

**ATMIYA UNIVERSITY**

**RAJKOT**



A

Report On

**College Placement Record Management System**

Under subject of

**MINI PROJECT**

B. TECH, Semester – VII

**(Computer Engineering)**

Submitted by:

Tanvi Sanjaybhai Khokhariya (Enrollment No: 220002041)

**Prof. Devangi Rasikbhai Paneri**

(Faculty Guide)

**Prof. Tosal M. Bhalodia**

(Head of the Department)

Academic Year

**(2025-26)**

## **CANDIDATE'S DECLARATION**

We hereby declare that the work presented in this project entitled “**College Placement Record Management System**” submitted towards completion of project in **7<sup>th</sup> Semester** of B. Tech. (Computer Engineering) is an authentic record of our original work carried out under the guidance of “**Devangi Rasikbhai Paneri**”.

We have not submitted the matter embodied in this project for the award of any other degree.

Semester: 7<sup>th</sup>

Place: Rajkot

**Signature:**

Tanvi Khokhariya (220002041)

---

**ATMIYA UNIVERSITY**  
**RAJKOT**



**CERTIFICATE**

**Date:** 13-09-2025

This is to certify that the “**College Placement Record Management System**” has been carried out by **Tanvi Khokhariya** under my guidance in fulfillment of the subject Mini Project in COMPUTER ENGINEERING (7<sup>th</sup> Semester) of Atmiya University, Rajkot during the academic year 2025.

Prof. Devangi R. Paneri

Prof. Tosal M. Bhalodia

**(Project Guide)**

**(Head of the Department)**



## **ACKNOWLEDGEMENT**

We have taken many efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to **Prof. Devangi Rasikbhai Paneri** for their guidance and constant supervision as well as for providing necessary information regarding the Project titled “**College Placement Record Management System**”. We would like to express our gratitude towards staff members of the Computer Engineering Department, Atmiya University for their kind co-operation and encouragement which helped us in completion of this project.

We even thank and appreciate our colleague in developing the project and people who have willingly helped us out with their abilities.

Tanvi Khokhariya (220002041)

---

## **ABSTRACT**

The College Placement Record Management System is designed to automate and streamline the process of tracking student placements. It replaces traditional manual record-keeping methods, reducing errors and saving time. The system allows administrators to add, view, filter, and export placement records efficiently. It maintains comprehensive data on students, companies, and job positions, ensuring accurate and up-to-date information. Users can easily access placement details, generating reports as needed for analysis. The system enhances data integrity, improves decision-making, and supports better planning for placement activities. It is user-friendly, scalable, and adaptable to future requirements. Overall, it provides a reliable platform for managing all placement-related information in a systematic manner.

---

# INDEX

Sr. No.	TITLES		Page No.
	<b>Acknowledgement</b>		-
	<b>Abstract</b>		-
	<b>List of Figures</b>		1
	<b>List of Tables</b>		2
<b>1.</b>	<b>Introduction</b>		3
	1.1	Purpose	3
	1.2	Scope	3
	1.3	Technology and tool	3
<b>2.</b>	<b>Project Management</b>		4
	2.1	Project Planning	4
	2.2	Project Scheduling	4
	2.3	Risk Management	4
	2.3.1	Risk Identification	4
	2.3.2	Risk Analysis	4
<b>3.</b>	<b>System Requirements Study</b>		5
	3.1	Hardware and Software Requirements	5
	3.1.1	Server-Side hardware requirement	5
	3.1.2	Software requirement	5
	3.1.3	Client-Side requirement	5
	3.2	Constraints	6
	3.2.1	Hardware Limitation	6
	3.2.2	Reliability requirements	6
	3.2.3	Safety and Security Consideration	6
<b>4.</b>	<b>System Analysis</b>		7
	4.1	Study of Current System	7
	4.2	Problem and Weaknesses of Current System	7
	4.3	Requirements of New System	8
	4.3.1	User Requirements	8
	4.3.2	System Requirements	8

	4.4	Feasibility Study	8
	4.5	Feature Of New System	8
<b>5</b>	<b>System Design</b>		9
	5.1	Input /output interface	9
	5.2	Interface Design	10
		5.2.1 Class Diagram	10
		5.2.2 Use Case Diagram	10
		5.2.3 Activity Diagram	11
		5.2.4 Data Flow Diagram	11
		5.2.5 State Diagram	12
		5.2.6 E-R Diagram	12
		5.2.7 Sequence Diagram	13
<b>6</b>	<b>Code Implementation</b>		14
	6.1	Implementation Environment	14
	6.2	Program/Module Specification	14
	6.3	Coding Standards	14
<b>7</b>	<b>Testing</b>		15
	7.1	Testing Strategy	15
	7.2	Testing Method	15
		7.2.1 Unit Testing	15
		7.2.2 Integration Testing	15
		7.2.3 Validation Testing	15
	7.3	Test Cases	16
		7.3.1 Test Suite	16
<b>8</b>	<b>Limitations and Future Enhancement</b>		17
	8.1	Limitations	17
	8.2	Future Enhancement	17
<b>9</b>	<b>Conclusion</b>		18
<b>10</b>	<b>References</b>		19

## **LIST OF FIGURES**

<b>Fig. No.</b>	<b>Figure Title</b>	<b>Page No.</b>
<b>1.1</b>	Class Diagram	10
<b>1.2</b>	Use Case Diagram	10
<b>1.3</b>	Activity Diagram	11
<b>1.4</b>	Data Flow Diagram	11
<b>1.5</b>	State Diagram	12
<b>1.6</b>	E-R Diagram	12
<b>1.7</b>	Sequence Diagram	13



## **LIST OF TABLES**

<b>Table No.</b>	<b>Table Title</b>	<b>Page No.</b>
<b>3.1.1</b>	Server-Side Hardware Requirements	5
<b>3.1.2</b>	Software Requirements	5
<b>3.1.3</b>	Client-Side Requirements	5

# **CHAPTER – 1**

## **INTRODUCTION**

### **1.1. Purpose**

- The purpose of the **College Placement Record Management System** is to provide an efficient and user-friendly online platform for managing student placement records in a college.
- It allows administrators to add companies, job positions, packages, and student details, and facilitates tracking and reporting placement records with various filters. The system aims to reduce paperwork, streamline data management, and provide quick access to placement information.

### **1.2 Scope**

- This platform is designed for educational institutions to maintain placement data for students. It covers functionalities such as adding student information, managing companies and job positions, recording placements, filtering data, and exporting reports in CSV and Excel formats.
- The system helps streamline placement-related tasks and improve record-keeping and reporting.

### **1.3 Technology and Tools**

- **Frontend:** HTML, CSS, JavaScript, Bootstrap
- **Backend:** PHP / Python Flask / Node.js
- **Database:** MySQL / PostgreSQL
- **Tools:** Visual Studio Code
- **Libraries:** DataTables.js for enhanced data filtering and reporting
- **Export:** PHP Excel or Pandas for Excel generation

# **CHAPTER – 2**

## **PROJECT MANAGEMENT**

### **2.1 Project Planning**

- A structured plan was created to ensure the successful development of the system. Initially, requirements were gathered by understanding the needs of users and stakeholders. The database structure was designed to efficiently store and retrieve data. The user interface was planned to be simple and user-friendly. Tasks were scheduled with clear timelines, and responsibilities were assigned to team members. Potential risks were identified, and strategies were prepared to minimize their impact.

### **2.2 Project Scheduling**

- The project tasks were organized into different phases to ensure smooth progress. It started with requirement analysis to understand the system's needs. Next, the design phase focused on creating the structure and layout. Coding was then done to build the system's features. After development, testing was carried out to find and fix any issues. Finally, the system was deployed for use. Gantt charts were used throughout to monitor progress and keep track of deadlines.

### **2.3 Risk Management**

- Risk management involves identifying potential issues like data loss or security threats and planning strategies to prevent or minimize their impact. It ensures the system runs smoothly and remains secure.

-

#### **2.3.1 Risk Identification**

- The project may face risks such as data loss or corruption, security vulnerabilities, and user authentication failures. There could also be technical limitations in server hosting that affect performance. Additionally, incorrect data input by users may lead to errors in the system. These risks need to be carefully managed to ensure smooth operation.

#### **2.3.2 Risk Analysis**

- Each risk was analyzed for its probability and impact, with mitigation strategies such as regular backups, secure password encryption, validation checks, and server monitoring.

## **CHAPTER – 3**

### **SYSTEM REQUIREMENTS STUDY**

#### **3.1 Hardware and Software Requirement**

- This shows minimum requirements to carry on to run this system efficiently.

##### **3.1.1 Hardware Requirements**

###### **Server-side Hardware Requirement:**

<b>Devices</b>	<b>Requirement*</b>
Processor	Intel Core i3 2.4 GHz or more
RAM	4 GB or more
Hard Disk	100 GB or more

Table 3.1.1 Server-side Hardware Requirement

##### **3.1.2 Software Requirements:**

<b>For Which</b>	<b>Software*</b>
Operating System	Windows 10/11, Linux
Front End	HTML, CSS, Bootstrap
Back End	PHP with XAMPP / Python with Flask
Coding Language	Visual Studio Code

Table 3.1.2 Software Requirement

##### **3.1.3 Client-side Requirements:**

<b>For Which</b>	<b>Requirement*</b>
Android	Version 5.0 or above
Desktop	Any version of Chrome, Firefox

Table 3.1.3 Client-side Requirement

\* Minimum Requirements and Software for uses

## **3.2 Constraints**

### **3.2.1 Hardware Limitations:**

- The server's hardware resources such as processor speed, memory, and storage may be limited. During peak usage times, this limitation can affect the system's ability to handle large amounts of data. If too many users access the system at once, performance may slow down. The system might experience delays in processing or storing data. Careful planning is needed to ensure efficient data handling. Upgrading hardware can help reduce such limitations.

### **3.2.2 Reliability Requirements:**

- The system must be designed to handle issues like data duplication or corruption. It should have mechanisms to recover from errors that occur during operation. Maintaining session persistence is important so users don't lose their work if interrupted. Backup systems should be in place to restore data when needed. Consistent data management ensures smooth functioning. These features help in providing a reliable user experience.

### **3.2.3 Safety and Security Consideration:**

- The system must ensure that data is kept secure from unauthorized access. Encryption techniques should be used to protect sensitive information. Access control should limit who can view or modify data. Validation checks are necessary to prevent incorrect or malicious data entry. Regular monitoring can help detect and prevent misuse. These measures are essential for protecting the system and its users from security threats.

# **CHAPTER – 4**

## **SYSTEM ANALYSIS**

### **4.1 Study Current System**

- At present, the placement records in many colleges are maintained manually. The data is often stored in spreadsheets or on paper-based forms, which makes it difficult to keep information organized. Since multiple entries are added by different people, errors such as duplication or incorrect information are common.
- Updating records requires a lot of time and effort, especially when large amounts of data need to be handled. Searching for specific placement information, like student names or companies, is also a slow and inefficient process. Additionally, manual records lack proper data validation, which increases the chances of mistakes being overlooked.
- There is no central system to track changes, which makes it hard to maintain accountability. Data integrity is compromised because backups are either not available or poorly managed. Over time, this leads to incomplete or outdated records.
- The system does not support reporting or analytics, which are important for decision-making. Furthermore, storing records on paper or unsecured spreadsheets makes them vulnerable to loss or theft.
- These limitations create challenges in managing placement information effectively, leading to delays in reporting and increased administrative work. A more automated and secure system is needed to overcome these issues and streamline the placement process.

### **4.2 Problem and weakness of current system**

- Data redundancy and inconsistency
- No centralized record management
- Inefficient reporting and filtering
- High risk of data loss
- Limited access and update tracking

## **4.3 Requirements of New System**

### **4.3.1 User Requirements:**

- Easy-to-use interface
- Secure login and user authentication
- Ability to add companies, job positions, and student records
- Filter and export placement reports

### **4.3.2 System Requirements:**

- Robust database design
- Responsive web interface
- Data validation and error handling
- Export functionality in Excel and CSV formats

## **4.4 Feasibility Study**

- The project is feasible because modern web technologies and database systems are widely available. Development tools like PHP, Python, and MySQL make building the system easier. It doesn't require expensive hardware or software, making it cost-effective for colleges. The system can be scaled to handle more students and companies as needed. With proper planning, it can be easily maintained and upgraded in the future. This makes it a practical solution for managing placement records efficiently.

## **4.5 Feature of New System**

- Centralized placement data management
- Efficient data retrieval using filters
- Automated reporting and downloads
- User authentication and security features

## CHAPTER – 5

### System Design

#### 5.1 Input/output interface

- **Input:** The system allows users to input data such as student details, company information, job positions, packages, and placement records through web forms.

**College Placement Record Management**

Company Name:  
TCS

Position:  
Software Developer

Package (LPA):  
12

[Add Company](#)

**College Placement Record Management**

Student Name:  
Tanvi Khokhariya

Branch:  
Computer Engineering

Year:  
2025

[Add Student](#)

**College Placement Record Management**

Select Student:  
Tanvi Khokhariya (Computer Engineering - Year 2024) ▼

Select Company:  
TCS (Software Developer - 12.0 LPA) ▼

[Place Student](#)

- **Output:** The system provides output in the form of placement records, filtered views based on different criteria, and downloadable reports in Excel or CSV format.

**College Placement Record Management**

**Placement Records**

Student Name Branch Year Company All ▼

Sort By ▼ [Filter](#)

[Export to CSV](#) [Export to Excel](#)

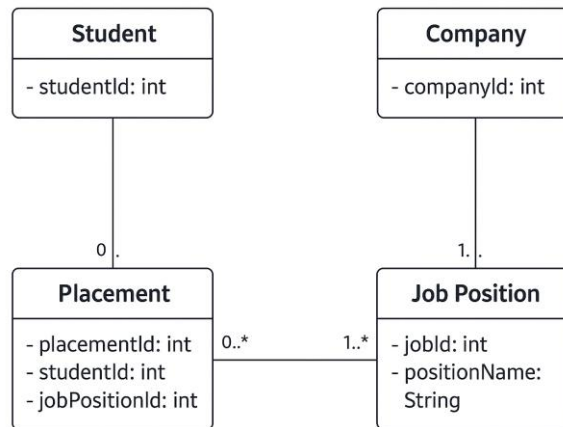
**Placed Students**

Student	Branch	Year	Company	Position	Package (LPA)	Resume
Tanvi Khokhariya	Computer Engineering	2024	TCS	Software Developer	12.0	N/A
Riddhi Khokhariya	Computer Engineering	2025	Infosys	Python Developer	10.0	N/A

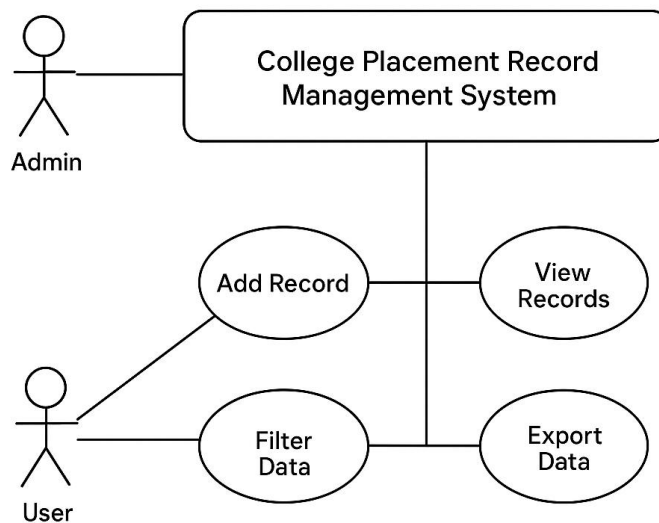


## 5.2 Interface Design

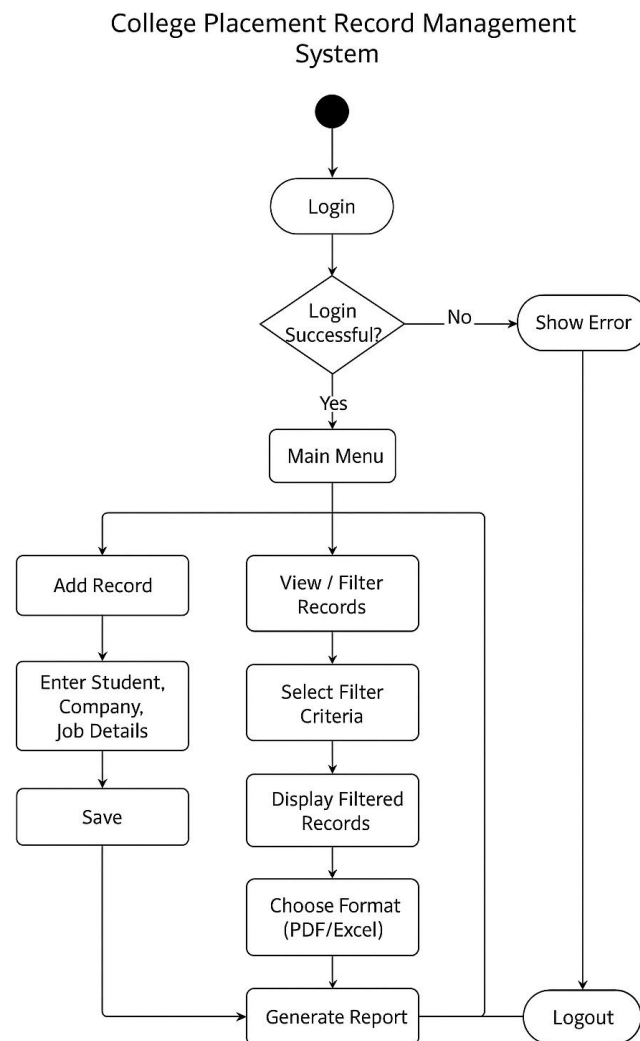
### 5.2.1 Class Diagram:



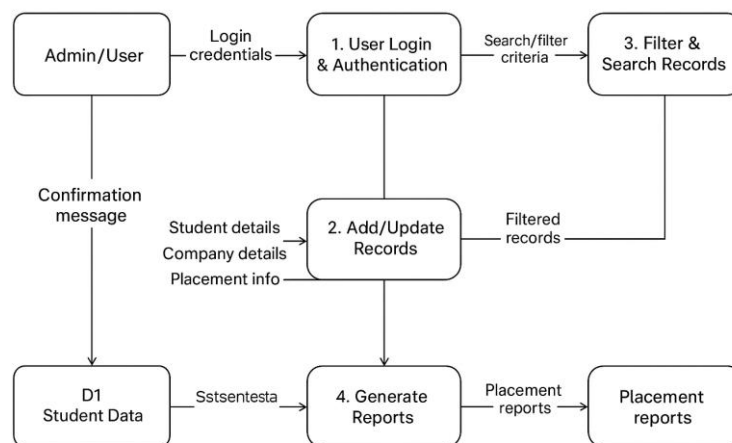
### 5.2.2 Use Case Diagram:



### 5.2.3 Activity Diagram:



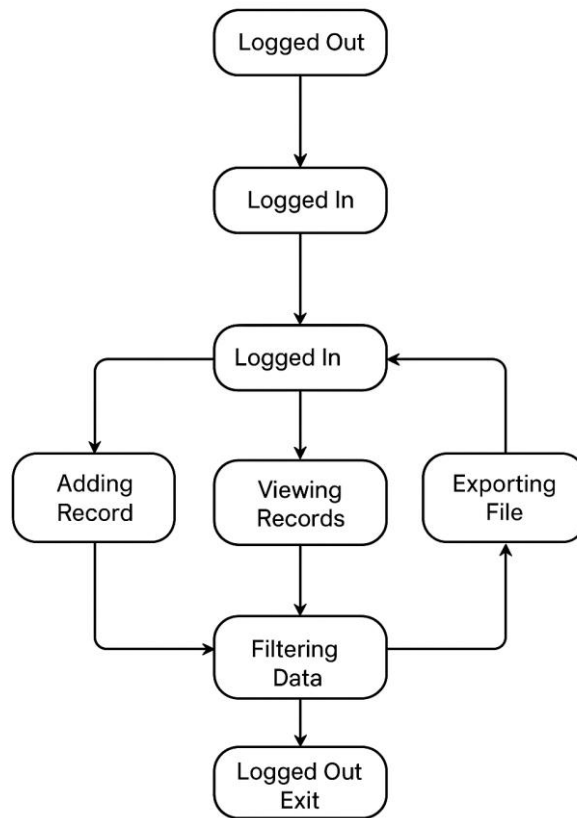
### 5.2.4 Data Flow Diagram:



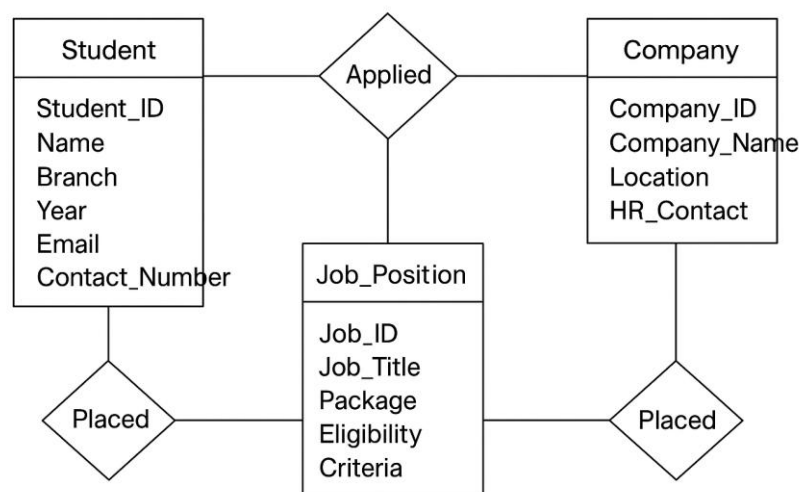
College Placenment Record Management System

### 5.2.5 State Diagram:

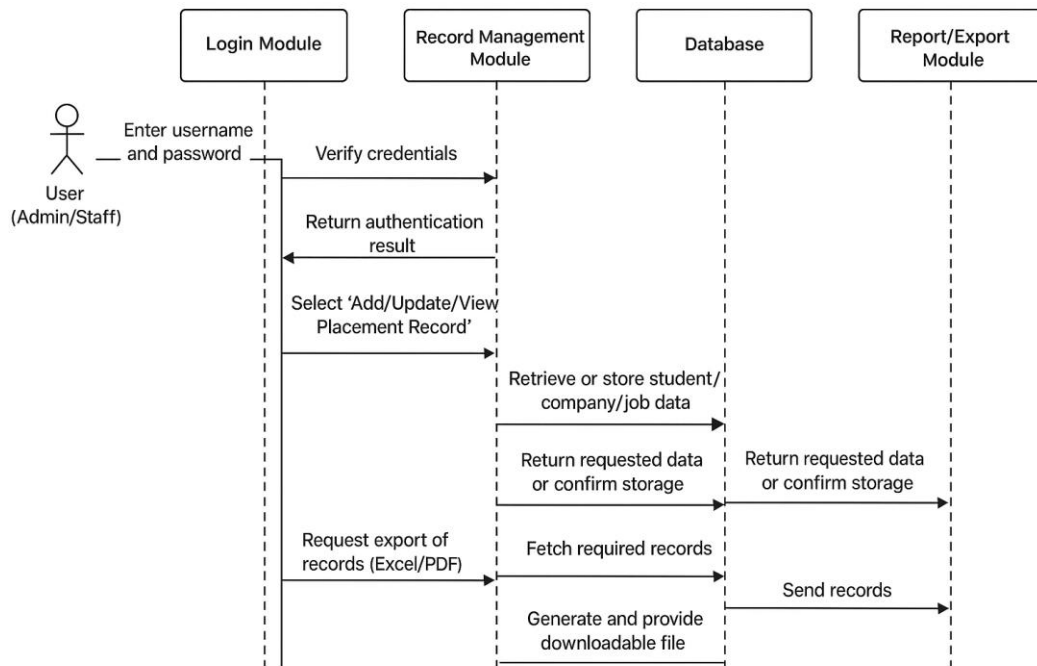
State Diagram



### 5.2.6 E-R Diagram:



## 5.2.7 Sequence Diagram:



# **CHAPTER – 6**

## **Code Implementation**

### **6.1 Implementation Environment**

- The implementation environment consists of a XAMPP server with Apache and MySQL for hosting and database management. Alternatively, Docker containers can be used for development and testing purposes. This setup provides a stable and flexible environment for deploying and running the application. It supports efficient testing, debugging, and maintenance of the system. Overall, it ensures smooth development and reliable performance.

### **6.2 Program/Module Specification**

Modules include:

- User Authentication
- Student Management
- Company Management
- Job Position Management
- Placement Record Module
- Report Filtering and Export

### **6.3 Coding Standards**

- Consistent naming conventions
- Commenting for maintainability
- Input validation and sanitization
- Secure password storage using hashing
- Error handling and logging

# **CHAPTER – 7**

## **Testing**

### **7.1 Testing Strategy**

- The testing strategy involves multiple levels to ensure the system's reliability and performance. Unit testing will verify individual components for correctness. Integration testing will check the interaction between different modules. Validation testing will ensure the system meets all functional and user requirements. This comprehensive approach helps identify and fix defects early. Overall, it ensures a robust and error-free application.

### **7.2 Testing Method**

#### **7.2.1 Unit Testing:**

- Unit testing focuses on verifying each module of the system individually. Modules such as login, adding student details, or filtering records are tested separately. This ensures that each component functions correctly on its own. Any errors or bugs are identified and fixed at this stage. Unit testing helps maintain code quality and prevents issues from affecting other parts of the system. It forms the foundation for reliable integration testing later.

#### **7.2.2 Integration Testing:**

- Integration testing focuses on verifying the interaction between different modules of the system. For example, after adding a student, the system is checked to ensure the details correctly appear in the placement records. This testing ensures that modules work together as intended. It helps identify issues that may arise when combining individual components. Integration testing improves overall system reliability and data consistency. It ensures smooth end-to-end functionality of the application.

#### **7.2.3 Validation Testing:**

- Data validation testing ensures that all inputs meet the required rules and constraints. It checks for proper email formats, mandatory fields, and restrictions on numeric data. This prevents incorrect or incomplete data from being entered into the system.

### **7.3 Test Cases**

- Test Cases include detailed scenarios with specific inputs, expected outputs, and the actual results obtained. They help verify that the system behaves correctly under different conditions. Each test case is designed to identify errors or deviations from expected behavior.

#### **7.3.1 Test Suite:**

- Test Suite is a comprehensive collection of these test cases. It covers various aspects of the system, including functionality, performance, and security. This organized approach ensures thorough testing and helps maintain overall system quality.

## **CHAPTER – 8**

### **Limitations and Future Enhancement**

#### **8.1 Limitations**

- The system is dependent on internet availability, which may affect accessibility during outages. It relies on manual data input, making the process time-consuming and prone to errors. There is no automated data syncing across institutions, limiting real-time updates and coordination. These factors can reduce overall efficiency and scalability of the system.
- Dependent on internet availability
- Limited to manual input
- No automated data syncing across institutions

#### **8.2 Future Enhancement**

- Future enhancements for the system include integration with existing college ERP systems to streamline data management. Automated job alerts via email can keep students informed about new opportunities in real-time.
- An AI-powered job recommendation feature can suggest suitable jobs based on individual student profiles and preferences. Support for multiple institutions can expand the system's usability across different colleges.
- Additional improvements may include advanced analytics, mobile app support, and enhanced security features. These enhancements aim to make the system more efficient, intelligent, and user-friendly.
- Integration with college ERP systems
- Automated job alerts via email
- AI-powered job recommendation based on student profiles
- Multi-institution support



## **CHAPTER – 9**

### **Conclusion**

- In conclusion, the College Placement Record Management System offers a streamlined and efficient approach to managing student placement records. It eliminates the challenges and errors associated with manual record-keeping.
- The system ensures accurate and organized data storage, making it easier to access and update information. It supports comprehensive reporting, enabling better analysis of placement trends and student performance.
- The user-friendly interface enhances usability for both administrators and students. With features like secure data management and systematic tracking, the system improves overall operational efficiency.
- It provides a reliable and scalable platform for handling placement activities. Overall, it significantly enhances the management and monitoring of college placements, benefiting all stakeholders.

## **CHAPTER – 10**

### **Reference**

- **W3Schools** – Web Development Tutorials
- **PHP Manual** – PHP Functions and Security
- **MySQL Documentation** – Database Management
- **Bootstrap Framework** – UI Design
- **DataTables.js** – Interactive Tables and Filters
- **PHP Excel / Pandas Documentation** – File Exporting Techniques