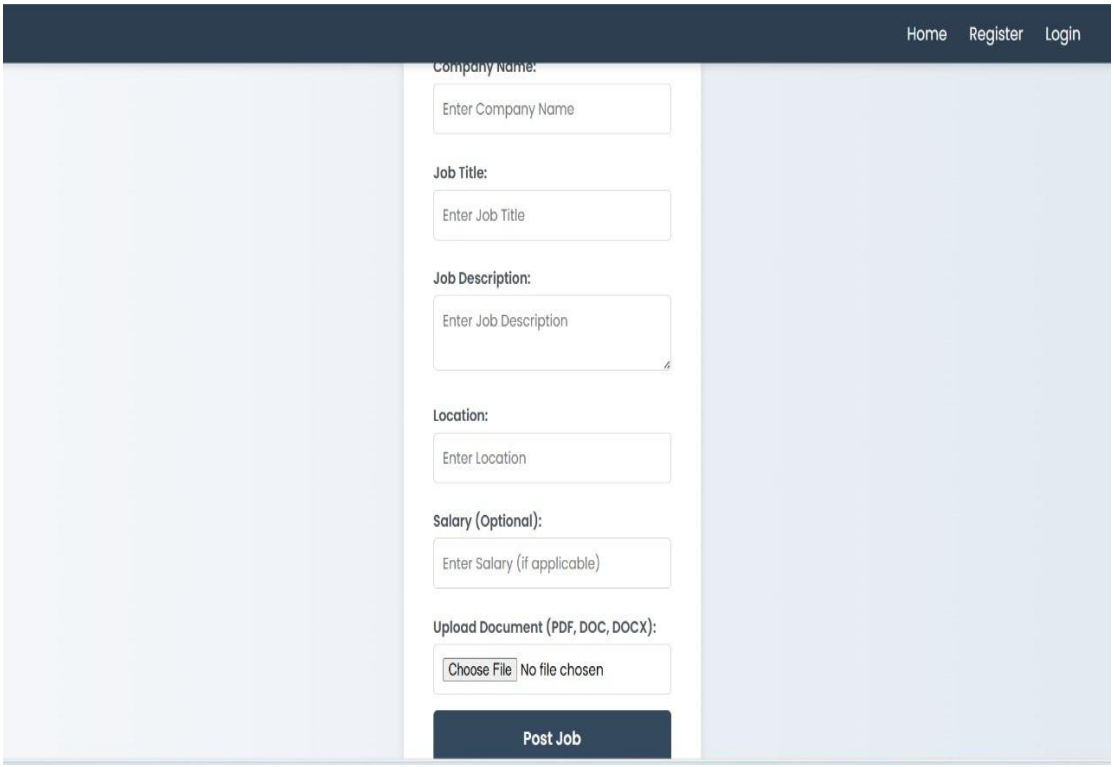

Fig 5.2(f) Job Posted by Company

5.3 Database Description

The database schema consists of several key tables that store critical information for an online recruitment platform. The admin table contains details of the platform's administrators, including login credentials and roles. The student table holds information about students, such as personal details, academic records, and enrolled programs. The company table stores data on companies looking to hire, including company profiles, industry types, and job listings. The jobs table tracks available job positions, while the job application table manages the applications submitted by students for various job openings. Together, these tables form the backbone of the system, ensuring smooth interaction between all users.

5.3.1 Snapshot of database table with brief description

TABLE II

ADMIN AND STUDENT REGISTRATION

```
mysql> desc admin;
+------------+--------------+------+-----+-------------------+-------------------+
| Field      | Type         | Null | Key | Default           | Extra             |
+------------+--------------+------+-----+-------------------+-------------------+
| id         | int          | NO   | PRI | NULL              | auto_increment    |
| name       | varchar(100) | NO   |     | NULL              |                   |
| email      | varchar(100) | NO   | UNI | NULL              |                   |
| contact    | varchar(10)  | NO   | UNI | NULL              |                   |
| password   | varchar(255) | NO   |     | NULL              |                   |
| created_at | timestamp    | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+------------+--------------+------+-----+-------------------+-------------------+
6 rows in set (0.21 sec)

mysql> desc student;
+----------+-----------------------------------------+------+-----+---------+----------------+
| Field    | Type                                    | Null | Key | Default | Extra          |
+----------+-----------------------------------------+------+-----+---------+----------------+
| id       | int                                     | NO   | PRI | NULL    | auto_increment |
| name     | varchar(100)                            | NO   |     | NULL    |                |
| email    | varchar(100)                            | NO   | UNI | NULL    |                |
| contact  | varchar(15)                             | NO   |     | NULL    |                |
| password | varchar(255)                            | NO   |     | NULL    |                |
| status   | enum('PENDING','APPROVED','REJECTED')   | YES  |     | PENDING |                |
+----------+-----------------------------------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

COMPANY REGISTRATION

```
mysql> desc company;
+----------+-----------------------------------------+------+-----+---------+----------------+
| Field    | Type                                    | Null | Key | Default | Extra          |
+----------+-----------------------------------------+------+-----+---------+----------------+
| id       | int                                     | NO   | PRI | NULL    | auto_increment |
| name     | varchar(100)                            | NO   |     | NULL    |                |
| email    | varchar(100)                            | NO   | UNI | NULL    |                |
| contact  | varchar(15)                             | NO   |     | NULL    |                |
| password | varchar(255)                            | NO   |     | NULL    |                |
| status   | enum('Pending','Approved','Rejected')   | YES  |     | Pending |                |
+----------+-----------------------------------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)

mysql> desc jobs;
+-------------+-----------------------------------------+------+-----+---------+----------------+
| Field       | Type                                    | Null | Key | Default | Extra          |
+-------------+-----------------------------------------+------+-----+---------+----------------+
| id          | int                                     | NO   | PRI | NULL    | auto_increment |
| title       | varchar(255)                            | NO   |     | NULL    |                |
| description | text                                    | NO   |     | NULL    |                |
| location    | varchar(255)                            | NO   |     | NULL    |                |
| salary      | varchar(50)                             | NO   |     | NULL    |                |
| status      | enum('Pending','Approved','Rejected')   | YES  |     | Pending |                |
+-------------+-----------------------------------------+------+-----+---------+----------------+
```

TABLE III

JOBS POSTED BY COMPANY

```
mysql> desc jobs;
+--------------+-------------------------------------+------+-----+---------+----------------+
| Field        | Type                                | Null | Key | Default | Extra          |
+--------------+-------------------------------------+------+-----+---------+----------------+
| id           | int                                 | NO   | PRI | NULL    | auto_increment |
| title        | varchar(255)                        | NO   |     | NULL    |                |
| description  | text                                | NO   |     | NULL    |                |
| location     | varchar(255)                        | NO   |     | NULL    |                |
| salary       | varchar(50)                         | NO   |     | NULL    |                |
| status       | enum('Pending','Approved','Rejected') | YES  |     | Pending |                |
| company_name | varchar(255)                        | NO   |     | NULL    |                |
| document_path | varchar(255)                       | YES  |     | NULL    |                |
+--------------+-------------------------------------+------+-----+---------+----------------+
8 rows in set (0.02 sec)
```

TABLE IV

JOB APPLIED BY THE STUDENTS

```
mysql> desc job_applications;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| id            | int          | NO   | PRI | NULL    | auto_increment |
| job_id        | int          | YES  | MUL | NULL    |                |
| student_name  | varchar(255) | YES  |     | NULL    |                |
| student_email | varchar(255) | YES  |     | NULL    |                |
| resume        | text         | YES  |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

TABLE V

DATABASE TABLES OF CAMPUS CONNECT

```
mysql> show tables;
+------------------+
| Tables_in_major  |
+------------------+
| admin            |
| company          |
| job_applications |
| jobs             |
| student          |
+------------------+
5 rows in set (0.97 sec)
```

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

Campus Connect: Bridging Opportunities offers an efficient solution to the challenges faced by traditional campus recruitment processes. By automating communication and providing a centralized platform for colleges, students, and companies, the system improves the overall experience for all parties involved. Colleges can better manage recruitment drives, students can apply for jobs with ease and track their progress, while companies gain access to a more streamlined and effective way of recruiting qualified candidates. The platform's eligibility tracking and automated notifications ensure that only the most relevant opportunities reach the right students, reducing time spent on mismatched applications and improving the quality of the hiring process. Overall, Campus Connect optimizes recruitment efficiency, enhances transparency, and simplifies coordination, benefiting all stakeholders

6.2 Future Scope

The future scope of the Campus Connect platform includes several areas for enhancement and expansion:

Integration with Other Platforms: The platform can be integrated with other job portals and professional networks like LinkedIn, Naukri, and Indeed, allowing for broader exposure and easier application management.

AI-Based Job Matching: By incorporating Artificial Intelligence (AI), the system can offer more advanced features like personalized job recommendations based on students skills, interests, and past application history.

Mobile App Development: The development of a mobile application would allow students and companies to stay connected on-the-go, receive real-time notifications, and track application statuses easily from their smartphones.

Alumni Networking: The platform can be expanded to include alumni, allowing them to offer mentorship, job opportunities, or participate in recruitment activities, creating a more robust network for students and companies.

REFERENCES

[1] LinkedIn. (2024). Professional Networking and Job Search . Available: https://www.linkedin.com

[2] Naukri Campus. (2024). Campus Hiring and Placement Solutions . Available: https://campus.naukri.com

[3] Remotive. (2024). Remote Job Board and Hiring Platform . Available: https://remotive.io

[4] Indeed. (2024). Job Search and Company Reviews. Available: https://www.indeed.com

Guided By:

Prof. Prashant Lakkadwala

Group members:

Jahnavi Mandloi(0827IT211048)

https://github.com/jahnavimandloi50

Jayshree Dave(0827IT211050)

https://github.com/Jayshreedave17

Mahee Dubey (0827IT211065)

https://github.com/mahidubey2003

Minakshi Soni(0827IT211069)

https://github.com/Minakshi-26