

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  
**“JNANA SANGAMA”, BELAGAVI - 590 018**



**A MINI PROJECT REPORT**  
on  
**“Campus Placement Management System”**

*Submitted by*

Mohammed Nihal	4SF23CI090
Samarth	4SF23CI129
Anees Hasan	4SF24CI400
Manvith B M	4SF22CI048

*In partial fulfillment of the requirements for the V semester*

**DATABASE MANAGEMENT SYSTEMS**

of

**BACHELOR OF ENGINEERING**

in

**COMPUTER SCIENCE AND ENGINEERING**

(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

*Under the Guidance of*

**Mrs. Sangeetha M S**

Assistant Professor, Department of CSE(AI&ML)

at



**SAHYADRI**  
College of Engineering & Management  
An Autonomous Institution  
**MANGALURU**

**2025 - 26**

**SAHYADRI**  
**College of Engineering & Management**  
**An Autonomous Institution**  
**MANGALURU**

**COMPUTER SCIENCE AND ENGINEERING**

**(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)**



**CERTIFICATE**

This is to certify that the Mini Project entitled “Campus Placement Management System” has been carried out by Mohammed Nihal (4SF23CI090), Samarth (4SF23CI129), Anees Hasan (4SF24CI400) and Manvith B M (4SF22CI048), the bonafide students of Sahyadri College of Engineering & Management in partial fulfillment of the requirements for the V semester Database Management Systems (AM522I1A) of Bachelor of Engineering in Computer Science and Engineering(AI&ML) of Visvesvaraya Technological University, Belagavi during the year 2025-26. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work.

---

**Mrs. Sangeetha M S**  
Assistant Professor  
Dept. of CSE(AI&ML), SCEM

---

**Dr. Pushpalatha K**  
Professor & Head  
Dept. of CSE(AI&ML), SCEM

**SAHYADRI**  
**College of Engineering & Management**  
**An Autonomous Institution**  
**MANGALURU**

**Department of Computer Science and Engineering**  
**(Artificial Intelligence and Machine Learning)**

**DECLARATION**

We hereby declare that the entire work embodied in this Mini Project Report titled “Campus Placement Management System” has been carried out by us at Sahyadri College of Engineering and Management, Mangaluru under the supervision of Mrs. Sangeetha M S as the part of the V semester Database Management Systems (AM522I1A) of Bachelor of Engineering in Computer Science and Engineering(AI&ML). This report has not been submitted to this or any other University.

Mohammed Nihal (4SF23CI090)

Samarth (4SF23CI129)

Anees Hasan (4SF24CI400)

Manvith B M (4SF22CI048)

SCEM, Mangaluru

# Abstract

The Campus Placement Management System is a web-based application designed to streamline and digitize the placement activities of colleges and universities. It provides a centralized platform where students can register, update their academic and personal details, browse available job opportunities, and track their application status, while recruiters and training-and-placement officers (TPOs) can efficiently manage company profiles, job postings, and selection processes. By automating routine tasks and maintaining structured placement records, the system reduces manual paperwork, minimizes errors, and offers real-time access to placement information for all stakeholders.

This project uses MySQL as the backend database and PHP as the server-side scripting language, ensuring secure storage of data and reliable transaction processing. The design follows principles of data normalization and relational integrity to maintain consistent student, company, and placement records across the system. By providing role-based access for students, companies, and administrators, the Campus Placement Management System offers a user-friendly platform that improves coordination, enhances transparency, and supports data-driven decision-making in campus recruitment.

# Acknowledgement

It is with great satisfaction and euphoria that we are submitting the Mini Project Report on “**Campus Placement Management System**”. We have completed it as a part of the V semester **Database Management Systems (AM522I1A)** of **Bachelor of Engineering in Computer Science and Engineering(AI&ML)** of Visvesvaraya Technological University, Belagavi.

We are profoundly indebted to our guide, **Mrs. Sangeetha M S**, Assistant Professor, Department of Computer Science and Engineering(AI&ML) for innumerable acts of timely advice, encouragement and We sincerely express our gratitude.

We express our sincere gratitude to **Dr. Pushpalatha K**, Professor & Head, Department of CSE(AI&ML) for her invaluable support and guidance.

We sincerely thank **Dr. S. S. Injaganeri**, Principal, Sahyadri College of Engineering & Management, who have always been a great source of inspiration.

Finally, yet importantly, We express our heartfelt thanks to our family & friends for their wishes and encouragement throughout the work.

**Mohammed Nihal( 4SF23CI090)**

**Samarth(4SF23CI129)**

**Anees Hasan( 4SF24CI400)**

**Manvith B M(4SF22CI048)**

V Sem, B.E., CSE(AI&ML)

SCEM, Mangaluru

# Table of Contents

<b>Abstract</b>	i
<b>Acknowledgement</b>	ii
<b>Table of Contents</b>	iv
<b>List of Figures</b>	v
<b>List of Tables</b>	vi
<b>1 Introduction</b>	1
<b>2 Requirements Specification</b>	2
2.1 Hardware Specification . . . . .	2
2.2 Software Specification . . . . .	2
<b>3 System Design</b>	3
3.1 ER Diagram . . . . .	3
3.2 Assumptions . . . . .	4
3.3 Mapping From ER Diagram to Schema Diagram . . . . .	4
3.4 Schema Diagram . . . . .	5
3.5 Normalization . . . . .	6
3.6 Implementation . . . . .	7
3.6.1 User Management Module . . . . .	7
3.6.2 Student Management Module . . . . .	7
3.6.3 Company and Job Management Module . . . . .	8
3.6.4 Application Management Module . . . . .	8
3.6.5 Interview Management Module . . . . .	9
3.7 Tables Used . . . . .	9
<b>4 Results and Discussion</b>	11



# List of Figures

3.1	ER Diagram for Campus Placement Management System . . . . .	3
3.2	Schema Diagram for Campus Placement Management System . . . . .	5
4.1	Student Dashboard . . . . .	11
4.2	Job Listings Page . . . . .	12
4.3	Applications Page . . . . .	13
4.4	TPO Dashboard . . . . .	14
4.5	Interview Management Screens . . . . .	15

# List of Tables

3.1	Work Flow of Campus Placement Management System Development . . . . .	10
3.2	Test Cases for Campus Placement Management System . . . . .	10

# Chapter 1

## Introduction

The Campus Placement Management System is a web-based solution that simplifies the campus recruitment process for colleges and universities. It tackles the problems of traditional manual placement procedures by offering a centralized digital platform where students, placement officers, college management, and companies can interact effectively. Students can register, update their profiles, upload resumes, and apply for jobs. Meanwhile, placement officers and administrators can post job openings, manage applications, schedule interviews, and track placement statistics in real time. This automation cuts down on paperwork, reduces errors, and fills communication gaps, making the process more open and accessible for everyone involved.

The main goals of this project are to manage details for students, companies, and colleges on one platform, lighten manual workloads, and provide current placement information. The system allows for multi-role access, ensuring that students, TPOs, management, and super admins have the right permissions and dashboards suited to their needs. The platform supports various functions like managing student profiles, posting job ads, handling resumes, scheduling interviews, and filtering candidates based on eligibility, ensuring that only qualified individuals are considered for each placement drive.

This project uses modern web technologies and follows a scalable design to allow for future improvements, such as SMS notifications, analytics modules, and connections with external portals. By automating routine tasks and providing immediate access to placement data, the system saves time and enhances the accuracy and fairness of the recruitment process. The proposed system is affordable, user-friendly, and adaptable, making it a valuable resource for educational institutions that want to update their campus placement processes and achieve better results for students and employers.

# **Chapter 2**

## **Requirements Specification**

### **2.1 Hardware Specification**

- Processor : Intel(R) Core(TM) i3-1005G1 CPU @ 1.20GHz 1.19 GHz
- RAM : 8GB
- Hard Disk : 1TB
- Input Device : Standard keyboard and Mouse
- Output Device : Monitor

### **2.2 Software Specification**

- Database : MySQL 5.6.17
- Markup Language : HTML
- Scripting Language: PHP 5.5.12
- IDE :NetBeans 8.0.2

# Chapter 3

## System Design

### 3.1 ER Diagram

The ER Diagram for ABC systemm is shown in fig 3.1 The ER diagram represents the

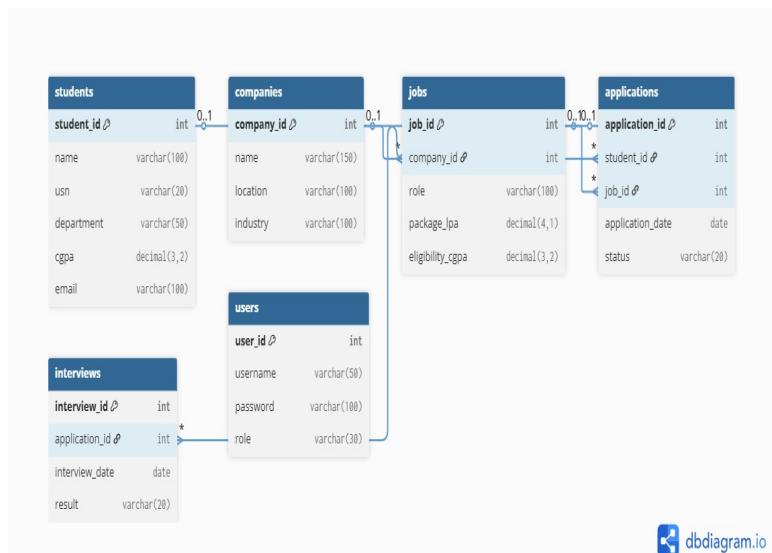


Figure 3.1: ER Diagram for Campus Placement Management System

main entities of the Campus Placement Management System and their interactions. It includes students, companies, jobs, applications, interviews, and users. Each entity is stored in a separate table with a primary key for unique identification. Students can apply for several jobs, and each job can receive applications from multiple students. The applications table acts as a bridge between students and jobs. Interviews link to specific applications, capturing the scheduled interview date and the result for each candidate. Companies can post several jobs, and authorized users manage company and placement-related tasks using their own login credentials stored in the users table.

Overall, the diagram shows clear one-to-many relationships that represent the real placement workflow, from job posting to student interviews and selection.

## 3.2 Assumptions

- One student can apply for many jobs, but each application belongs to just one student.
- One company can post many jobs, but each job is created by just one company.
- One job can receive applications from many students, and each application is linked to just one job.
- One application can have at most one interview scheduled, and each interview is for only one application.
- One student can have zero or more applications and interviews; not every student is required to apply for jobs.
- One system user, such as a TPO or admin, can manage many companies or jobs, but a job or company is linked to just one responsible user at a time.

## 3.3 Mapping From ER Diagram to Schema Diagram

Figure 3.2 shows the relational schema of the proposed Campus Placement Management System. In this system, the student, company, job, application, interview, and user entities from the ER diagram are mapped to related relational tables with the right primary and foreign keys. Each student submits applications for available jobs posted by companies. These applications are recorded in the applications table along with their status and application date, which helps establish the relationships between students and jobs. Placement administrators and approved users manage company details and job postings using the users and companies tables. This setup makes sure every job record links to a valid company and is managed by a specific user role. Based on the applications received, interviews are scheduled and noted in the interviews table. This table connects to specific applications to track the interview date and results. Therefore, the schema diagram effectively turns the conceptual ER design into relational structures that support

the entire workflow, from job creation and student applications to interview scheduling and final selection in the placement process.

### 3.4 Schema Diagram

The schema diagram of the Campus Placement Management System illustrates the logical structure of the database and shows how the entities from the ER diagram fit into relational tables. The STUDENTS table holds attributes like student\_id (primary key), name, USN, department, CGPA, and email. This setup ensures each student record is uniquely identified and their academic details are kept. The COMPANIES table includes company\_id as the primary key, along with name, location, and industry. The JOBS table has job\_id as the primary key and company\_id as a foreign key to connect each job with the company offering it, in addition to attributes such as role, package\_lpa, and eligibility\_cgpa. The APPLICATIONS table records the many-to-many relationship between students and jobs. It does this by using application\_id as the primary key and student\_id and job\_id as foreign keys, along with application\_date and status to track each application. The INTERVIEWS table has interview\_id as the primary key and application\_id as a foreign key to document scheduled interviews for specific applications, which also include interview\_date and result. Lastly, the USERS table holds user\_id, username, password, and role to manage access for different system users like administrators and placement officers. Overall, this schema diagram effectively converts the conceptual design into normalized tables with clear primary and foreign key constraints. This approach supports efficient storage, retrieval, and management of all placement-related data.

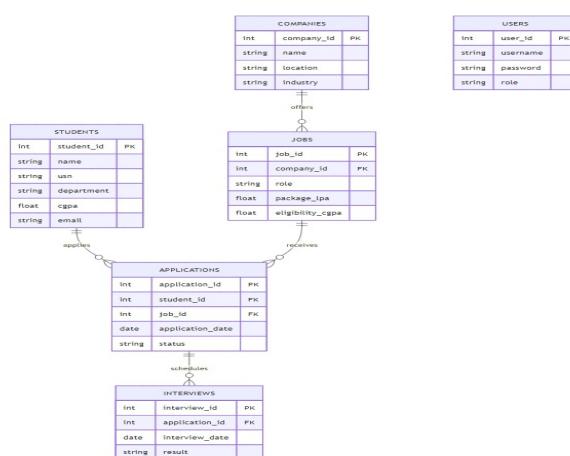


Figure 3.2: Schema Diagram for Campus Placement Management System

### 3.5 Normalization

In the Campus Placement Management System, normalization is important for organizing the database. It reduces data redundancy and maintains data integrity across all modules. In this project, key entities like Students, Companies, Jobs, Applications, Interviews, and Users are represented as separate tables. Each table holds attributes specific to that entity, ensuring the design meets the first normal form by removing repeating groups and keeping atomic attribute values. For instance, student\_id uniquely identifies each record in the STUDENTS table and does not combine with any other attribute, while details such as name, USN, department, CGPA, and email depend entirely on this single primary key. Similarly, the COMPANIES table uses company\_id as its primary key and contains only company-specific attributes like name, location, and industry, preventing the same company information from being repeated in other parts of the database.

As the design moves to second normal form, it removes all partial dependencies. Every non-key attribute depends on the entire primary key, not just a part of it. Since each table uses a single-attribute primary key (student\_id, company\_id, job\_id, application\_id, interview\_id, user\_id), no partial dependencies exist in these tables. The many-to-many relationship between students and jobs is managed using the APPLICATIONS table, where application\_id is the primary key and student\_id and job\_id are foreign keys. The INTERVIEWS table builds on this design by connecting each interview record to a specific application through application\_id and storing only interview\_id, interview\_date, and result, so changes to student or job details do not require updates in the interviews data.

Third normal form is achieved by removing transitive dependencies, where non-key attributes depend on other non-key attributes. No table contains derived or indirectly dependent attributes; for example, placement statistics or counts are calculated during queries instead of being stored redundantly. By enforcing primary and foreign key constraints between these tables, the system maintains referential integrity so that an application cannot exist without a valid student and job, and an interview cannot exist without a valid application. Overall, normalizing the Campus Placement Management System up to third normal form results in a clean, well-structured schema that avoids insertion, update, and deletion anomalies and provides a strong base for efficient queries, maintenance, and reporting on campus placements.

## 3.6 Implementation

This chapter explains the implementation of the major modules of the Campus Placement Management System along with pseudocode to illustrate the working logic. The system is divided into functional modules such as User Management, Student Management, Company and Job Management, Application Management, and Interview Management. Each module interacts with the database tables (**USERS**, **STUDENTS**, **COMPANIES**, **JOBS**, **APPLICATIONS**, **INTERVIEWS**) to perform operations efficiently and maintain data integrity.

### 3.6.1 User Management Module

The User Management Module handles the registration and authentication of system users such as administrators, TPOs, and coordinators. It ensures that each user record is securely stored in the **USERS** table with an appropriate role.

#### Pseudocode: User Registration

```
BEGIN
    INPUT username, password, role
    VALIDATE user input
    HASH the password
    INSERT user details into USERS table
    DISPLAY "User Registration Successful"
END
```

### 3.6.2 Student Management Module

This module manages student profiles including personal and academic details such as name, USN, department, CGPA, and email. It allows students to register and update their profile, which is stored in the **STUDENTS** table and later used for eligibility checks.

#### Pseudocode: Student Registration

```
BEGIN
    INPUT name, usn, department, cgpa, email
    VALIDATE student input
    INSERT student details into STUDENTS table
    DISPLAY "Student Profile Created"
```

END

### 3.6.3 Company and Job Management Module

The Company and Job Management Module enables authorized users to add company details and post job openings. Company information is stored in the COMPANIES table, and each job linked to a company is stored in the JOBS table with role, package, and eligibility criteria.

#### Pseudocode: Add Job

```
BEGIN
    INPUT company_id, role, package_lpa, eligibility_cgpa
    VERIFY company_id exists in COMPANIES
    INSERT job details into JOBS table
    DISPLAY "Job Posted Successfully"
END
```

### 3.6.4 Application Management Module

This module allows students to apply for available jobs and tracks the status of each application. It calculates eligibility based on CGPA and prevents duplicate applications for the same job.

#### Pseudocode: Apply for Job

```
BEGIN
    INPUT student_id, job_id
    VERIFY student_id and job_id exist
    CHECK if student meets eligibility_cgpa for job
    IF eligible THEN
        CHECK if application already exists
        IF not exists THEN
            INSERT record into APPLICATIONS table with status = "Applied" and curr...
            DISPLAY "Application Submitted"
        ELSE
            DISPLAY "Already Applied for this Job"
        ENDIF
    ENDIF
```

```

    ELSE
        DISPLAY "Not Eligible for this Job"
    ENDIF
END

```

### 3.6.5 Interview Management Module

The Interview Management Module manages scheduling and recording of interview outcomes for each application. It links each interview to a specific application and updates the application status after the result is recorded.

#### Pseudocode: Schedule Interview

```

BEGIN
    INPUT application_id, interview_date
    VERIFY application_id exists and status = "Shortlisted"
    INSERT interview_id, application_id, interview_date into INTERVIEWS table with
    UPDATE APPLICATIONS table SET status = "Interview Scheduled" WHERE application_id =
    DISPLAY "Interview Scheduled Successfully"
END

```

#### Pseudocode: Update Interview Result

```

BEGIN
    INPUT interview_id, result
    VERIFY interview_id exists
    UPDATE INTERVIEWS table SET result = given result WHERE interview_id = given interview_id
    IF result = "Selected" THEN
        UPDATE APPLICATIONS table SET status = "Selected"
    ELSE
        UPDATE APPLICATIONS table SET status = "Rejected"
    ENDIF
    DISPLAY "Interview Result Updated"
END

```

## 3.7 Tables Used

Sl. No.	Work	Duration (in Weeks)
1	Requirement gathering and problem definition	1
2	Designing ER diagram and database schema	1
3	Creating database tables for students, companies, jobs, applications, interviews, users	1
4	Developing backend modules (authentication, CRUD APIs)	2
5	Developing frontend interfaces for students, companies and admin	2
6	Integrating application and interview workflows	1
7	Testing, debugging and documentation	1

Table 3.1: Work Flow of Campus Placement Management System Development

TC#	Description	Expected Result	Actual Result	Status
TC-1	Student registration	New student record stored and visible in STUDENTS table/profile page	Student details saved and profile displayed correctly	Pass
TC-2	Company posts a job	Job created and listed under company with correct eligibility and package	Job displayed in jobs list for eligible students	Pass
TC-3	Student applies for a job	Application stored with status “Applied” and no duplicate entry created	Application inserted once in APPLICATIONS table with correct status	Pass
TC-4	Schedule interview for shortlisted application	Interview created and status updated to “Interview Scheduled”	Interview record inserted, application status updated correctly	Pass
TC-5	Update interview result	Application status changed to “Selected” or “Rejected” as per result	Status updated correctly in APPLICATIONS and visible to student	Pass

Table 3.2: Test Cases for Campus Placement Management System

# Chapter 4

## Results and Discussion

This chapter shows the output of the Campus Placement Management System, along with snapshots of the user interface and explanations. The system offers an interactive, user-friendly way to manage student profiles, job postings, applications, and interview schedules in one place. Each screen demonstrates how the corresponding module works and illustrates how data flows through the underlying database tables.

### Student Dashboard

The homepage includes links to major modules like viewing available jobs, managing applications, and checking interview schedules. Students can quickly see their basic details and current application status. This transparency helps them track their placement activities effectively.

ID	Student	Job (Role - Company)	Application Date	Status
1	Gita Hegde (4SF23AI066)	Machine Learning Engineer - BlueSys	2025-05-21	Applied
2	Pooja Sharma (4SF23AI027)	QA Engineer - ApexTechnologies	2025-03-01	Applied
3	Kunal Fernandes (4SF23AI004)	Platform Engineer - ApexSolutions	2025-11-09	Shortlisted
4	Sneha Nair (4SF23AI077)	Platform Engineer - GlobalEnterpris	2025-08-18	Applied
5	Bharath Varma (4SF23AI072)	Technical Support - SoftTechnologie	2025-05-19	Rejected
6	Pavan Das (4SF23AI024)	Fullstack Developer - AIConsulting	2025-02-26	Applied
7	Lokesh Rao (4SF23AI003)	Product Engineer - BlueStudios	2025-08-11	Shortlisted
8	Pooja Sharma (4SF23AI027)	Data Analyst - BlueSystems	2025-06-17	Applied
9	Varsha Das (4SF23AI005)	Business Analyst - NextConsulting	2025-03-02	Applied
10	Janani Bhat (4SF23AI058)	Research Intern - InnoSolutions	2025-09-23	Applied
11	Kumar Fernandes (4SF23AI051)	Product Engineer - BlueStudios	2025-08-16	Rejected
12	Ritika Prasad (4SF23AI093)	Backend Developer - GreenStudios	2025-11-14	Applied
13	Saurabh Chauhan (4SF23AI014)	QA Engineer - ApexTechnologies	2025-03-15	Withdrawn
14	Bindu Naik (4SF23AI031)	Platform Engineer - TechWorks	2025-06-09	Applied
15	Bharath Menon (4SF23AI023)	Research Intern - SigmaConsulting	2025-06-17	Shortlisted
16	Mira Fernandes (4SF23AI088)	Backend Developer - SoftConsulting	2025-09-18	Rejected
17	Ritika Prasad (4SF23AI093)	Backend Developer - SoftConsulting	2025-04-28	Shortlisted
18	Arjun Menon (4SF23AI065)	System Administrator - InfoWorks	2025-08-09	Rejected
19	Pavan Das (4SF23AI024)	Technical Support - SoftTechnologie	2025-07-18	Applied
20	Nikita Menon (4SF23AI008)	Data Scientist - SysDynamics	2025-07-12	Shortlisted
21	Nivedita Kumar (4SF23AI033)	Fullstack Developer - CoreStudios	2025-03-08	Rejected
22	Gita Hegde (4SF23AI066)	Data Scientist - SysDynamics	2025-01-19	Applied
23	Shanaya Chauhan (4SF23AI055)	Backend Developer - BlueWorks	2025-01-01	Applied
24	Sowmya Iyer (4SF23AI067)	Technical Support - CloudLabs	2025-10-12	Rejected
25	Shreyas Iyer (4SF23AI091)	Platform Engineer - NetStudios	2025-09-21	Applied

Figure 4.1: Student Dashboard

## Job Listings Page

This page lists all active job opportunities pulled from the JOBS and COMPANIES tables. Users can view the company name, role, package, location, and required CGPA before applying. This process ensures that only relevant and eligible openings are considered.

company_id	name	location	industry
1	InfoWorks 1	Goa	Research
2	VertexLabs 2	Kochi	Healthcare
3	GlobalEnterprises 3	Chennai	Fintech
4	NetTechnologies 4	Chennai	IT
5	SoftTechnologies 5	Noida	Fintech
6	VertexWorks 6	Goa	Fintech
7	AISystems 7	Mangalore	IT
8	CloudStudios 8	Hyderabad	IT
9	SysDynamics 9	Mumbai	Consulting
10	VertexLabs 10	Delhi	Healthcare
11	AIEnterprises 11	Chennai	EdTech
12	DataDynamics 12	Bangalore	Automotive
13	BlueServices 13	Hyderabad	Consulting
14	DataDynamics 14	Goa	Consulting
15	InfoSystems 15	Kochi	Automotive
16	DataLabs 16	Pune	IT Services
17	SigmaEnterprises 17	Pune	Research
18	DataDynamics 18	Mumbai	IT
19	SigmaServices 19	Kochi	IT
20	AIConsulting 20	Goa	Consulting
21	PrimeWorks 21	Mumbai	Research
22	VertexConsulting 22	Mumbai	Fintech
23	ApexLabs 23	Pune	Fintech
24	InfoSystems 24	Pune	IT Services
25	TechWorks 25	Hyderabad	E-commerce
26	InnoTechnologies 26	Bangalore	Consulting
27	BlueSystems 27	Pune	Research

Figure 4.2: Job Listings Page

## Apply for Job / Applications Page

The applications interface allows a logged-in student to apply for a selected job in one step. After submission, a new record is saved in the APPLICATIONS table with the status

“Applied”, and the page shows all applications with their current status, such as Applied, Shortlisted, Interview Scheduled, Selected, or Rejected.

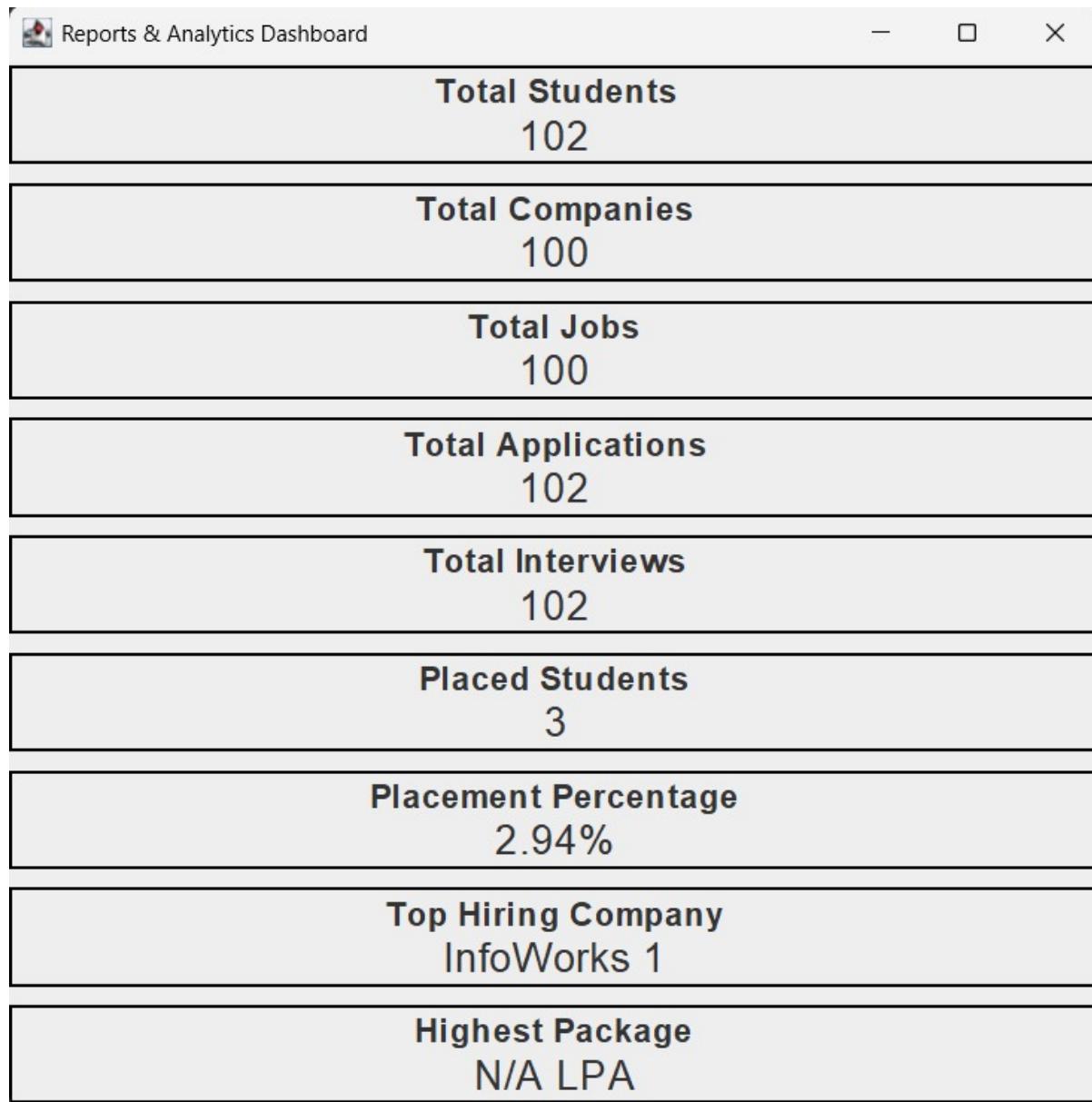


Figure 4.3: Applications Page

## Admin / TPO Dashboard

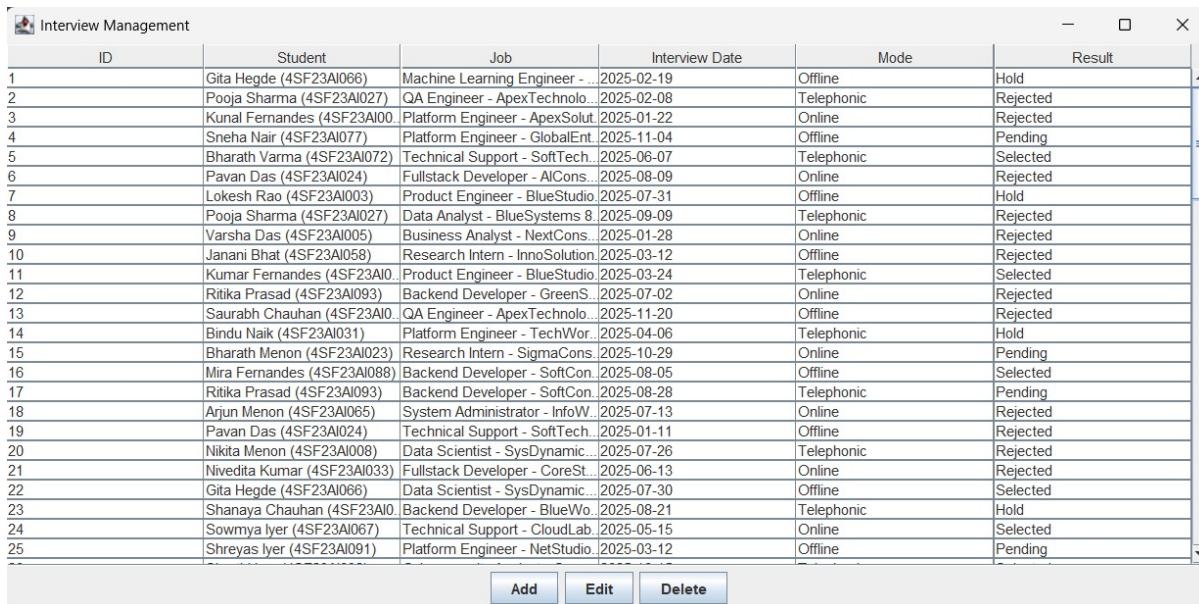
The admin or TPO module shows combined information about students, companies, jobs, and applications. From this dashboard, the administrator can post new jobs, shortlist candidates, schedule interviews, and update results. This setup demonstrates smooth coordination between the **USERS**, **STUDENTS**, **COMPANIES**, **JOBS**, and **APPLICATIONS** tables.

	application_id	student_id	job_id	application_date	status
▶	1	66	27	2025-05-21	Applied
	2	27	67	2025-03-01	Applied
	3	4	94	2025-11-09	Shortlisted
	4	77	3	2025-08-18	Applied
	5	72	5	2025-05-19	Rejected
	6	24	20	2025-02-26	Applied
	7	3	30	2025-08-11	Shortlisted
	8	27	88	2025-06-17	Applied
	9	5	32	2025-03-02	Applied
	10	58	62	2025-09-23	Applied
	11	51	30	2025-08-16	Rejected
	12	93	95	2025-11-14	Applied
	13	14	67	2025-03-15	Withdrawn
	14	31	25	2025-06-09	Applied
	15	23	48	2025-06-17	Shortlisted
	16	88	46	2025-09-18	Rejected
	17	93	46	2025-04-28	Shortlisted
	18	65	75	2025-08-09	Rejected
	19	24	5	2025-07-18	Applied
	20	8	44	2025-07-12	Shortlisted
	21	33	54	2025-03-08	Rejected
	22	66	44	2025-01-19	Applied
	23	55	53	2025-01-01	Applied
	24	67	51	2025-10-12	Rejected
	25	91	38	2025-09-21	Applied
	26	32	29	2025-04-22	Shortlisted
	27	83	30	2025-07-28	Shortlisted

Figure 4.4: TPO Dashboard

## Interview Management Screens

Screens for scheduling and updating interviews allow the TPO to set interview dates for shortlisted applications and record outcomes. Once the result is updated, the application status changes, and students can immediately see the decision. This confirms that the INTERVIEWS module is well integrated with the rest of the system.



ID	Student	Job	Interview Date	Mode	Result
1	Gita Hegde (4SF23AI066)	Machine Learning Engineer - ...	2025-02-19	Offline	Hold
2	Pooja Sharma (4SF23AI027)	QA Engineer - ApexTechnolo...	2025-02-08	Telephonic	Rejected
3	Kunal Fernandes (4SF23AI00...	Platform Engineer - ApexSolut...	2025-01-22	Online	Rejected
4	Sneha Naik (4SF23AI072)	Platform Engineer - GlobalEnt...	2025-11-04	Offline	Pending
5	Bharath Varma (4SF23AI072)	Technical Support - SoftTech...	2025-06-07	Telephonic	Selected
6	Pavan Das (4SF23AI024)	Fullstack Developer - AICons...	2025-08-09	Online	Rejected
7	Lokesh Rao (4SF23AI003)	Product Engineer - BlueStudio...	2025-07-31	Offline	Hold
8	Pooja Sharma (4SF23AI027)	Data Analyst - BlueSystems 8...	2025-09-09	Telephonic	Rejected
9	Varsha Das (4SF23AI005)	Business Analyst - NextCons...	2025-01-28	Online	Rejected
10	Janani Bhat (4SF23AI058)	Research Intern - InnoSolution...	2025-03-12	Offline	Rejected
11	Kumar Fernandes (4SF23AI0...	Product Engineer - BlueStudio...	2025-03-24	Telephonic	Selected
12	Ritika Prasad (4SF23AI093)	Backend Developer - GreenS...	2025-07-02	Online	Rejected
13	Saurabh Chauhan (4SF23AI0...	QA Engineer - ApexTechnolo...	2025-11-20	Offline	Rejected
14	Bindu Naik (4SF23AI031)	Platform Engineer - TechWor...	2025-04-06	Telephonic	Hold
15	Bharath Menon (4SF23AI023)	Research Intern - SigmaCons...	2025-10-29	Online	Pending
16	Mira Fernandes (4SF23AI088)	Backend Developer - SoftCon...	2025-08-05	Offline	Selected
17	Ritika Prasad (4SF23AI093)	Backend Developer - SoftCon...	2025-08-28	Telephonic	Pending
18	Arjun Menon (4SF23AI065)	System Administrator - InfoW...	2025-07-13	Online	Rejected
19	Pavan Das (4SF23AI024)	Technical Support - SoftTech...	2025-01-11	Offline	Rejected
20	Nikita Menon (4SF23AI008)	Data Scientist - SysDynamic...	2025-07-26	Telephonic	Rejected
21	Nivedita Kumar (4SF23AI033)	Fullstack Developer - CoreSt...	2025-06-13	Online	Rejected
22	Gita Hegde (4SF23AI066)	Data Scientist - SysDynamic...	2025-07-30	Offline	Selected
23	Shanaya Chauhan (4SF23AI0...	Backend Developer - BlueWo...	2025-08-21	Telephonic	Hold
24	Sowmya Iyer (4SF23AI067)	Technical Support - CloudLab...	2025-05-15	Online	Selected
25	Shreyas Iyer (4SF23AI091)	Platform Engineer - NetStudio...	2025-03-12	Offline	Pending

Figure 4.5: Interview Management Screens

## Discussion

The results indicate that the Campus Placement Management System works as intended and supports the entire placement workflow from job creation to final selection. All modules integrate well with the database, ensuring accurate storage and retrieval of data related to students, companies, jobs, applications, and interviews. The job listing and application features ensure that only eligible students can apply, while the interview management module reliably tracks each candidate's progress. The admin dashboard gives clear visibility into ongoing drives, helping placement staff monitor applications and outcomes effectively. Overall, the project meets its goals by providing a reliable, efficient, and user-friendly solution for managing campus placement activities in an educational institution.

# **Chapter 5**

## **Conclusion and Future Work**

### **Conclusion**

The Campus Placement Management System developed in this project digitizes and automates key activities such as student registration, profile management, job posting, eligibility filtering, and application tracking, which reduces manual effort and errors for both students and the Training and Placement Cell. By providing a centralized platform for storing and accessing placement-related data, the system improves transparency in the recruitment workflow and offers a more efficient and reliable alternative to traditional paper-based or spreadsheet-based processes.

### **Future Work**

The current implementation can be enhanced by integrating analytics dashboards that provide insights into placement trends, company requirements, and student performance over multiple recruitment cycles. Future work may also include role-based mobile applications, real-time notification services, AI-driven resume screening and job recommendation, as well as stronger security and audit mechanisms to support scalable multi-institution usage.

# References

- [1] S. R. Pawar and A. Kulkarni, “Design and Development of a College Placement Management System Using Relational Database Concepts,” in *Proc. IEEE Int. Conf. Advanced Computing and Data Engineering (ACDE)*, 2024, pp. 112–118.
- [2] R. Sharma and D. Gupta, “A Web-Based Placement Portal with Centralized Database for Student Recruitment,” in *Proc. IEEE Int. Conf. Computing, Communication, and Networking Technologies (ICCCNT)*, 2024, pp. 355–361.
- [3] P. N. Rao and K. S. Prasad, “Database Design and Implementation of a Student Recruitment Management System,” in *Proc. IEEE Conf. Data Management, Analytics and Innovation (ICDMAI)*, 2025, pp. 89–94.
- [4] T. Das and S. Mitra, “Normalization and Schema Optimization for College Placement Databases,” in *Proc. IEEE Int. Conf. Smart Systems and Inventive Technology (ICSSIT)*, 2024, pp. 701–706.
- [5] M. Thomas and R. K. Singh, “A Relational Database Approach for Managing Campus Placement Processes,” in *Proc. IEEE Int. Conf. Information Systems and Computer Networks (ISCON)*, 2025, pp. 214–220.
- [6] V. L. Patil and K. M. Joshi, “Design of a Secure SQL-Based Student–Company Interaction System for Campus Recruitments,” in *Proc. IEEE Int. Conf. Electronics, Computing and Communication Technologies (CONECCT)*, 2024, pp. 341–347.
- [7] S. Iyer and P. Deshmukh, “ER-Modeling and Backend Development of a College Placement Automation System,” in *Proc. IEEE Int. Conf. Computing Methodologies and Communication (ICCMC)*, 2024, pp. 504–510.
- [8] A. Banerjee and L. S. Rao, “Database-Centric Architecture for Managing Student Profiles and Job Applications,” in *Proc. IEEE Region 10 Symposium (TENSYMP)*, 2025, pp. 188–193.

- [9] N. Krishnan and J. Abraham, “SQL Query Optimization in Multi-User College Placement Portals,” in *Proc. IEEE Int. Conf. Computational Intelligence and Data Science (ICCIDIS)*, 2024, pp. 132–138.
- [10] D. Mehta and S. Paul, “A DBMS-Based Placement Cell Management Application Using PHP and MySQL,” in *Proc. IEEE Int. Conf. Advances in Engineering, Applied Science and Materials (AEASM)*, 2025, pp. 77–83.