



# **Industrial Internship Report**

**Tech Elecon Private Limited**

*Submitted by*

**MOKSHABEN GHANSHYAMSINH BODANA**

**12102040701121**

*In partial fulfillment for the award of the degree of*

**BACHELOR OF ENGINEERING**

*in*

**Computer Engineering**

**Madhuben and Bhanubhai Patel Institute of Technology**

**The Charutar Vidya Mandal (CVM) University,**

**Vallabh Vidyanagar - 388120**

**April, 2025**



**Madhuben & Bhanubhai Patel Institute of Technology**  
**Computer Engineering**

**CERTIFICATE**

This is to certify that **Mokshaben Ghanshyamsinh Bodana (12102040701121)** has submitted the Industrial Internship report based on internship undergone at **Tech Elecon Pvt. Ltd, Anand.** for a period of **16 weeks** from **1<sup>st</sup> January 2025** to **30<sup>th</sup> April 2025** in partial fulfillment for the degree of Bachelor of Engineering in **Computer Engineering, Madhuben and Bhanubhai Patel Institute of Technology** at The Charutar Vidya Mandal (CVM) University, Vallabh Vidyanagar during the academic year 2024 – 25.

---

Prof. Ankita Chauhan

Internal Guide(s)

---

Dr. Gopi Bhatt

Head of the Department

Date: /04/2025

## TO WHOM IT MAY CONCERN

This is to certify that **Mokshaben Ghanshyamsinh Bodana**, a student of **COMPUTER ENGINEERING** department of **MADHUBEN AND BHANUBHAI PATEL INSTITUTE OF TECHNOLOGY**, has successfully completed her internship in the field of **React.js** from **01/01/2025** to **30/04/2025** under the guidance of **Mr. Satyam Raval**, Deputy General Manager at **Tech Elecon Pvt. Ltd.**

Her internship activities include successful completion of the assigned project at the given period of time along with abiding by companies' rules and regulation.

During the period of her internship program with us, she had been exposed to different processes and was found diligent, hardworking, and inquisitive.

We wish her every success in her life and career.

For Tech Elecon Pvt. Ltd.



## DECLARATION

I, Mokshaben Ghanshyamsinh Bodana (12102040701121), hereby declare that the Industrial Internship report submitted in partial fulfillment for the degree of Bachelor of Engineering in Computer Engineering, Madhuben and Bhanubhai Patel Institute of Technology, The Charutar Vidya Mandal (CVM) University, Vallabh Vidyanagar, is a bonafide record of work carried out by me at **Tech Elecon Pvt. Ltd.** under the supervision of Mr. Satyam Raval and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

Name of the Student

Sign of Student

Mokshaben Ghanshyamsinh Bodana

---

## **ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to Tech Elecon Pvt Ltd. for providing me with the opportunity to undergo this industrial internship. I extend my heartfelt thanks to my industry mentor, Mr. Satyam Raval, for their invaluable guidance and support throughout the 16-week internship period. Their insights and feedback have been instrumental in shaping my learning experience.

I also wish to convey my deep appreciation to my internal guide, Prof. Ankita Chauhan, and the faculty members of Madhuben and Bhanubhai Patel Institute of Technology for their constant encouragement and advice. Their mentorship has greatly contributed to the successful completion of this report.

Finally, I would like to thank my family and friends for their unwavering support and motivation during this journey.

Mokshaben Ghanshyamsinh Bodana.

# **ABSTRACT**

In today's fast-paced business environment, efficient project management is essential for organizations to meet deadlines, track progress, and collaborate effectively. A well-structured Project Management System ensures seamless coordination between Project Managers, Team Members, and Administrators, improving productivity and task execution.

Our system leverages React.js for the front-end, Java with Spring Boot for the back-end, and MySQL for database management to provide a responsive, scalable, and user-friendly platform. React.js ensures a dynamic user interface, while Java and MySQL offer a robust backend to handle project data efficiently.

This system allows Project Managers to create and assign tasks, track project progress, and manage teams efficiently, while Team Members can view assigned tasks, update their status, and collaborate effectively. The inclusion of role-based access control ensures that each user has the necessary permissions based on their role.

The objective of this system is to simplify project management, automate task tracking, and enhance team collaboration, making it easier for organizations to plan, execute, and complete projects successfully.

Through this internship, I enhanced my technical skills in React.js, API integration, and UI/UX design, while also improving my problem-solving and teamwork abilities. The experience gained during this period has been invaluable in bridging the gap between academic learning and real-world software development.

## List of Figures

Fig 5.1.1 Use Case of Admin .....	24
Fig 5.1.2 Use Case of Project Manager.....	25
Fig 5.1.3 Use Case of Team Member .....	26
Fig 5.2.1 Activity Diagram.....	28
Fig 5.3.1 ER Diagram .....	30
Fig 5.4.1 Context Level DFD .....	33
Fig 5.4.2 First Level DFD of Admin .....	34
Fig 5.4.3 Second Level DFD of Admin .....	34
Fig 5.4.4 Third Level DFD of Admin .....	35
Fig 5.4.5 Fourth Level DFD of Admin .....	35
Fig 5.4.6 First Level DFD of Project Manager.....	36
Fig 5.4.7 Second Level DFD of Project Manager .....	37
Fig 5.4.8 Second Level DFD of Project Manager .....	37
Fig 5.4.9 First Level DFD of Team Member .....	38
Fig 6.1 Home Page.....	44
Fig 6.2 User Login Page.....	44
Fig 6.3 User Register Page .....	45
Fig 6.4 Project Manager Dashboard Page .....	45
Fig 6.5 Project Manager Profile Page .....	46
Fig 6.6 Project Manager Edit Profile Page.....	46
Fig 6.7 Project Manager Project List Page.....	47
Fig 6.8 Project Manager Edit/Add Project Page.....	47
Fig 6.9 Project Manager Tasks List Page.....	48
Fig 6.10 Project Manager Edit/Add Task Page .....	48
Fig 6.11 Project Manager Event Page.....	49
Fig 6.12 Project Manager Team Member Page .....	49
Fig 6.13 Team Member Dashboard .....	50
Fig 6.14 Team Member Profile Page .....	50
Fig 6.15 Team Member Tasks Page.....	51

Fig 6.16 Team Member Page .....	51
Fig 6.17 Admin Login Page .....	52
Fig 6.18 Admin Dashboard Page .....	52
Fig 6.19 Admin User Management Page .....	53
Fig 6.20 Admin Project Management Page.....	54
Fig 6.21 Admin Task Management Page .....	55
Fig 7.1 System Test.....	56



## **List of Tables**

Table 5.5.1 Users Table.....	39
Table 5.5.2 Project Table .....	40
Table 5.5.3 Admin Table.....	40
Table 5.5.4 Task Table .....	41
Table 5.5.5 Team Members Table .....	41
Table 5.5.6 Events Table .....	42
Table 5.5.7 Comments Table.....	42
Table 5.5.8 Files Table .....	43

# Table of Contents

<b>Content</b>	<b>page no.</b>
Acknowledgement	i
Abstract	ii
List of Figures	iii
List of Tables	v
Table of Contents	vi
<b>1. Overview of the company</b>	<b>01</b>
1.1 Company Profile	01
1.2 Different Service's of the Company	01
<b>2. Introduction to Internship and Project</b>	<b>03</b>
2.1 Internship Brief	03
2.2 Overview	03
2.3 Purpose of the Project	04
2.4 Objectives	05
2.5 Scope of the project	06
2.5.1 Features and Functionalities	06
2.6 Limitations Of Project Management System	06
2.7 Technology And Literature Review	07
2.7.1 Technologies Used	07
2.7.2 Literature Review	07
2.8 Project Planning	08
2.8.1 Development Approach and Justification	08
2.8.2 Effort, Time, and Cost Estimation	09
2.8.3 Role And Responsibilities	09
2.8.4 Dependencies	11
2.9 Project Timeline	12
<b>3. Tools and technology</b>	<b>14</b>
3.1 React.js	14
3.2 HTML(markup language)	15

3.3	CSS	15
3.4	Material UI	15
3.5	React router DOM	15
3.6	React hooks	16
3.7	React-Date Picker	16
3.8	React-Bootstrap	16
3.9	React-recharts	17
3.10	Javascript	17
3.11	GitHub	18
<b>4.</b>	<b>System analysis</b>	<b>19</b>
4.1	System Features	19
4.1.1	Front-End Development	19
4.1.2	Dynamic Content and Interactions	19
4.1.3	Form Management and Validation	19
4.1.4	Personalization and User Experience	20
4.1.5	Single-Page Application (SPA) Approach	20
4.2	Feasibility Study	20
4.2.1	Technical Feasibility	20
4.2.2	Operational Feasibility	21
4.2.3	Economic Feasibility	21
4.3	System Software and Hardware Requirements for Web App	22
4.3.1	Development Environment	22
4.3.2	Deployment Environment	22
<b>5.</b>	<b>System Design</b>	<b>23</b>
5.1	Use Case Diagram	23
5.1.1	Use Case of Admin	24
5.1.2	Use Case of Project Manager	25
5.1.3	Use Case of Team Member	26
5.2	Activity Diagram	27
5.2.1	Activity Diagram	28
5.3	ER Diagram	29
5.3.1	ER Diagram	30
5.4	DFD Diagram	31
5.4.1	Context Level DFD	33

5.4.2	First Level DFD of Admin	34
5.4.3	Second Level DFD of Admin(3.0)	34
5.4.4	Third Level DFD of Admin(4.0)	35
5.4.5	Fourth Level DFD of Admin(5.0)	35
5.4.6	First Level DFD of Project Manager	36
5.4.7	Second Level DFD of Project Manager(6.0)	37
5.4.8	Third Level DFD of Project Manager(7.0)	37
5.4.9	First Level DFD of Team Member	38
5.5	Data Dictionary	39
5.5.1	Users Table	39
5.5.2	Projects Table	40
5.5.3	Admin Table	40
5.5.4	Task Table	41
5.5.5	Team Members Table	41
5.5.6	Events Table	42
5.5.7	Comments Table	42
5.5.8	Files Table	43
<b>6.</b>	<b>Implementation</b>	<b>44</b>
6.1	Home page	44
6.2	User Login page	44
6.3	User Register page	45
6.4	Project Manager Dashboard page	45
6.5	Project Manager Profile page	46
6.6	Project Manager Edit Profile page	46
6.7	Project Manager Project List page	47
6.8	Project Manager Edit/Add Project page	47
6.9	Project Manager Task List page	48
6.10	Project Manager Edit/Add Task Page	48
6.11	Project Manager Events Page	49
6.12	Project Manager Team Member page	49
6.13	Team Member Dashboard Page	50
6.14	Team Member Profile Page	50
6.15	Team Member Task Page	51
6.16	Team Member Page	51

6.17	Admin Login Page	52
6.18	Admin Dashboard Page	52
6.19	Admin User Management Page	53
6.20	Admin Project Management Page	54
6.21	Admin Task Management Page	55
<b>7.</b>	<b>Testing</b>	<b>56</b>
7.1	System Test	56
<b>8.</b>	<b>Conclusion</b>	<b>57</b>
<b>9.</b>	<b>References</b>	<b>59</b>

## **1. OVERVIEW OF THE COMPANY**

### **1.1 COMPANY PROFILE:**

Tech Elecon Pvt. Ltd. is the It division of the Elecon group of companies and has more than 25 years of experience in the fields of hardware, software, and networking solutions.

It is situated in the heart of Vithal Udyognagar Industrial Estate and in the proximity of the educational town of Vallabh Vidyanagar. Tech Elecon is all set to reach new heights in the field of IT solutions.

Tech Elecon is ready with all sorts of solutions and delivers any application that is web based and further our solutions are designed to adapt your business rather than your business adapting the software. Their solutions are 100% fruitful and empower you to take control of client's business online and in real time.

Tech Elecon have more than 100 employees with specialized skills in software development, custom software development, and e-commerce software development using custom software programming including .NET, C#.NET, PHP, and Open Source and Oracle.

Tech Elecon delivers quality products and services with a focus on integrating the same with existing technologies, providing the required automation to our customers to help them achieve their business objectives.

Mr. Nilesh Naik, the company's Vice President, is at the helm of the Tech Elecon organization. Mr. Satyam Raval, as Deputy General Manager, and after that, Manager and Associate Manager positions are listed. At the bottom, there are trainees at entry level, who follow up to engineer, senior engineer, also executive and senior executive manager.

### **1.2 DIFFERENT SERVICE'S OF THE COMPANY:**

Tech Elecon has extensive experience in providing IT services and has successfully adapted to technological advancements, making it the leading IT infrastructure management service provider in the region. Our cutting edge delivery model covers all the stages of the solution lifecycle, including planning, deploying, managing, maintaining, auditing, upgrading, and improving.

Tech Elecon recognize that each client has unique needs and expectations when it comes to infrastructure and service providers. Our clients have the flexibility to choose from a wide range of IT infrastructure management and performance services based on their specific requirements. They can opt for on-site services on hybrid solutions that include on-site troubleshooting and support services.

**Tech Elecon provides various services for business:**

- Hardware maintenance and repairing
- Service desk management
- Desktop management
- Network management
- Messaging administrator
- Back-up management

**Other services:**

- Software Development Services
- Software Licensing
- Microsoft Product Implementation
- Linux Servers/Desktop Implementation

## 2. INTRODUCTION TO INTERNSHIP AND PROJECT

### 2.1 INTERNSHIP BRIEF:

I completed my internship at Tech Elecon Private Limited, located in Anand, Gujarat. The duration of the internship was four months, from January to April.

During this period, I worked on a project titled Project Management System(PMS). The aim of this system is to help the company manage its projects more efficiently by keeping records of all projects, their timelines, statuses, assigned teams, and associated tasks.

The system was designed, developed, and implemented from scratch using modern web technologies. Although similar systems exist in the market, this solution is customized to suit the specific needs of the organization and is simpler and more intuitive to use.

The project provided valuable hands-on experience in full-stack development and helped in understanding how real-world project management systems are planned, developed, and deployed in an industrial environment.

### 2.2 OVERVIEW:

The **Project Management System** is designed to help organizations manage their projects, tasks, and team collaboration efficiently. It is a role-based system where users are categorized into three primary roles:

- **Admin** – Manages user accounts and system settings.
- **Project Manager** – Creates and assigns projects, manages tasks, and oversees team members.
- **Team Member** – Views and updates assigned tasks and collaborates with other team members.



This system aims to eliminate the challenges faced in manual project tracking, such as miscommunication, unclear task assignments, and delayed project completion. It provides a centralized platform where users can log in, access relevant project details, track progress, and ensure tasks are completed on time.

With a secure authentication system, only authorized users can access specific functionalities based on their roles. The system ensures data integrity and improves work efficiency through a structured workflow.

### 2.3 PURPOSE OF THE PROJECT:

The main purpose of this Project Management System is to help organizations manage their projects, tasks, and team members efficiently in one place. It simplifies project tracking by allowing project managers to assign tasks, monitor progress, and manage team collaboration easily. Team members can view their assigned tasks, update their work status, and communicate with their team, ensuring smooth project execution.

With a well-defined role-based system, each user has specific access and responsibilities. Admins manage system users, Project Managers create and monitor projects, and Team Members focus on their assigned tasks. This ensures that everyone knows their role, reducing confusion and improving work efficiency.

By using this system, organizations can:

- **Save time** by reducing manual work.
- **Improve communication** between managers and team members.
- **Track project progress** in real-time.
- **Ensure tasks are completed on time** with better planning and task distribution.

the system enhances **accountability** by maintaining detailed records of all projects, tasks, and user activities. Project Managers can easily track who is responsible for each task and follow up when needed, leading to better performance and transparency across the team.

## 2.4 OBJECTIVES:

- **User Role Management**
  - Allow users to register and log in based on their roles (Admin, Project Manager, and Team Member).
  - Provide **secure authentication** to prevent unauthorized access.
- **Project Management**
  - Enable **Project Managers** to create, edit, and delete projects.
  - Provide a clear **project overview**, including deadlines, progress tracking, and assigned tasks.
  - Allow easy **search and filter options** to manage multiple projects efficiently.
- **Task Assignment and Tracking**
  - **Project Managers** can assign tasks to **Team Members** and set deadlines.
  - Team Members can **update task progress** (e.g., In Progress, Completed, and Pending).
  - Provide a **task filtering system** to view tasks based on status, priority, or project.
- **User-Friendly Interface**
  - Ensure an **intuitive design** with a clean and simple UI.
  - Provide **clear navigation** for different roles to access their features easily.
  - Implement **notifications** to keep users informed about updates and deadlines.
- **Secure System and Data Management**
  - Store all project and task details **securely** in a backend database.
  - Restrict access to **authorized users** only.
- **Collaboration and Team Communication**
  - Provide **visibility into task assignments** so team members know their responsibilities.
  - Allow **editing and updating of tasks in real-time** to improve collaboration.
  - Enable **efficient teamwork** by ensuring each member knows their role in a project.

## 2.5 SCOPE OF THE PROJECT:

The **scope** of this Project Management System defines its functionalities, features, and limitations. The system is designed for **small to medium-sized teams and organizations** looking to streamline their **project planning, execution, and tracking**.

### 2.5.1 Features and Functionalities:

#### 1. Admin Role:

- Manage user accounts (Project Managers and Team Members).
- Monitor system usage and ensure smooth operations.

#### 2. Project Manager Role:

- Create, edit, and manage projects.
- Assign tasks to team members and set deadlines.
- View and manage team members involved in a project.

#### 3. Team Member Role:

- View assigned tasks and update their status.
- Collaborate with other team members for task completion.
- Access team member profiles to understand project responsibilities.

## 2.6 LIMITATIONS OF PROJECT MANAGEMENT SYSTEM:

- **No Real-Time Alerts:** Updates such as new tasks or project changes are not instantly pushed; real-time notifications can be a future enhancement.
- **Only Web-Based:** The system is limited to browser access; no mobile app version exists yet.
- **No Offline Support:** Requires a stable internet connection to interact with the backend (Java + MySQL).

## 2.7 TECHNOLOGY AND LITERATURE REVIEW:

The Project Management System was developed using a modern and scalable web stack that enables smooth functionality, cross-browser compatibility, and a user-friendly interface.

### 2.7.1 Technologies Used:

- **Frontend:**
  - **React.js:** Used for building dynamic user interfaces through components.
  - **Bootstrap CSS:** Utilized for responsive layout and pre-built UI components.
  - **React Router DOM:** Handles page routing and role-based view control.
  - **React Icons:** For clean and intuitive iconography in the interface.
- **Backend:**
  - **Java Servlets (via NetBeans IDE):** Manages request-response operations between the frontend and backend.
  - **MySQL:** Stores persistent data (users, projects, tasks, etc.).
  - **JDBC (Java Database Connectivity):** Used to execute SQL queries.
- **Authentication:**
  - Session-based login management using cookies and servlet sessions.
  - Role-based views for Admin, Project Manager, and Team Member.

### 2.7.2 Literature Review:

Many project management tools like Jira, Trello, and Zoho Projects can be expensive or too complicated for small teams. Our Project Management System (PMS) solves this problem by providing:

- A simple system with clear roles for each user.
- Easy-to-use dashboards and task management features.
- Visual tracking of task status (To Do, In Progress, Completed).
- Features like file upload, announcement boards, and PDF report generation.

## 2.8 PROJECT PLANNING:

### 2.8.1 Development Approach and Justification:

The development of the **Project Management System (PMS)** was carried out using a structured and role-based approach tailored to meet the specific needs of Admins, Project Managers, and Team Members. The workflow was divided into multiple phases to ensure smooth implementation and easy handling of different user functionalities:

- **Requirement Analysis:** First, the specific requirements for each role (Admin, Project Manager, Team Member) were identified. Admins manage users, Project Managers handle projects and tasks, and Team Members focus on completing assigned tasks.
- **Design Phase:** A clean and user-friendly frontend was designed using **React.js** and **Bootstrap** to create separate dashboards and pages for each role, ensuring clarity and ease of navigation.
- **Backend Development:** The backend was developed in **Java (Servlets)** using **NetBeans IDE**, with the business logic supporting different operations like user management, project creation, task assignment, and task updates.
- **Database Design:** A structured database was created using **MySQL**, with tables such as users, projects, tasks, and project\_members to handle all system data efficiently.
- **API Integration and Testing:** REST APIs were developed and tested using tools like **Thunder Client** to ensure smooth communication between frontend and backend for functionalities like login, project management, task management, and team handling.
- **Testing and Debugging:** Manual testing was performed for all major modules like login workflows, project handling, task updates, and role-specific functionalities to identify and fix bugs early.
- **Deployment:** The system was hosted on a **local GlassFish Server** connected to MySQL to demonstrate a fully working Project Management System.

### 2.8.2 Effort, Time, and Cost Estimation:

The Project Management System (PMS) was developed as an individual effort over a span of approximately 3 months (around 12–13 weeks). The estimated weekly time allocation across different phases was as follows:

- **Planning & Requirement Analysis:** 2 weeks
- **Frontend Development (React.js with Bootstrap):** 4 weeks
- **Backend Development (Java Servlets using NetBeans) and MySQL Database Integration:** 4 weeks
- **Testing and Debugging:** 2 weeks
- **Deployment on Local GlassFish Server and Finalization:** 1 week

Since all the tools and technologies used were **open-source** or available under free licensing, the cost of development was zero. This includes the use of React.js, Bootstrap CSS, NetBeans IDE, GlassFish Server, and MySQL Community Edition for database management.

### 2.8.3 Role and Responsibilities:

Since the **Project Management System (PMS)** was developed as an individual project, all responsibilities were undertaken independently. Each aspect of the system was carefully planned and executed to ensure a seamless and efficient solution for managing projects, tasks, and team collaboration. Several challenges were faced, such as handling role-based access and ensuring smooth communication between the frontend and backend. However, I received valuable guidance and support from my mentor throughout the project, which helped me improve the system and overcome obstacles. Their feedback and encouragement were crucial in motivating me to successfully complete the project.

#### Key Responsibilities Included:

##### 1. Frontend Responsibilities (React + Bootstrap):

- Designed and implemented a responsive UI using React.js and Bootstrap for Admins, Project Managers, and Team Members.
- Integrated frontend with backend APIs for real-time interaction, including task updates, project management, and user authentication.

**2. Backend Responsibilities (Java + MySQL):**

- Built RESTful APIs using Java Servlets and NetBeans for handling user login, project creation, task assignments, and data retrieval.
- Implemented JWT-based authentication and role-based access control (RBAC) to ensure that only authorized users can access specific sections of the system (Admin, Project Manager, Team Member).
- Managed data with MySQL for storing user profiles, project details, and task information.

**3. Planning and Documentation:**

- Planned project phases and defined clear milestones to track progress and ensure all features (like user login, project management, and task assignment) were completed on time.
- Maintained documentation for each phase and testing reports to ensure smooth development and identify issues early.

**Key Contributions:**

- Developed frontend components using React.js and integrated Bootstrap CSS for responsive design.
- Built backend RESTful APIs using Java servlets (NetBeans) and connected to MySQL for data management.
- Implemented user authentication and role-based access control (RBAC) for Admin, Project Manager, and Team Member roles.
- Integrated MySQL database to manage data for users, projects, tasks, and team members.
- Ensured smooth user experience by managing role-specific views and actions (Admin, Project Manager, Team Member).
- Tested and debugged the system to ensure seamless operation across different user roles.

### 2.8.4 Dependencies:

The Project Management System (PMS) relies on several key external dependencies and libraries to ensure optimal performance, security, and functionality

#### Frontend Dependencies:

- **React.js** – For building dynamic and interactive user interfaces.
- **React Router** – For handling client-side routing.
- **React Context API** – For global state management across components.
- **Bootstrap CSS** – For responsive, mobile-first design and pre-styled UI components.
- **React Icons** – To include vector icons for clean visuals.
- **Axios** – For making HTTP requests to interact with the backend API.
- **React-Bootstrap** – For Bootstrap components in React.

#### Backend Dependencies:

- **Java (JDK 11)** – For building the backend logic of the system using servlets.
- **MySQL** – For relational data storage, managing tables for users, projects, tasks, and more.
- **Java Servlets (NetBeans)** – For handling HTTP requests and creating RESTful APIs.
- **JDBC (Java Database Connectivity)** – For database connectivity in Java.
- **JWT (JSON Web Token)** – For secure token-based authentication and authorization.
- **BCrypt** – For encrypting user passwords.
- **Apache Commons** – For utility functions such as file handling.
- **GlassFish 6.1.0** – For server-side deployment of Java web applications.

#### Testing Tools:

- **Jest** – For unit testing React components and ensuring frontend functionality.
- **JUnit** – For testing backend Java servlets and business logic.



## 2.9 PROJECT TIMELINE:

The **Project Management System (PMS)** project was completed over a duration of **16 weeks**. Below is the timeline outlining the key activities and milestones achieved each week — including learning, development, testing, deployment, and documentation.

### Week 1–4: Pre-Development Learning

- **Week 1:** Learned HTML fundamentals and practiced basic markup.
- **Week 2:** Studied CSS styling techniques and layout systems.
- **Week 3:** Learned JavaScript basics including DOM manipulation and events.
- **Week 4:** Revised HTML, CSS, and JavaScript to strengthen frontend foundation.

### Week 5–6: React & Requirement Gathering

- **Week 5:** Began React.js — learned components, props, hooks, and routing.
- **Week 6:** Received project definition, researched similar systems, and conducted requirement gathering.

### Week 7: UI Design

- Created initial wireframes and layout plans for the Admin, Project Manager, and Team Member dashboards.
- Designed the navigation structure and component hierarchy for each role to ensure a smooth user experience across the system. Planned out elements for project management, task management, and user role-based views.

### Week 8–10: Frontend Development

- **Week 8:** Set up the React.js project, installed necessary dependencies (React Router, Bootstrap). Developed the login page, sidebar, and routing for Admin, Project Manager, and Team Member roles.
- **Week 9:** Developed the Dashboard page with components for quick project/task overview and role-specific data. Implemented role-based routing and conditional views for different user roles.

- **Week 10:** Created the Profile Management page for users (Admin, Project Manager, Team Member), implemented Project creation and viewing, and built Task management with dynamic data integration from the backend.

### **Week 11–13: Backend Development & API Integration**

- **Week 11:** Set up the backend using Java Servlets (NetBeans) and connected it to a MySQL database. Created tables for users, projects, tasks, and roles.
- **Week 12:** Implemented JWT authentication for secure user login and role-based access control (RBAC) for Admin, Project Manager, and Team Member roles. Developed APIs to handle user management, project creation, task assignment, and team management.
- **Week 13:** Integrated the frontend with the backend using **Axios** for API calls. Tested the API connections with **Postman** to ensure that data was flowing correctly between the frontend and backend.

### **Week 14–15: Testing & Bug Fixes**

- **Week 14:** Performed manual testing of all modules (Admin, Project Manager, Team Member) to verify functionality. Fixed issues related to data handling, user permissions, and project/task updates.
- **Week 15:** Optimized performance, refined the UI, improved error handling (e.g., form validation). Ensured that task status updates (To Do, In Progress, Completed) were functioning correctly for all roles.

### **Week 16: Deployment & Documentation**

- Deployed the frontend and backend locally on your development environment using Java (NetBeans) and MySQL.
- Prepared project documentation, user manual, and final report for submission.

### 3. TOOLS AND TECHNOLOGY

#### 3.1 REACT.JS:

React.js is a JavaScript library for building user interfaces, primarily for web applications. It enables developers to create interactive UI components efficiently by using a component-based architecture. Here's a breakdown:

- **Component-Based:** React breaks down the UI into reusable components, each responsible for rendering a small, self-contained part of the UI. Components can be nested within each other, allowing for complex UI structures.
- **Virtual DOM:** React uses a virtual DOM to improve performance. Instead of directly manipulating the browser's DOM, React creates a lightweight virtual representation of the DOM in memory. When the state of a component changes, React compares the virtual DOM with the real DOM and only updates the parts that have changed, minimizing DOM manipulation and increasing performance.
- **JSX:** JSX is a syntax extension for JavaScript that allows you to write HTML-like code within JavaScript. It simplifies the process of creating React elements, making the code more readable and expressive.
- **Unidirectional Data Flow:** React follows a unidirectional data flow, meaning data flows in one direction from parent to child components. This makes it easier to understand how data changes over time and debug applications.
- **State Management:** React components can have state, which represents the data that change over time. When the state of a component changes, React automatically re-renders the component, updating the UI to reflect the new state.
- **Lifecycle Methods:** React components have lifecycle methods that allow developers to hook into various points in a component's lifecycle, such as when it is first mounted or updated. This enables developers to perform actions like fetching data or cleaning up resources at the appropriate times.
- **Declarative Syntax:** React uses a declarative syntax, where you describe what you want the UI to look like, and React takes care of updating the DOM to match that description. This makes it easier to reason about your code and build complex UIs.

### **3.2 HTML (MARKUP LANGUAGE):**

HTML is the standard markup language used to create the structure and content of web pages. It consists of a series of elements, represented by tags, which define the different parts of a web page such as headings, paragraphs, images, links, and more. HTML elements are organized in a hierarchical structure, with nested elements representing the relationship between different parts of the content. HTML provides semantic meaning to the content, making it accessible to both users and search engines.

### **3.3 CSS:**

CSS is a style sheet language used to control the presentation and layout of HTML elements on a web page. It allows developers to define styles such as colors, fonts, spacing, and positioning, making it possible to create visually appealing and responsive designs. CSS operates by selecting HTML elements and applying styles to them using selectors and declaration blocks. CSS can be applied inline within HTML documents, embedded within `<style>` tags in the document's `<head>` section, or linked externally to the HTML document as a separate stylesheet.

### **3.4 MATERIAL UI:**

Material-UI is a popular React component library that implements Google's Material Design guidelines. It provides pre-designed and customizable UI components, such as buttons, cards, menus, forms, and more, to help developers build modern and visually appealing web applications with ease. Material-UI components are built with React, making them easy to integrate into React projects and leverage React's component-based architecture. Additionally, Material-UI offers extensive theming capabilities, allowing developers to customize the look and feel of their applications to match their brand or design preferences. With its rich set of components, thorough documentation, and active community support, Material-UI simplifies the process of creating responsive and intuitive user interfaces for React-based web applications.

### **3.5 REACT-ROUTER-DOM:**

React Router DOM is a popular routing library for React applications, enabling developers to handle navigation and routing within single-page applications (SPAs). It allows you to define different routes in your application, each corresponding to a

specific URL path, and render different components based on the current URL. React Router DOM provides a ``<BrowserRouter>`` component to manage browser history using HTML5 history API, allowing for navigation without full page reloads. It also offers various route components like ``<Route>``, ``<Switch>``, ``<Redirect>``, and ``<Link>`` to define route matching, switch between routes, redirect users, and create navigation links, respectively. With React Router DOM, developers can create dynamic and interactive web applications with multiple views, enabling seamless navigation between different pages while maintaining a single-page experience.

### **3.6 REACT HOOKS:**

React Hooks allow functional components to use state and other React features without needing class components. They include `useState` for state management, `useEffect` for side effects, `useContext` for accessing context, `useReducer` for more complex state logic, `useCallback` and `useMemo` for performance optimization, and `useRef` for creating mutable references. Hooks provide a more concise and readable way to manage state and side effects in React applications, promoting cleaner and more maintainable code.

### **3.7 REACT-DATE PICKER:**

React-DatePicker is a flexible and customizable date picker component for React applications, widely used for selecting dates and times within forms or other user interfaces. It is designed to be user-friendly and offers a range of features to enhance date selection, such as customizable date and time formats, support for date range selection, and localization options. This makes it ideal for applications requiring precise date inputs, like booking systems or scheduling applications. React-DatePicker integrates seamlessly with form libraries and state management tools, making it easy to manage form data and validations. Additionally, it offers extensive theming options to ensure the date picker matches the look and feel of the application.

### **3.8 REACT-BOOTSTRAP**

React-Bootstrap is a popular library that brings the power of Bootstrap components to React applications, providing a comprehensive set of UI components that adhere to the Bootstrap design system. This ensures consistency and responsiveness across different devices and screen sizes. React-Bootstrap includes a wide range of pre-built components

such as buttons, modals, forms, and navigation bars, which can be easily customized and used in projects. Its integration with Bootstrap makes building responsive and mobile-friendly user interfaces straightforward. The library also allows for extensive customization of component styles to match the application's branding, ensuring a visually appealing and accessible user experience.

### 3.9 REACT-RECHARTS:

React-Recharts is a composable charting library built on React components, designed to be simple to use, highly customizable, and performant. It supports various types of charts, including line charts, bar charts, pie charts, and scatter plots, making it ideal for creating data visualizations in React applications. React-Recharts offers extensive customization options for each chart type, including colors, legends, tooltips, and animations. Its composability allows for the combination of different chart components to create complex and interactive data visualizations. The library ensures that charts are responsive and adapt to different screen sizes and devices, making it suitable for displaying business metrics, scientific data, and interactive dashboards.

### 3.10 JAVASCRIPT:

JavaScript is a high-level, interpreted programming language primarily used for web development.

- **Dynamic:** JavaScript is dynamic, meaning it can adapt and change as the program runs. This makes it well-suited for creating interactive web pages.
- **Client-Side Scripting:** JavaScript is mainly used as a client-side scripting language in web browsers, allowing developers to create dynamic and interactive web pages by manipulating the HTML and CSS content.
- **Event-Driven:** JavaScript is event-driven, meaning it can respond to user actions such as mouse clicks, keyboard inputs, and form submissions. This enables developers to create responsive and interactive user interfaces.
- **Functional and Object-Oriented:** JavaScript supports both functional and object-oriented programming paradigms, allowing developers to write code in a variety of styles. It also features first-class functions, closures, and prototypes, which contribute to its flexibility and expressiveness.

- **Cross-Platform:** JavaScript is supported by all major web browsers and can run on various platforms, including desktops, mobile devices, and servers. This makes it a versatile language for building a wide range of applications.
- **Libraries and Frameworks:** JavaScript has a rich ecosystem of libraries and frameworks, such as React, Angular, and Vue.js, which streamline the development process and provide additional features and functionalities for building web applications.

### 3.11 GITHUB:

GitHub is a web-based platform and version control system that allows developers to collaborate on projects, host code repositories, and manage software development workflows. Here's a brief overview:

- **Version Control:** GitHub utilizes Git, a distributed version control system, to track changes to files and manage codebase versions. Developers can create branches to work on features or fixes independently, merge changes back into the main branch, and revert to previous versions if needed.
- **Code Hosting:** GitHub provides a centralized platform for hosting Git repositories, making it easy for developers to share their code with others. Repositories can be public or private, and GitHub offers features like issue tracking, project boards, and wikis to support collaboration and project management.
- **Collaboration:** GitHub facilitates collaboration among developers by enabling them to fork repositories, make changes, and submit pull requests to the original repository maintainers for review and integration. Teams can discuss code changes, review proposed modifications, and provide feedback using comments and code reviews.
- **Community and Open Source:** GitHub is widely used by the open-source community to contribute to projects, discover new libraries and frameworks, and engage with other developers. It hosts millions of public repositories covering a diverse range of topics, from web development and data science to machine learning and gaming.

## 4. SYSTEM ANALYSIS

### 4.1 SYSTEM FEATURES:

#### 4.1.1 Front-End Development

- **Responsive Design:**
  - The system will be built using **React.js**, ensuring it works smoothly on desktops, tablets, and mobile devices.
  - The UI will automatically adjust to different screen sizes for a **consistent and user-friendly experience**.

#### 4.1.2 Dynamic Content and Interactions

- **Real-Time Updates:**
  - Users will get instant notifications and updates about **projects, tasks, and team activities**.
  - The system ensures that all changes, like adding or updating tasks, reflect in real time.
- **Interactive Features:**
  - **Buttons, pop-ups, and animations** will improve user interaction and navigation.
  - Users can **easily add, edit, and manage projects, tasks, and members** with interactive elements.

#### 4.1.3 Form Management and Validation

- **Easy-to-Use Forms:**
  - **Project Managers** can **create projects, assign tasks, and update statuses** using simple forms.
  - **Team Members** can **update their task progress** effortlessly.
- **Secure Login and Signup:**
  - The system will include a **secure authentication process** for login and registration.
- **Error Checking:**
  - Forms will automatically **validate user inputs**, checking for missing fields or incorrect data.
  - Instant feedback will help user's **correct errors before submission**.



#### 4.1.4 Personalization and User Experience

- **Custom Dashboards:**
  - Each user will have a personalized dashboard showing their **assigned tasks, project status, and team members**.
- **Role-Based Access Control:**
  - **Admins:** Manage user accounts and system settings.
  - **Project Managers:** Oversee projects, assign tasks, and manage team members.
  - **Team Members:** View and update their tasks.
- **Smooth and Responsive UI:**
  - **Fast navigation, clean design, and easy accessibility** for all users.

#### 4.1.5 Single-Page Application (SPA) Approach

- **Fast and Smooth Navigation:**
  - The system is built as a **Single-Page Application (SPA)** using **React.js**.
  - This ensures **quick loading** and **seamless transitions** between different sections without full page reloads.

### 4.2 FEASIBILITY STUDY:

#### 4.2.1 Technical Feasibility

- **Technology Stack:**

The system uses React.js for the front end to create a smooth and interactive user experience. The backend is developed using Java (Spring Boot) to handle server side logic efficiently. MySQL is used for securely storing and managing project, task, and user data.
- **Infrastructure:**

The system is hosted on a cloud-based platform like AWS or Digital Ocean, ensuring scalability and reliability. The MySQL database is structured to handle large volumes of data efficiently, ensuring smooth performance. The backend server using Java (Spring Boot) ensures a fast and stable connection between users and the database.
- **Development Resources:**

React.js, Java (Spring Boot), and MySQL are widely used technologies, making it easier to find developers for future updates.

#### 4.2.2 Operational Feasibility:

- **User Accessibility:**

The Project Management System is a web-based platform, accessible from any device with an internet connection, including desktops, laptops, tablets, and mobile phones. The responsive design ensures a smooth user experience across all screen sizes. Role-based access ensures that Admins, Project Managers, and Team Members only see the features relevant to them.

- **Maintenance:**

The system is built using a component-based architecture in React.js, making it easy to update and maintain. UI improvements, bug fixes, and feature enhancements can be deployed without affecting the entire system. Regular updates ensure smooth performance, security enhancements, and compatibility with new web standards.

- **Training and Support:**

The system is designed to be user-friendly, minimizing the need for extensive training. However, user guides, tutorials, and customer support will be available to help users navigate and effectively use the platform. Interactive onboarding features and tooltips further enhance user learning and engagement.

#### 4.2.3 Economic Feasibility

- **Development Cost:**

The Project Management System is built using open-source technologies like React.js, Java (Spring Boot), and MySQL, which reduces development costs. Initial expenses include server hosting, domain registration, and developer resources, but these are manageable with efficient planning.

- **Operational Costs:**

Since the system is web-based, there is no need for expensive hardware. Cloudbased hosting services like AWS, DigitalOcean, or Firebase ensure cost-effective scalability. Regular maintenance, security updates, and minor improvements will have minimal ongoing costs.

- **Long-Term Benefits:**

The system will help increase productivity, reduce miscommunication, and improve project tracking, leading to cost savings in management and efficiency.

## 4.3 SYSTEM SOFTWARE AND HARDWARE REQUIREMENTS FOR WEB APP

### 4.3.1 Development Environment:

- **Operating System:**
  - Recommended: Windows 10 or 11, macOS 10.10 or later, Ubuntu 16 or later (for best compatibility and performance)
  - Minimum: Most modern operating systems with a compatible web browser should work (may require additional configuration)
- **Hardware:**
  - Minimum:
    - Processor: Dual-core processor (e.g., Intel Pentium 4 or equivalent)
    - RAM: 4 GB
    - Storage: 10 GB free disk space
  - Recommended:
    - Processor: Quad-core processor (e.g., Intel Core i5 or equivalent)
    - RAM: 8 GB or more
    - Storage: 50 GB or more (depending on project complexity and additional tools)
- **Software:**
  - **Frontend:** React.js for a responsive user interface.
  - **Backend:** Java (Spring Boot) for handling business logic.
  - **Development Tools:** Visual Studio Code, IntelliJ IDEA, or Eclipse for coding.
  - **Database:** MySQL for storing and managing project, task, and user data.
  - **Package Managers:** npm (for React.js) and Maven (for Java).
  - **Version Control:** Git/GitHub for code collaboration and tracking.

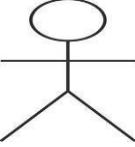
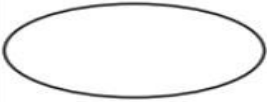

### 4.3.2 Deployment Environment

The deployment environment refers to where the web application is hosted and operated after development. It includes the requirements for both the **frontend (React.js)** and **backend (Java with MySQL)** components.

## 5. SYSTEM DESIGN

### 5.1 USE CASE DIAGRAM:

- A Use Case Diagram is a vital tool in system design. It provides a visual representation of how users interact with a system.
- A Use Case Diagram is a type of Unified Modeling Language (UML) diagram that represents the interaction between actors (users or external systems) and a system under consideration to accomplish specific goals.

<u>Symbol</u>	<u>Name</u>	<u>Description</u>
	Actor	Actors are external entities that interact with the system. These can include users, other systems.
	Use Case	Use cases are like scenes in the play. They represent specific things your system can do
	Boundary	The system boundary is a visual representation of the scope or limits of the system you are modeling. It defines what is inside the system and what is outside.

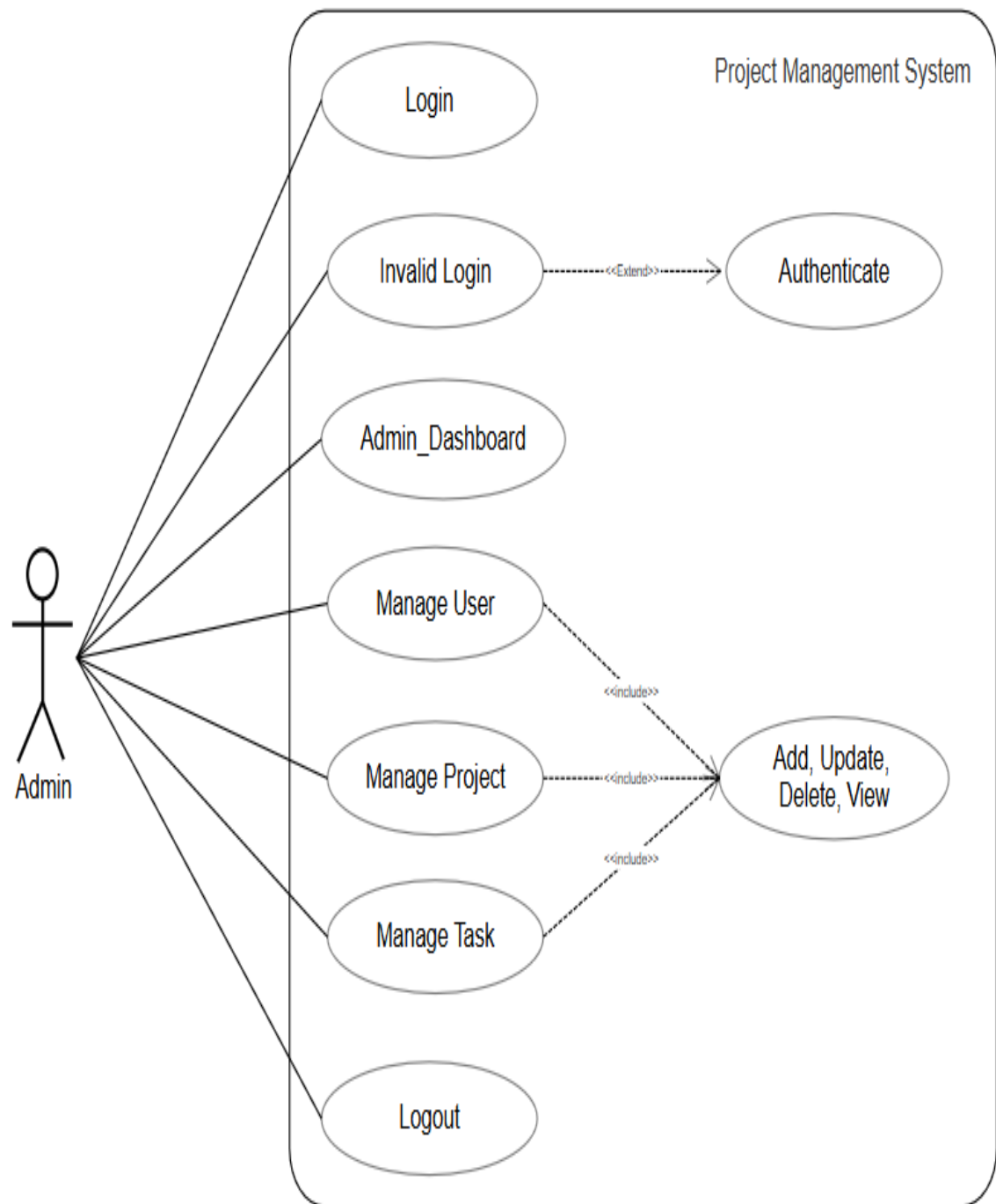
**5.1.1 Use Case of Admin:**

Fig 5.1.1 Use Case of Admin

### 5.1.2 Use Case of Project Manager:

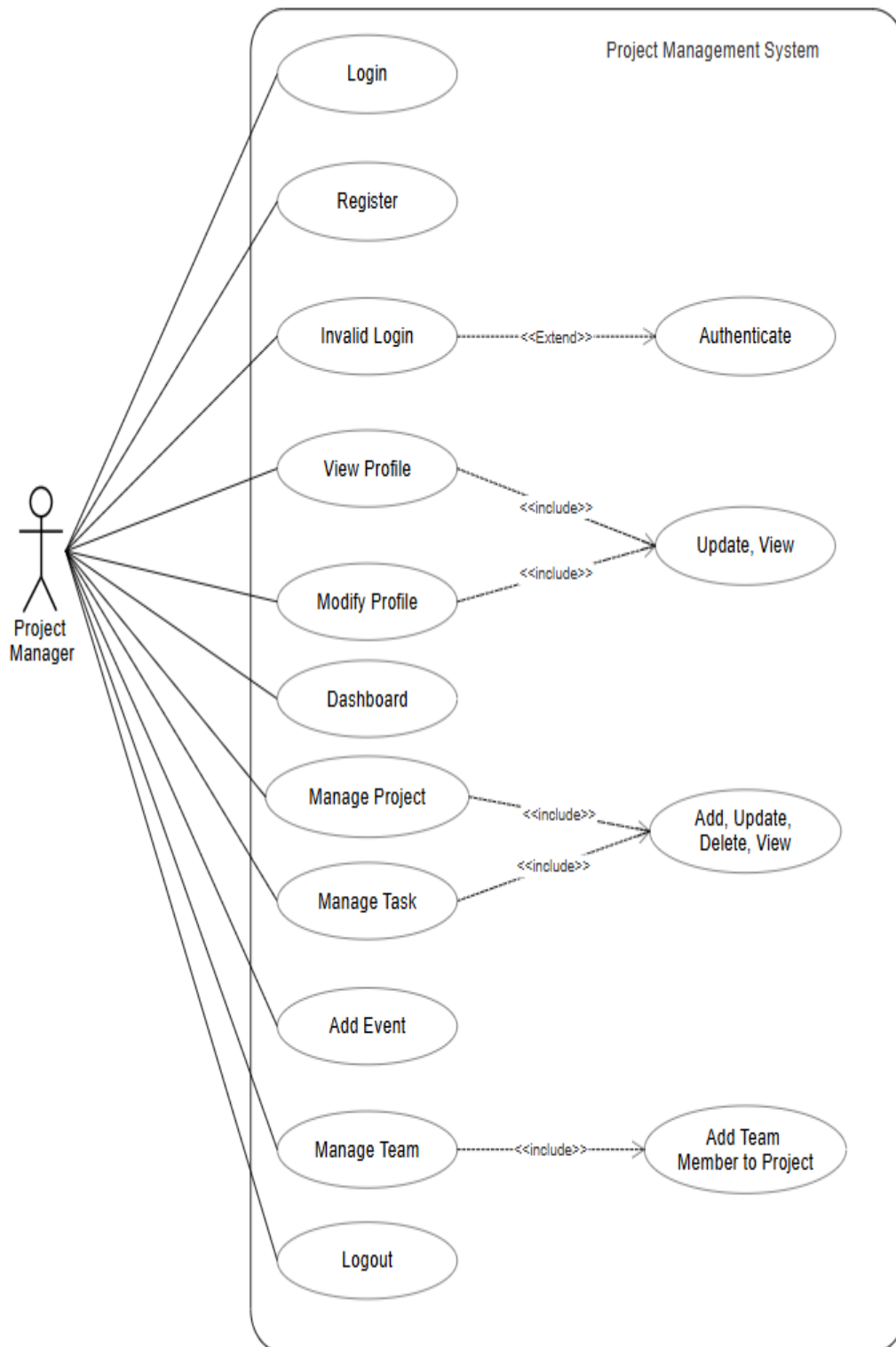


Fig 5.1.2 Use Case of Project Manager

### 5.1.3 Use Case of Team Member:

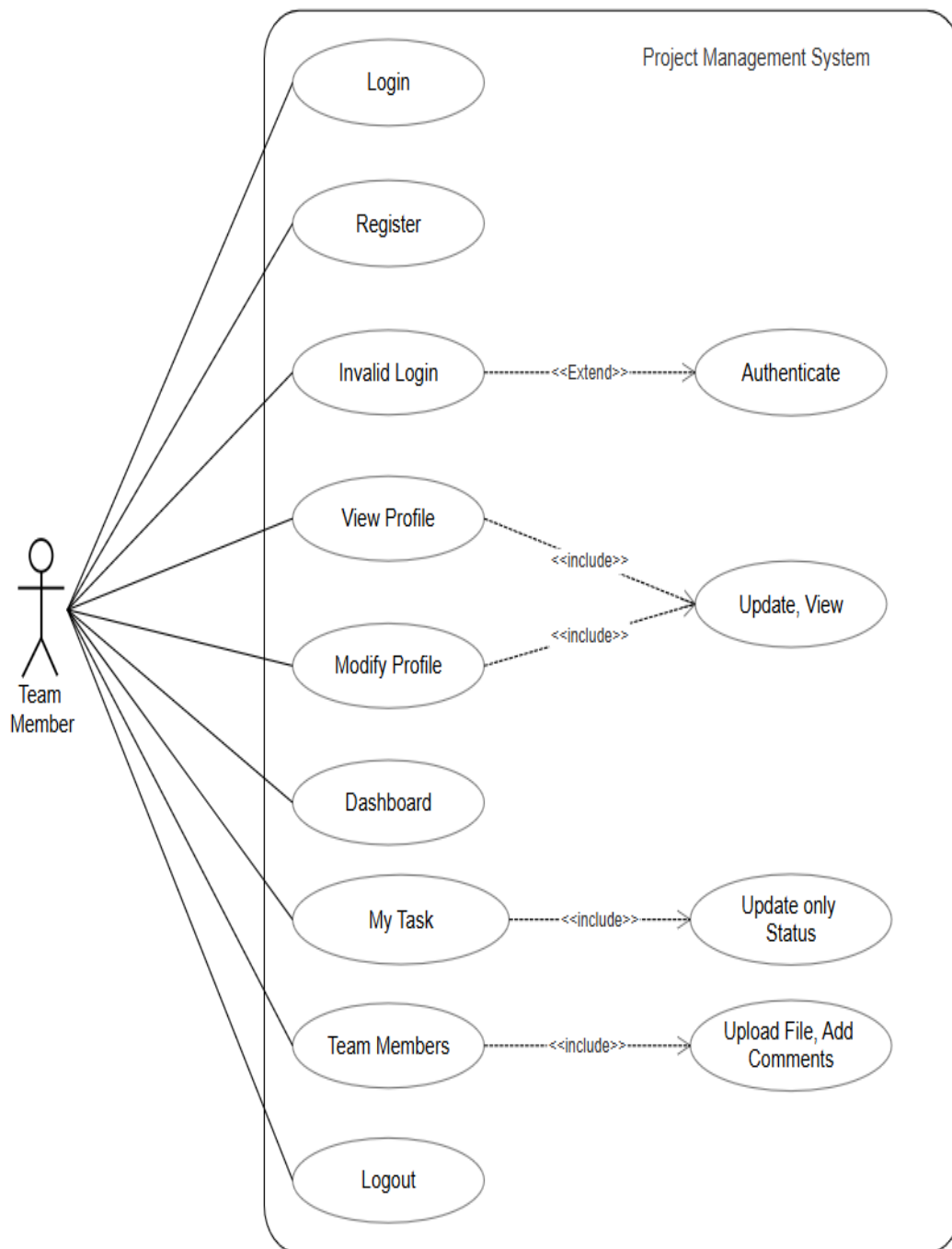

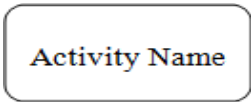
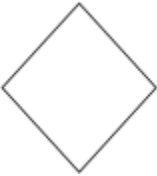




Fig 5.1.3 Use Case of Team Member

## 5.2 Activity Diagram:

- Activity diagram represents the dynamic aspects of the system.
- Activity diagram is basically a flow chart to represent the flow from one activity to another activity.
- **Activity Diagram Notations:** Activity diagram notations can be generated using the following notations.

<u>Symbol</u>	<u>Name</u>	<u>Description</u>
	Initial State	The starting stage before an activity takes place is depicted as the initial state.
	Activity Box	It represents an activity that is connected to other activities.
	Decision Box	It is a diamond shape box which represents a decision with alternate paths. It represents the flow of control.
	Parallel Activity	It is long, flat rectangle that is used to represent the parallel activities.
	Final State	The state where a system reaches when a specific process ends is known as final state.



### 5.2.1 Activity Diagram:

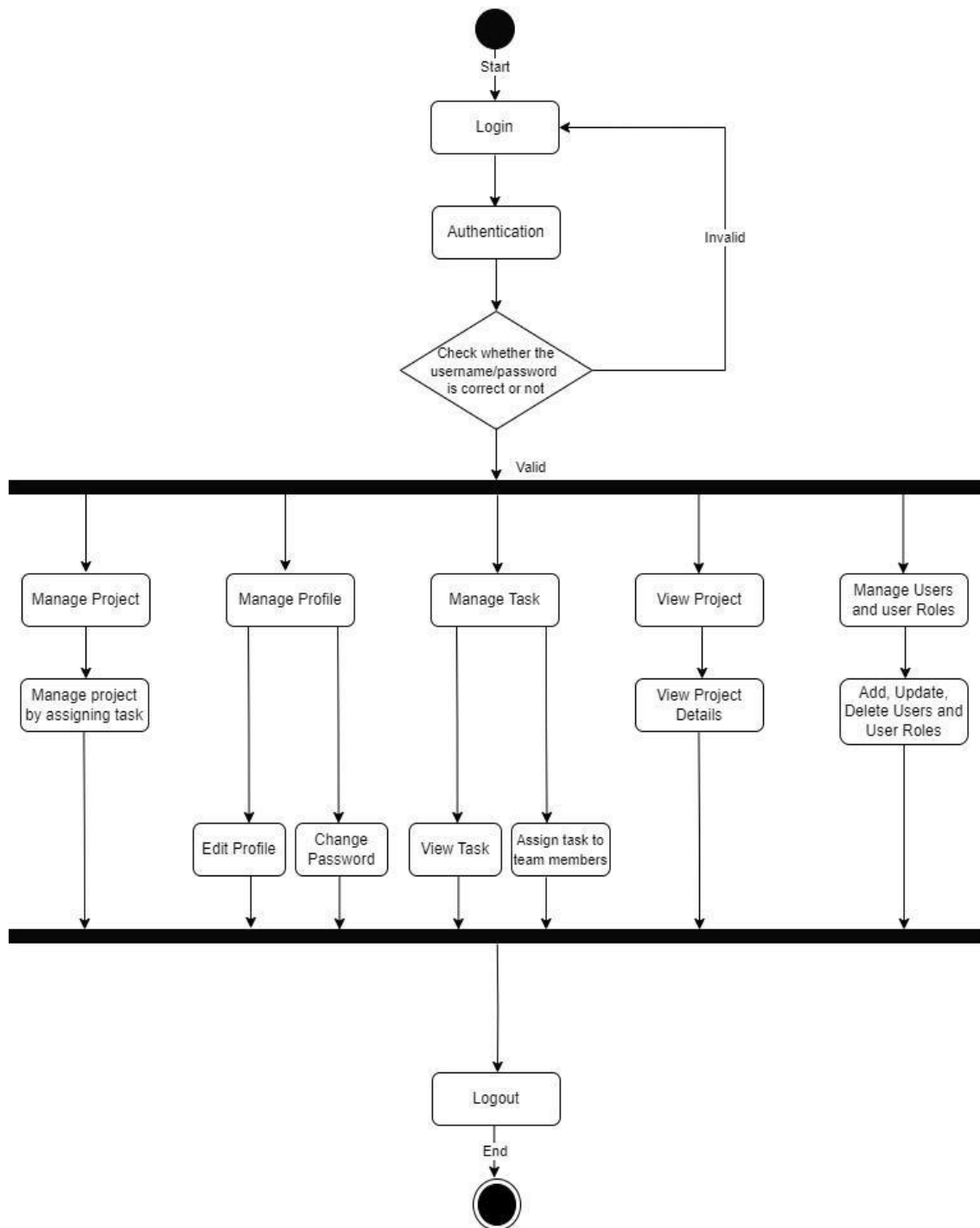
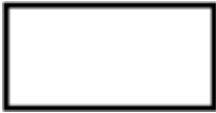

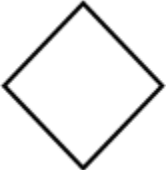



Fig 5.2.1 Activity Diagram

### 5.3 ER Diagram:

- The Entity Relational Model is a model for identifying entities to be represented in the database and representation of how those entities are related.
- The ER data model specifies enterprise schema that represents the overall logical structure of a database graphically.
- The Entity Relationship Diagram explains the relationship among the entities present in the database.
- **ER Diagram Notations:** Activity diagram notations can be generated using the following notations.

<u>Symbol</u>	<u>Name</u>	<u>Description</u>
	Rectangle	It represents the entities in ER Model
	Ellipse	It represents the attributes in ER Model
	Diamond	It represents the relationships in ER Model
	Line	It represents the attributes to entities and entity sets with other relationship types

### 5.3.1 ER Diagram:

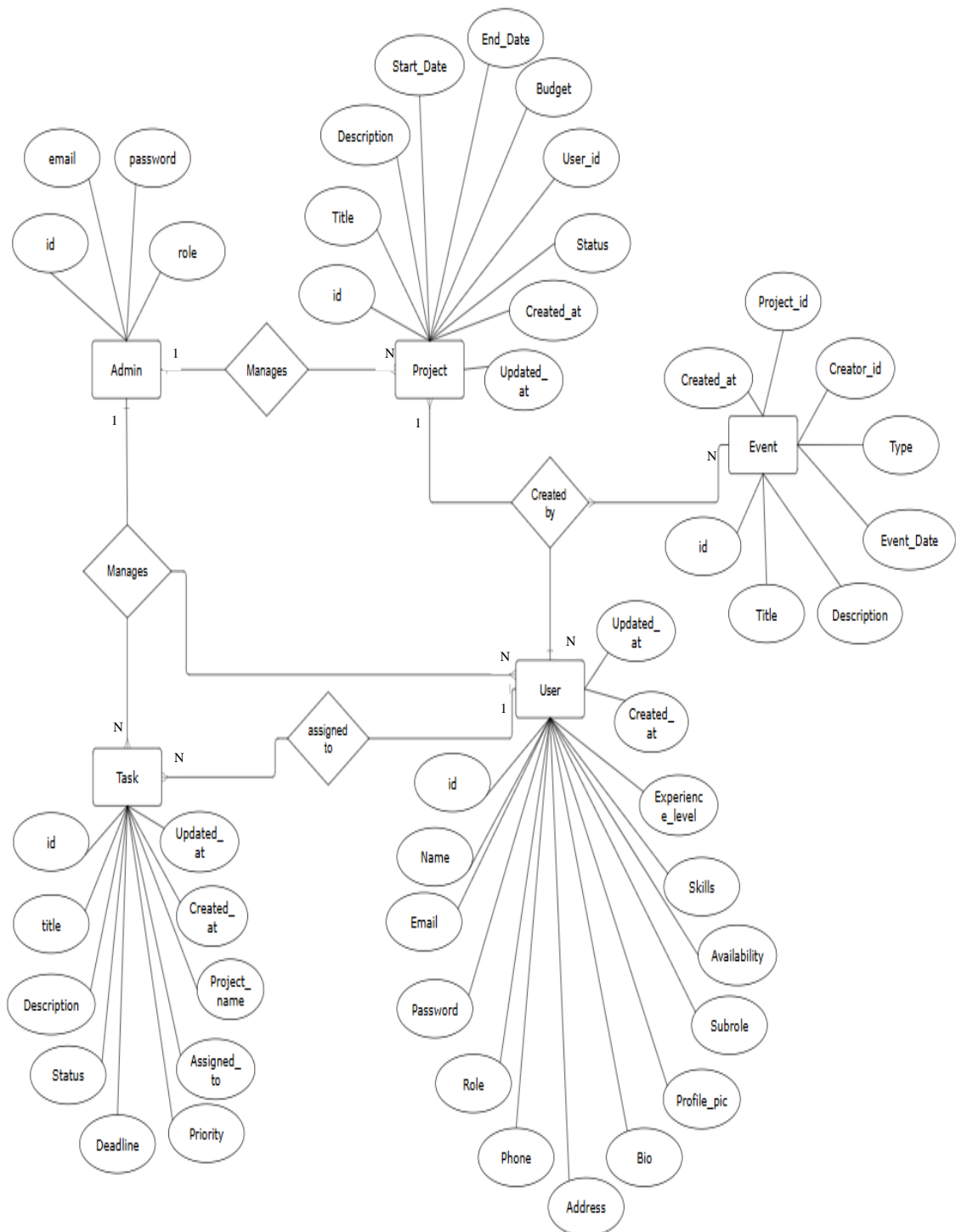

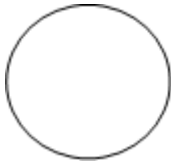




Fig 5.3.1 ER Diagram

## 5.4 DFD DIAGRAM:

- A data flow diagram is a graphical tool used to describe and analyze movement of data through a system.
- These are the central tool and the basis from which the other components are developed.
- The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams.
- The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations.
- A full description of a system actually consists of a set of data flow diagrams. Process is further identified with a number that will be used for identification purpose.
- The development of DFD'S is done in several levels. Each process in lower- level diagrams can be broken down into a more detailed DFD in the next level.
- The top-level diagram is often called context diagram. The process in the context level diagram is exploded into other process at the first level DFD.
- Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical from, this lead to the modular design.
- ADFD is also known as a "Bubble Chart".

- **Symbols used in DFD:**

<u>Symbol</u>	<u>Name</u>	<u>Description</u>
	External Entity	Entities include source and destination of the data. Entities are represented by rectangle with their corresponding names. For Example, Customer, Manager, etc....
	Process	The task performed on the data is known as process. Process is represented by circle. For Example, insertion of new item, updating the existing item, etc.
	Data Flow	The movement of data in the system is known as dataflow. It is represented with the help of arrow. The tail of the arrow is source and the head of the arrow is destination.
	Data Store	Data storage includes the database of the system. It is represented by rectangle with both smaller sides missing or in other words within two parallel lines.

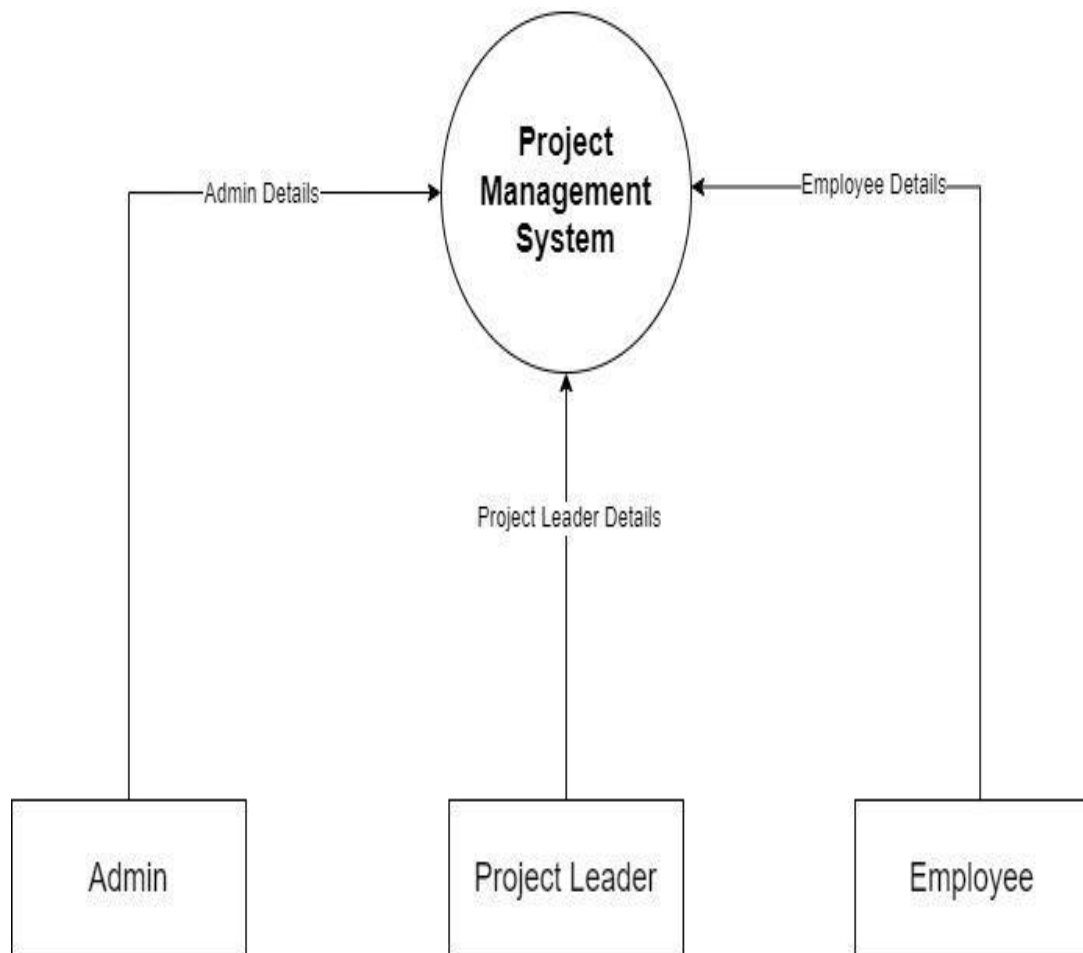
**5.4.1 Context Level DFD:**

Fig 5.4.1 Context Level DFD

### 5.4.2 First Level DFD of Admin:

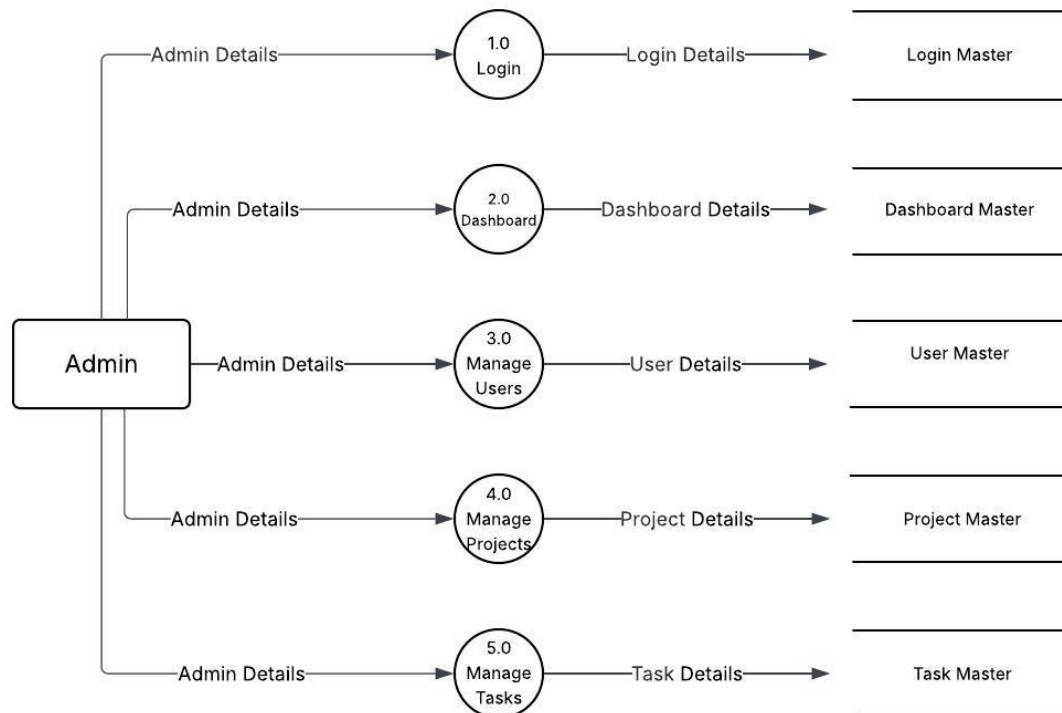


Fig 5.4.2 First Level DFD of Admin

### 5.4.3 Second Level DFD of Admin(3.0)

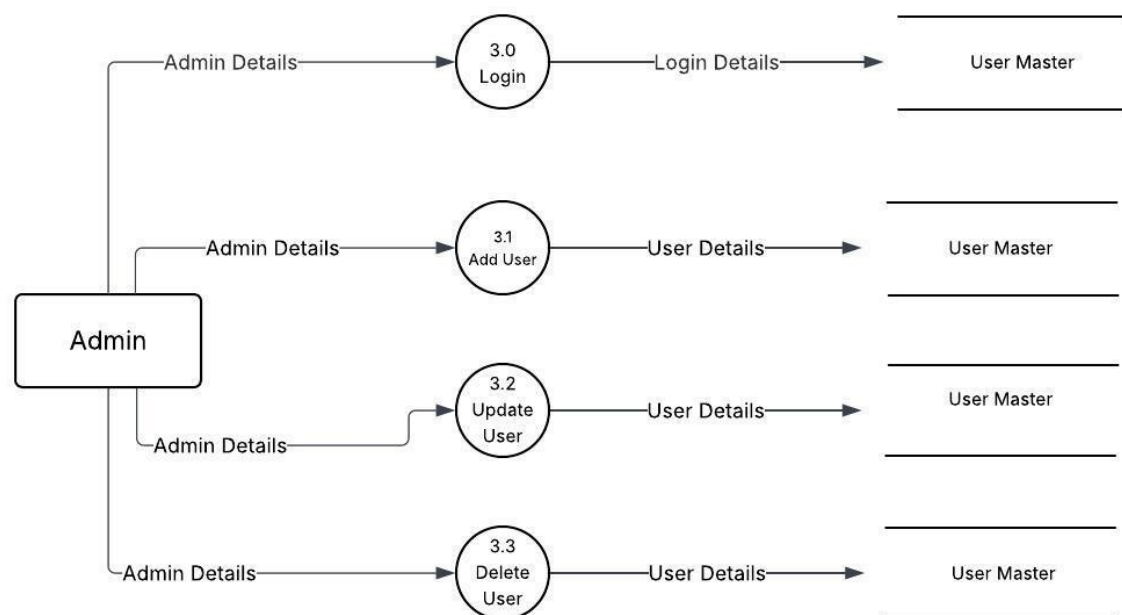


Fig 5.4.3 Second Level DFD of Admin

#### 5.4.4 Third Level DFD of Admin(4.0):

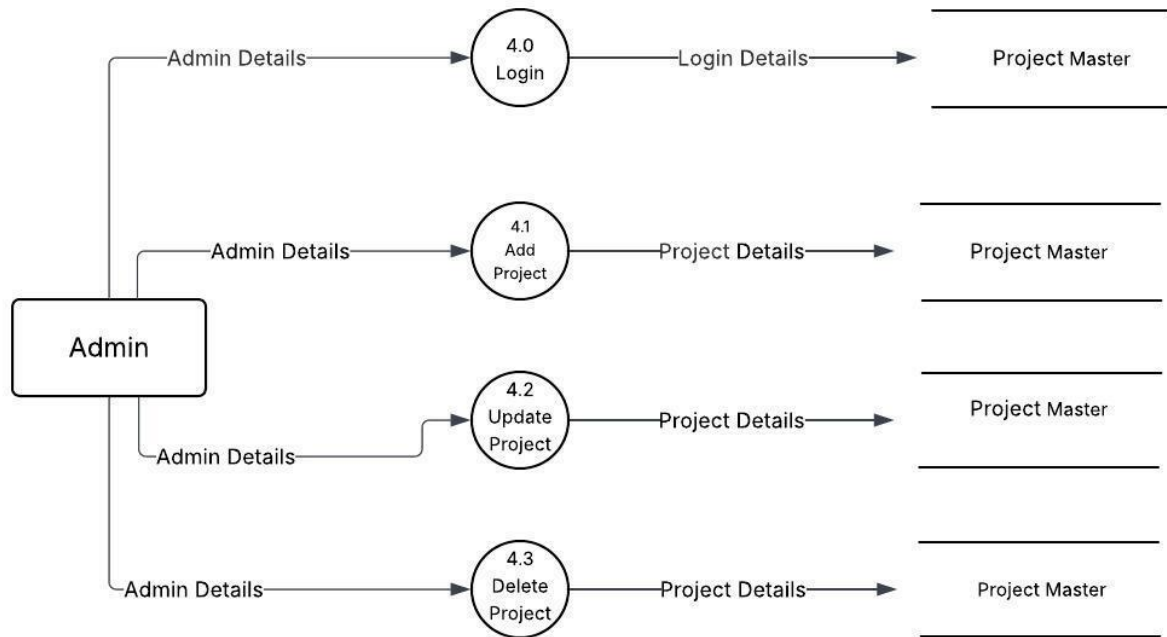


Fig 5.4.4 Third Level DFD of Admin

#### 5.4.5 Fourth Level DFD of Admin(5.0):

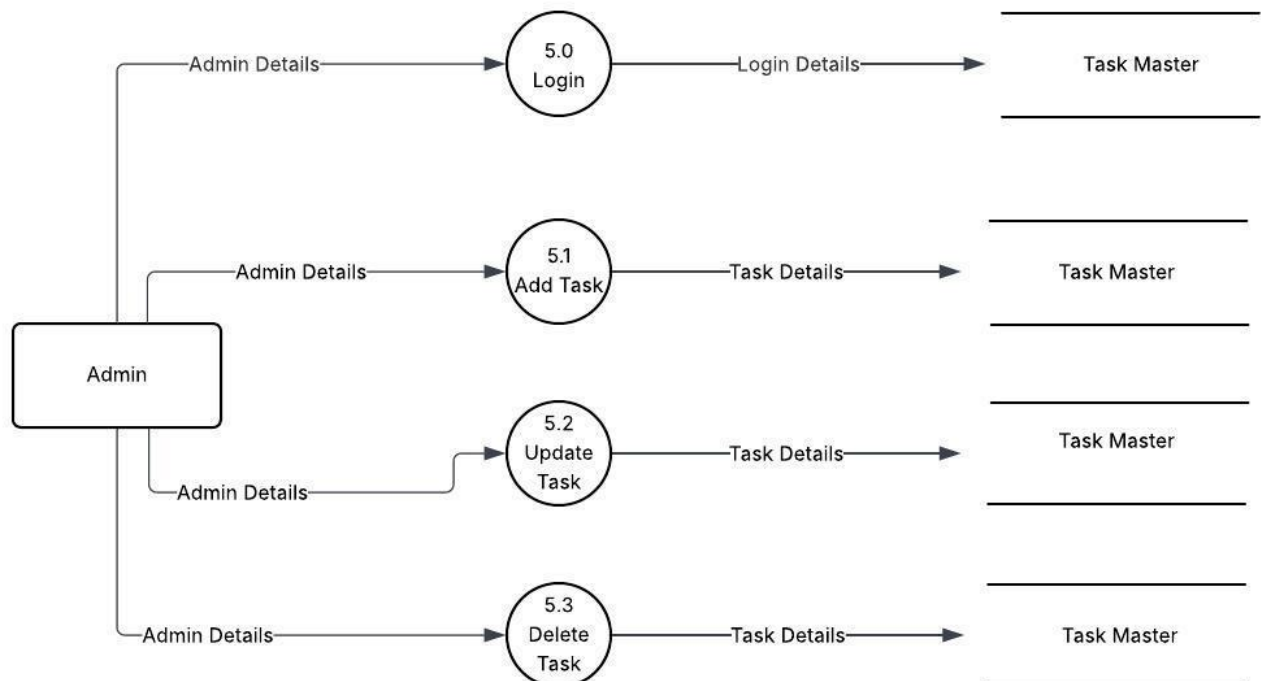


Fig 5.4.5 Fourth Level DFD of Admin



### 5.4.6 First Level DFD of Project Manager:

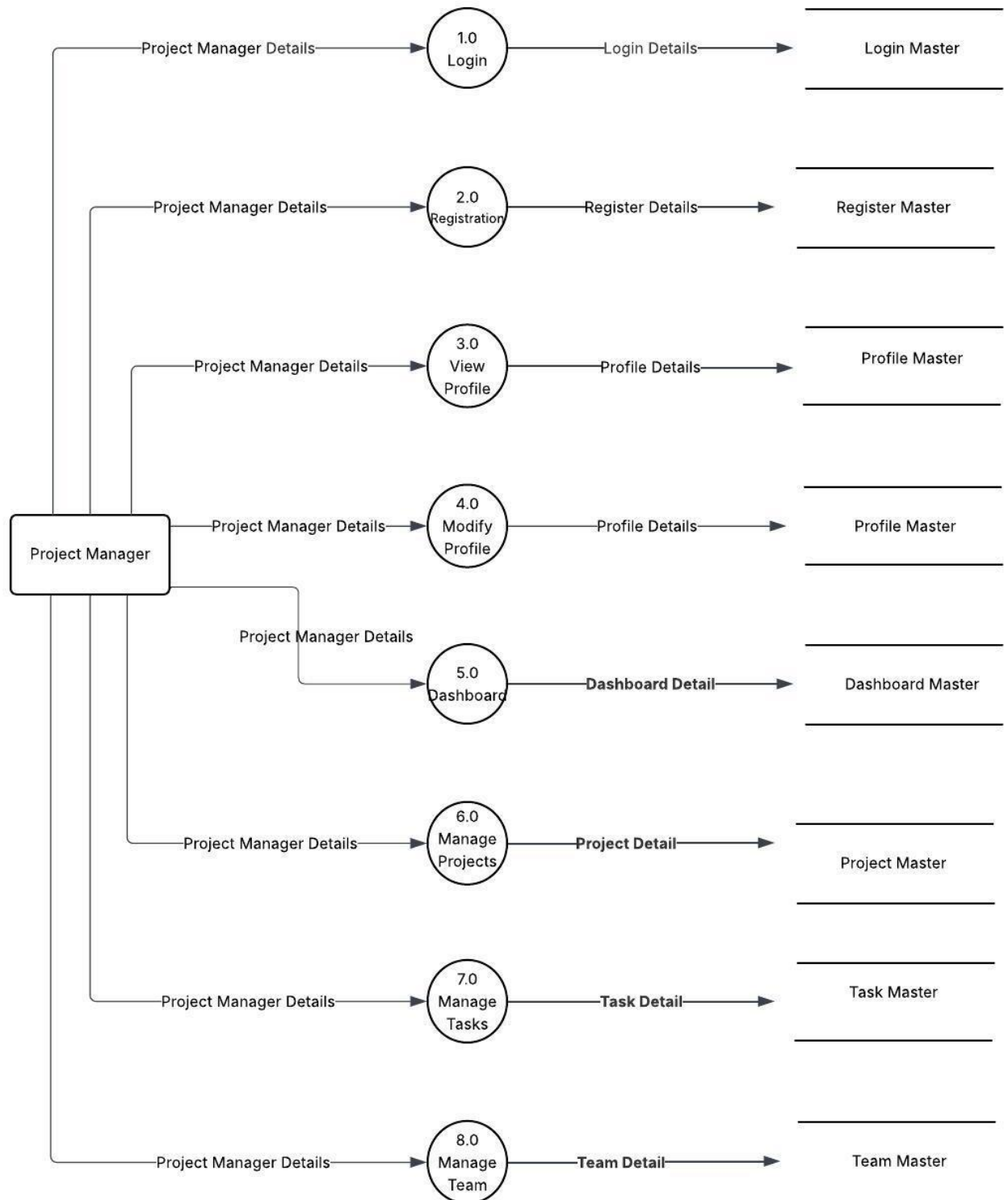


Fig 5.4.6 First Level DFD of Project Manager

#### 5.4.7 Second Level DFD of Project Manager(6.0):

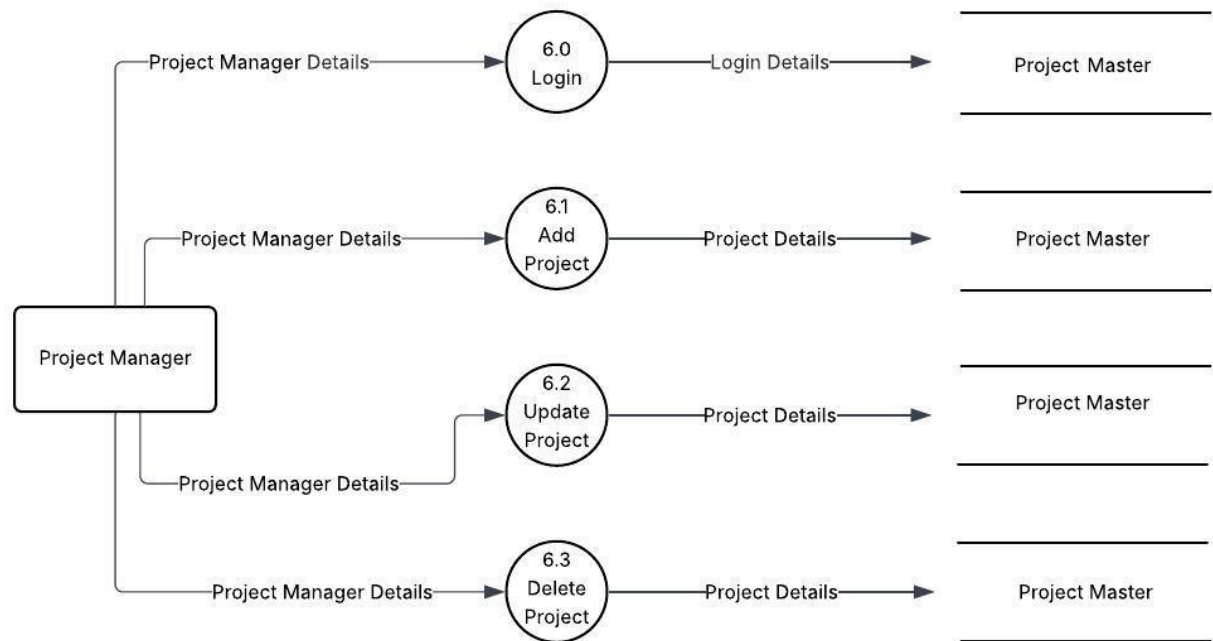


Fig 5.4.7 Second Level DFD of Project Manager

#### 5.4.8 Third Level DFD of Project Manager(7.0):

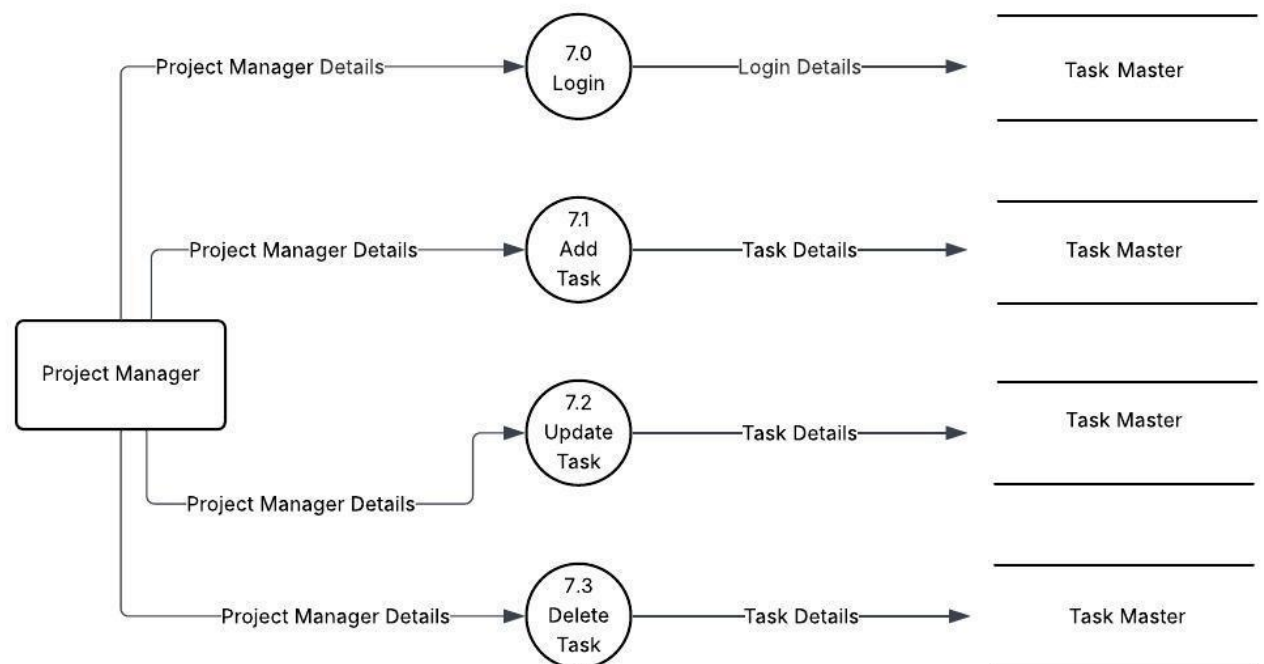


Fig 5.4.8 Second Level DFD of Project Manager

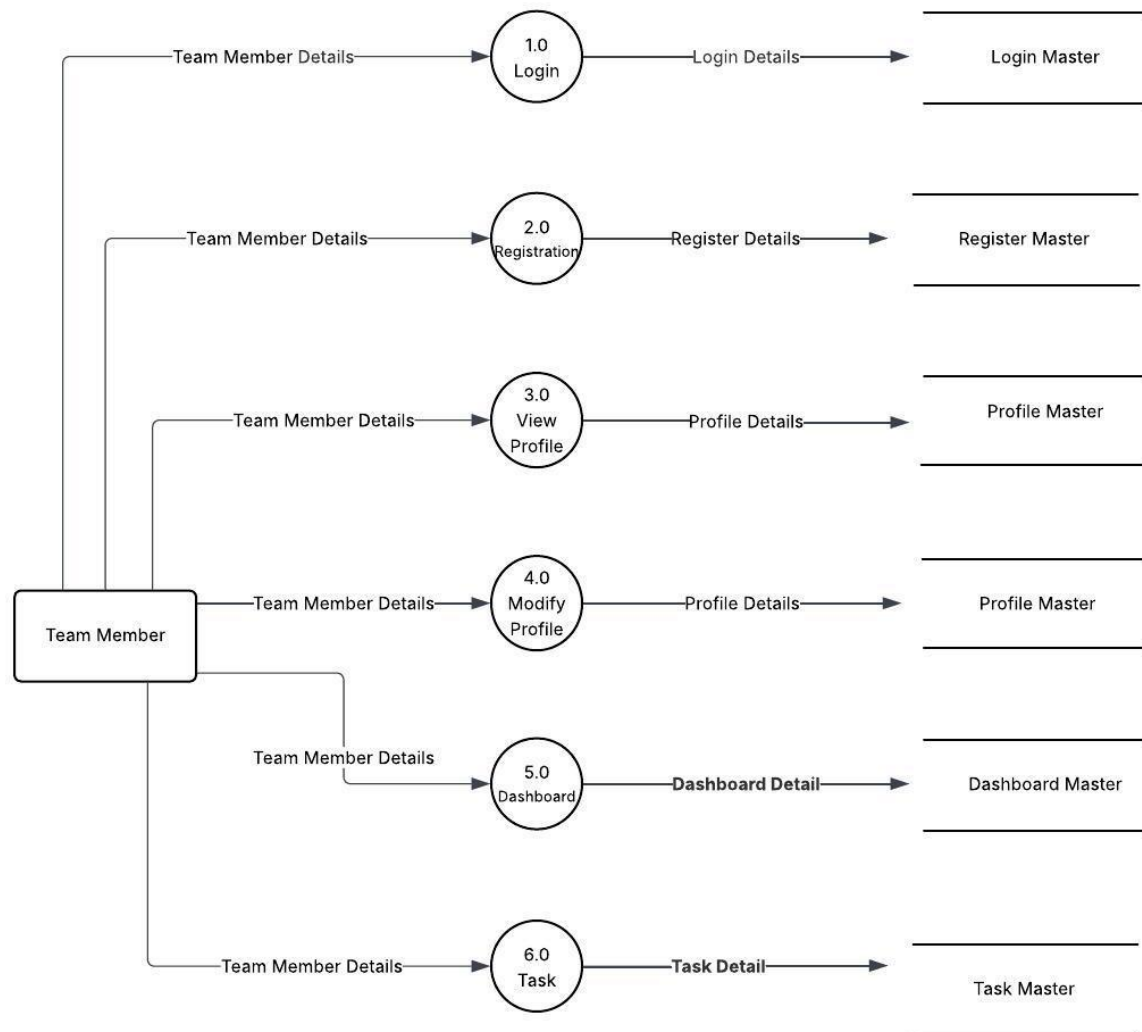
**5.4.9 First Level DFD of Team Member:**

Fig 5.4.9 First Level DFD of Team Member

## 5.5 Data Dictionary:

### 5.5.1 Users Table:

Field Name	Data Type	Constraints	Description
id	INT	Primary Key, Auto-Increment	Unique identifier for each user
name	VARCHAR(255)	NOT NULL	Full name of the user
email	VARCHAR(255)	NOT NULL, UNIQUE	User's email address
password	VARCHAR(255)	NOT NULL	Encrypted user password
role	VARCHAR(50)	NOT NULL	User's role (e.g., sponsor, influencer, admin)
phone	VARCHAR(20)	NULL	Phone number of the user
company	VARCHAR(255)	NULL	Company name (for sponsors)
address	TEXT	NULL	Physical address of the user
bio	TEXT	NULL	Short biography or profile description
profilePic	VARCHAR(255)	NULL	File path or URL to user's profile picture
subrole	VARCHAR(100)	NULL	Sub-category or specialization of the user
availability	VARCHAR(100)	NULL	Availability status (e.g., available, busy)
skills	TEXT	NULL	User's skills or areas of expertise
experienceLevel	VARCHAR(100)	NULL	Experience level (e.g., Beginner, Intermediate, Expert)
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Date and time when the account was created
updated_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	Date and time of the last update to the account

Table 5.5.1 Users Table

### 5.5.2 Projects Table:

Field Name	Data Type	Constraints	Description
Project_ID	INT	Primary Key, Auto-Increment	Unique identifier for each project
Title	VARCHAR(255)	NOT NULL	Title or name of the project
Description	TEXT	NULL	Detailed description of the project
Start_Date	DATE	NULL	Project start date
End_Date	DATE	NULL	Project end date
Budget	DECIMAL(10,2)	NULL	Budget allocated for the project
userId	INT	Foreign Key (users.id)	ID of the user who created or owns the project
Status	VARCHAR(100)	NULL	Current status (e.g., active, completed, pending)
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Date and time when the project was created
updated_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	Date and time of the last update to the project

Table 5.5.2 Project Table

### 5.5.3 Admin Table:

Column Name	Data Type	Description
id	INT (PK, AI)	Unique admin ID (Primary Key, Auto Increment)
email	VARCHAR(100)	Admin email address
password	VARCHAR(255)	Encrypted password
role	VARCHAR(50)	Role (default: "Admin")
created_at	DATETIME	When the admin account was created

Table 5.5.3 Admin Table

**5.5.4 Task Table:**

Field Name	Data Type	Constraints	Description
id	INT	Primary Key, Auto-Increment	Unique identifier for each task
title	VARCHAR(255)	NOT NULL	Title or name of the task
description	TEXT	NULL	Detailed description of the task
status	VARCHAR(50)	NOT NULL	Current status (e.g., pending, in progress, completed)
deadline	DATE	NULL	Task deadline date
assigned_to	INT	Foreign Key (users.id)	User ID to whom the task is assigned
priority	VARCHAR(50)	NULL	Task priority level (e.g., low, medium, high)
project_name	VARCHAR(255)	NULL	Name of the related project
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Date and time when the task was created
updated_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	Date and time of the last update to the task

Table 5.5.4 Task Table

**5.5.5 Team Members Table:**

Column Name	Data Type	Description
id	INT (PK, AI)	Unique ID for the association entry
project_id	INT	Foreign Key referencing the associated project
member_id	INT	Foreign Key referencing the assigned team member
assigned_at	DATETIME	Timestamp when the member was assigned to project

Table 5.5.5 Team Members Table

**5.5.6 Events Table:**

Field Name	Data Type	Constraints	Description
id	INT	Primary Key, Auto-Increment	Unique identifier for each event
title	VARCHAR(255)	NOT NULL	Title or name of the event
description	TEXT	NULL	Detailed description of the event
event_date	DATE	NOT NULL	Scheduled date of the event
type	VARCHAR(100)	NULL	Type of event (e.g., meeting, deadline, reminder)
creator_id	INT	Foreign Key (users.id)	ID of the user who created the event
project_id	INT	Foreign Key (projects.Project_ID)	ID of the project associated with the event
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Date and time when the event was created

Table 5.5.6 Events Table

**5.5.7 Comments Table:**

Field Name	Data Type	Constraints	Description
id	INT	Primary Key, Auto-Increment	Unique identifier for each comment
project_id	INT	Foreign Key (projects.Project_ID), NOT NULL	ID of the project the comment is associated with
user_id	INT	Foreign Key (users.id), NOT NULL	ID of the user who posted the comment
content	TEXT	NOT NULL	The actual content or message of the comment
timestamp	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Date and time when the comment was posted

Table 5.5.7 Comments Table

**5.5.8 Files Table:**

Field Name	Data Type	Constraints	Description
id	INT	Primary Key, Auto-Increment	Unique identifier for each file
name	VARCHAR(255)	NULL	Name of the uploaded file
type	VARCHAR(100)	NULL	MIME type or file extension (e.g., image/png, pdf)
size	BIGINT(20)	NULL	Size of the file in bytes
data	LONGBLOB	NULL	Binary data of the file
uploadedBy	INT	Foreign Key (users.id), NULL	ID of the user who uploaded the file
projectId	INT	Foreign Key (projects.Project_ID), NOT NULL	ID of the associated project
timestamp	DATETIME	NULL	Date and time when the file was uploaded

Table 5.5.8 Files Table



## 6. IMPLEMENTATION

### 6.1 HOME PAGE:

The Home Page serves as the initial entry point for users. It introduces the platform and offers easy navigation to the Login Page, Admin Page, and additional platform features. The page provides an overview of the system's functionalities, ensuring that users can quickly access the services they need.

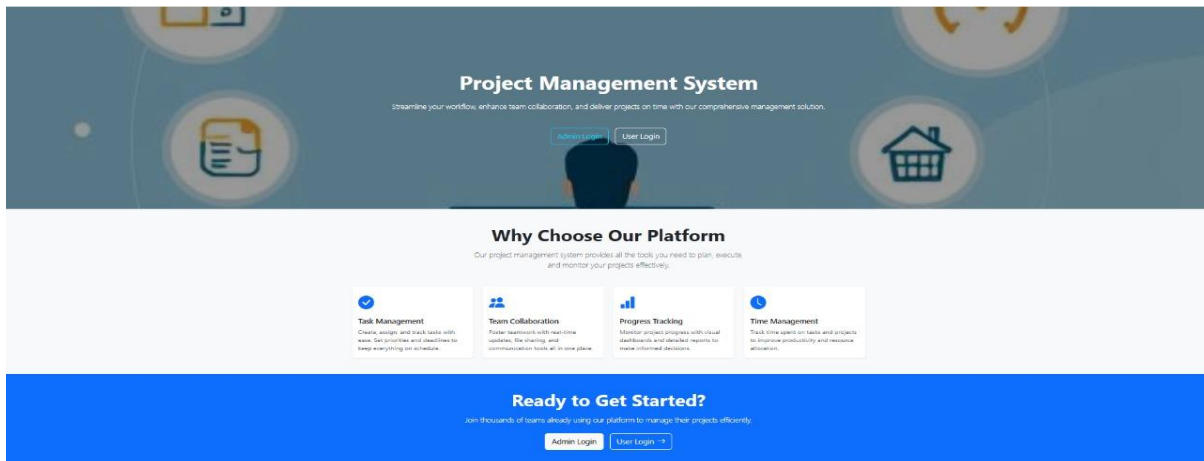


Fig 6.1 Home Page

### 6.2 USER LOGIN PAGE:

The Login Page allows users to access the system by entering their credentials. Users will enter their registered email and password to log in. If the login credentials are correct, they will be redirected to their respective dashboards based on their roles (Admin, Project Manager, or Team Member). This page has a clean, user-friendly design with clear fields for email and password, ensuring an easy and secure login experience.

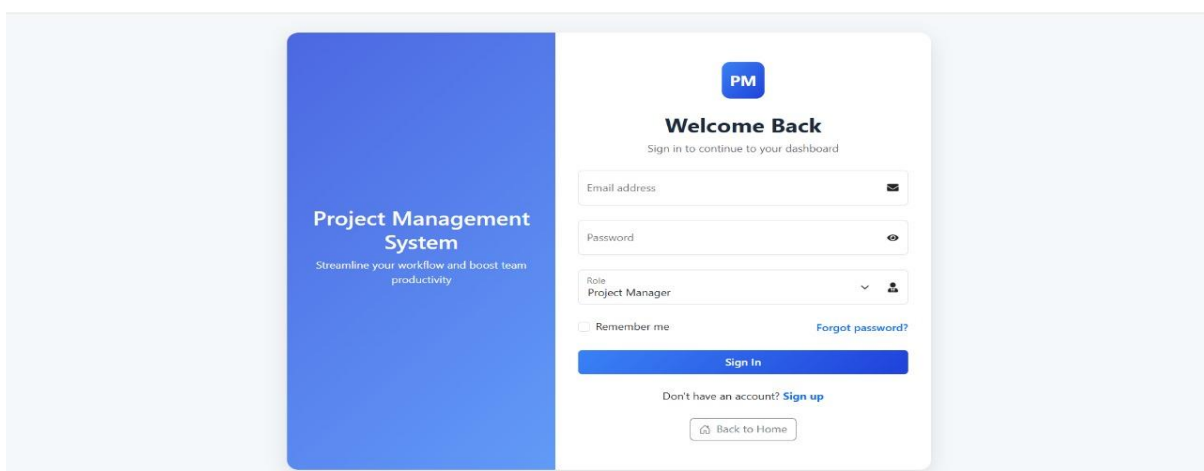


Fig 6.2 User Login Page

### 6.3 USER REGISTER PAGE:

The Registration Page allows new users to create an account. Users will be asked to provide basic information such as name, email, password, and role ( Project Manager, or Team Member). Once the registration form is completed, users are redirected to a profile page where they can enter more detailed information (e.g., job title, team role). This process ensures that each user has a complete profile upon registration.

Fig 6.3 User Register Page

### 6.4 PROJECT MANAGER DASHBOARD PAGE:

The Project Manager Dashboard serves as the central hub for Project Managers. On this page, Project Managers can create, edit, or delete projects, assign tasks to team members, and monitor the progress of their projects. The dashboard provides real-time updates on the status of various projects, with easy-to-use navigation for managing multiple projects at once.

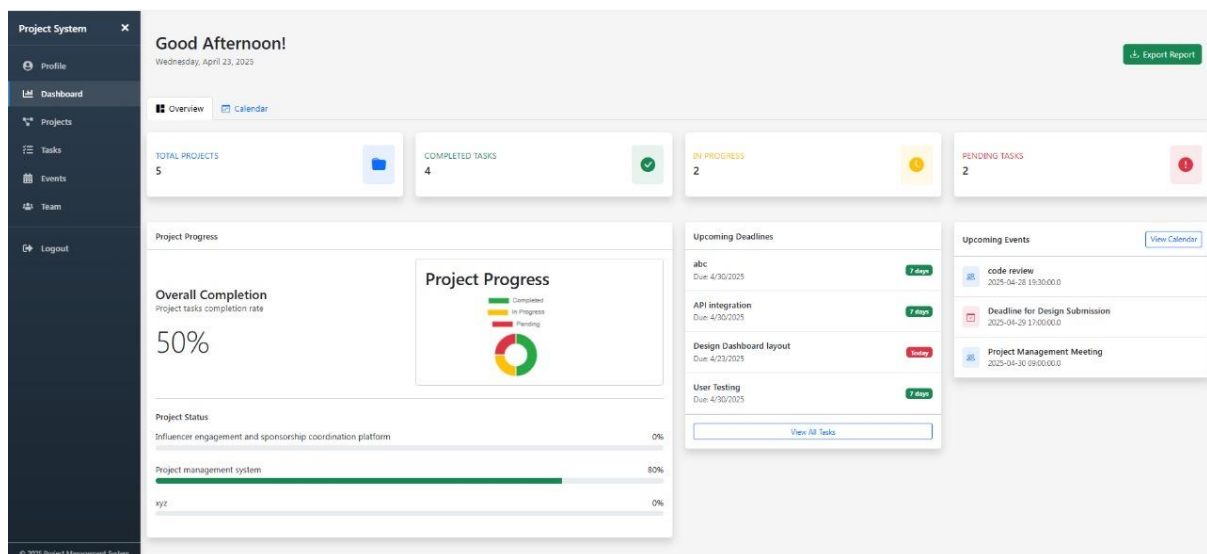


Fig 6.4 Project Manager Dashboard Page

6.5 PROJECT MANAGER PROFILE PAGE:

The Profile Page allows users to view and update their personal information, including username, email, role, and project preferences. Users can also edit their contact information and add a brief bio or description about their professional background. This page ensures that each user’s profile is accurate and up to date.

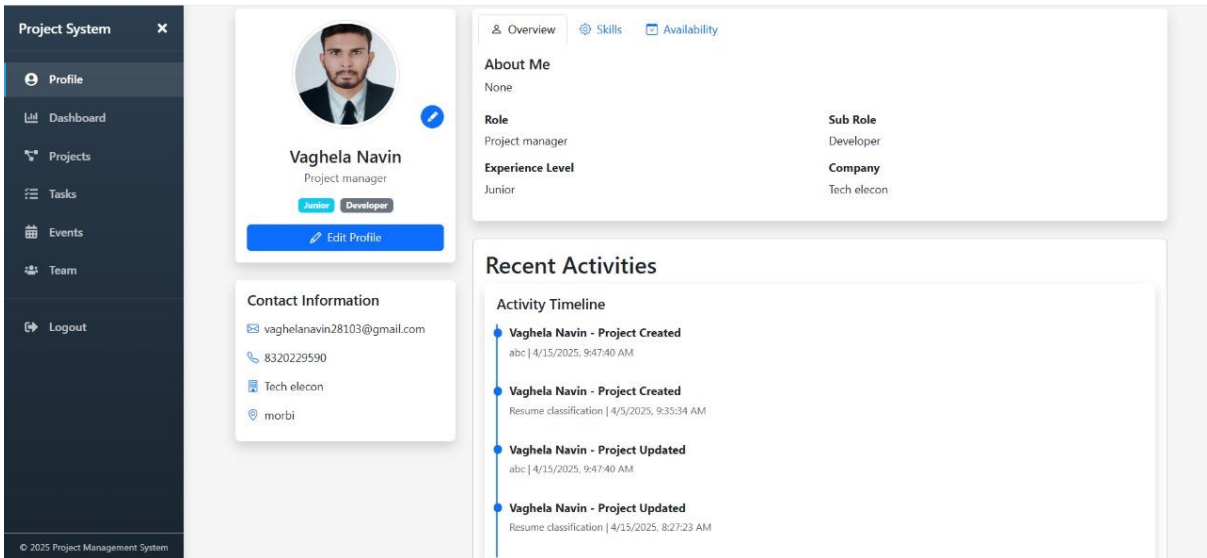


Fig 6.5 Project Manager Profile Page

6.6 PROJECT MANAGER EDIT PROFILE PAGE:

Edit Profile

Profile Picture

Choose FileNo file chosen

Name

Moksha Bodana

Email

moksha1@gmail.com

Phone

Company

Address

Bio

Role

Project Manager

Sub Role

Developer

Save Changes

Fig 6.6 Project Manager Edit Profile Page

## 6.7 PROJECT MANAGER PROJECT LIST PAGE:

The Project List Page displays a comprehensive overview of all projects currently being managed. Project Managers can search, filter, and sort projects based on various criteria such as project status, priority, or deadline. Each project in the list includes relevant details such as its name, status, and assigned team members, enabling Project Managers to easily track progress.

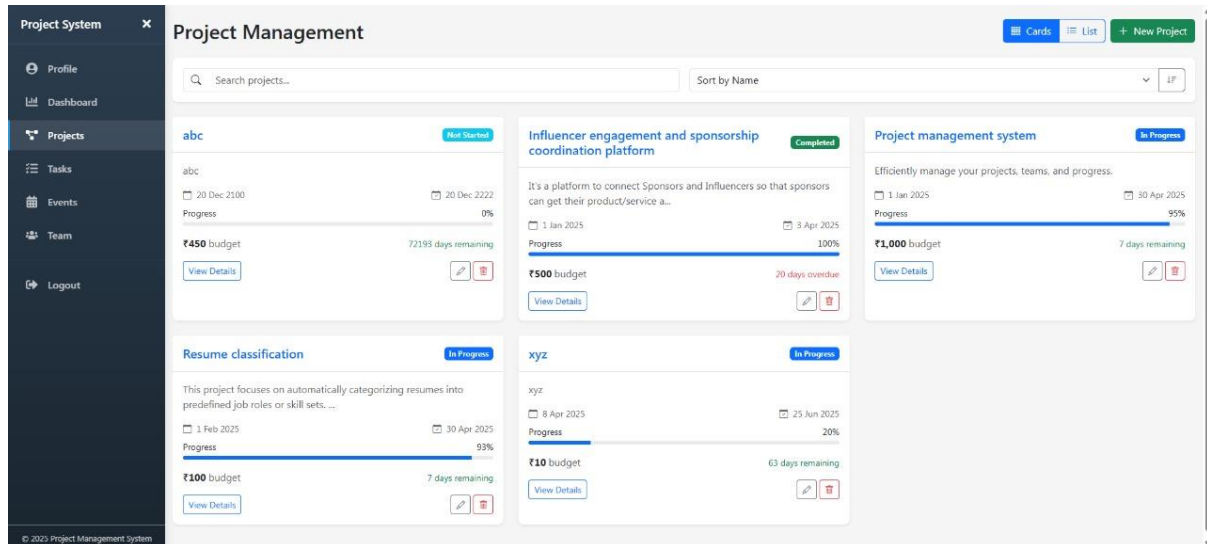


Fig 6.7 Project Manager Project List Page

## 6.8 PROJECT MANAGER EDIT/ADD PROJECT PAGE:

The Project Form Page enables Project Managers to create new projects. Project Managers will fill out a form detailing the project name, description, timeline, team members, and specific goals. The form includes clear labels and instructions, guiding users through the process of entering the project details. Once completed, users can submit the form to create the project.

### Edit Project

Project Name

Description

Start Date

End Date

Budget

### Create New Project

Project Name:

Description:

Start Date:

End Date:

Budget:

Fig 6.8 Project Manager Edit/Add Project Page

## 6.9 PROJECT MANAGER TASKS LIST PAGE:

The Task List Page provides an overview of all tasks within a project, displaying important details like deadlines, assigned team members, and current status. Project Managers and Team Members can easily filter and search tasks based on various parameters, ensuring quick access to critical tasks. This page is designed to facilitate efficient task management and status tracking.

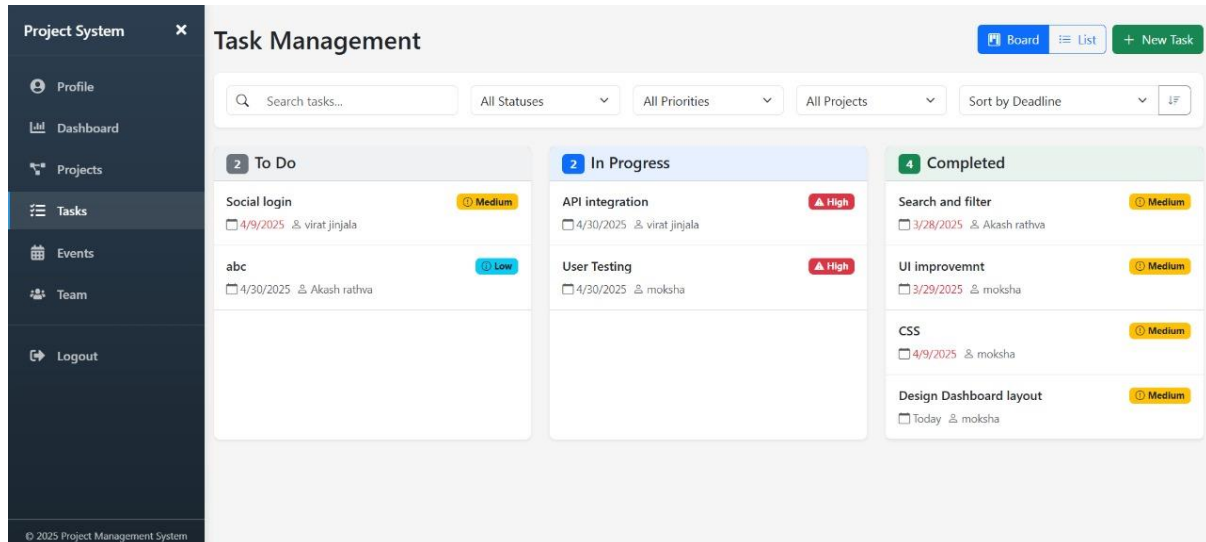


Fig 6.9 Project Manager Tasks List Page

## 6.10 PROJECT MANAGER EDIT/ADD TASK PAGE:

The Task Form Page allows Project Managers to create tasks associated with a project. Here, they can assign specific tasks to Team Members, set deadlines, and provide task descriptions. The page also includes options for adding task priorities and status (e.g., In Progress, Completed, Pending). Clear labels and instructions are provided to guide users through the process.

Fig 6.10 Project Manager Edit/Add Task Page

6.11 PROJECT MANAGER EVENTS PAGE:

The Event Creation form allows project managers or team members to schedule new events related to their projects. Users can input the event title, description, date and time, and select the type of event (e.g., Meeting). They must also choose the associated project from a dropdown list. Once all details are filled in, the event can be created with a single click. This feature helps streamline team coordination and ensures all important events are tracked within the project system.

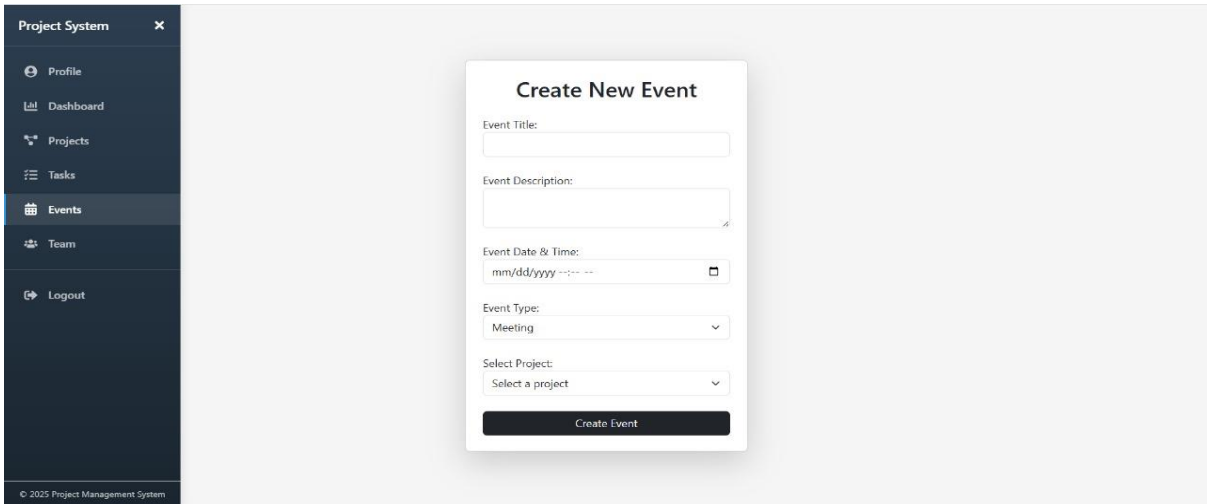


Fig 6.11 Project Manager Event Page

6.12 PROJECT MANAGER TEAM MEMBER PAGE:

The Team Members Page allows Project Managers to view all team members associated with a particular project. The page displays the name, role, and assigned tasks for each team member, helping Project Managers monitor team participation and performance. Additionally, Project Managers can add or remove team members from the project.

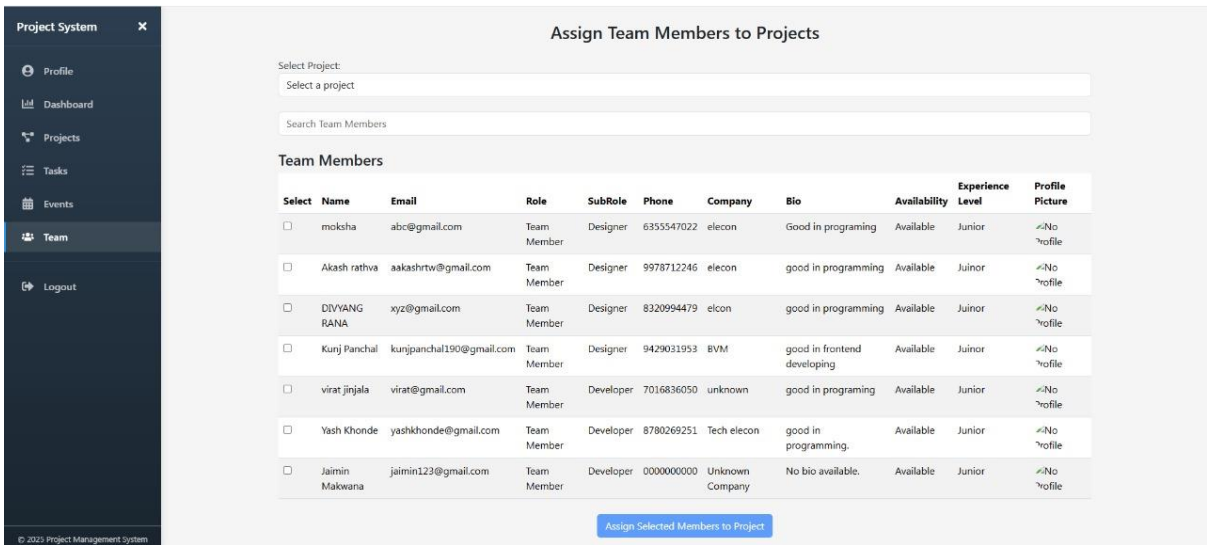


Fig 6.12 Project Manager Team Member Page

### 6.13 TEAM MEMBER DASHBOARD PAGE:

The Team Member's Dashboard displays a summary of all tasks assigned to the user. This dashboard is designed to help Team Members track their ongoing work, upcoming tasks, and deadlines. It includes a list of active projects, along with the ability to filter and prioritize tasks based on their importance or due date. The dashboard promotes effective task management and allows Team Members to stay organized.

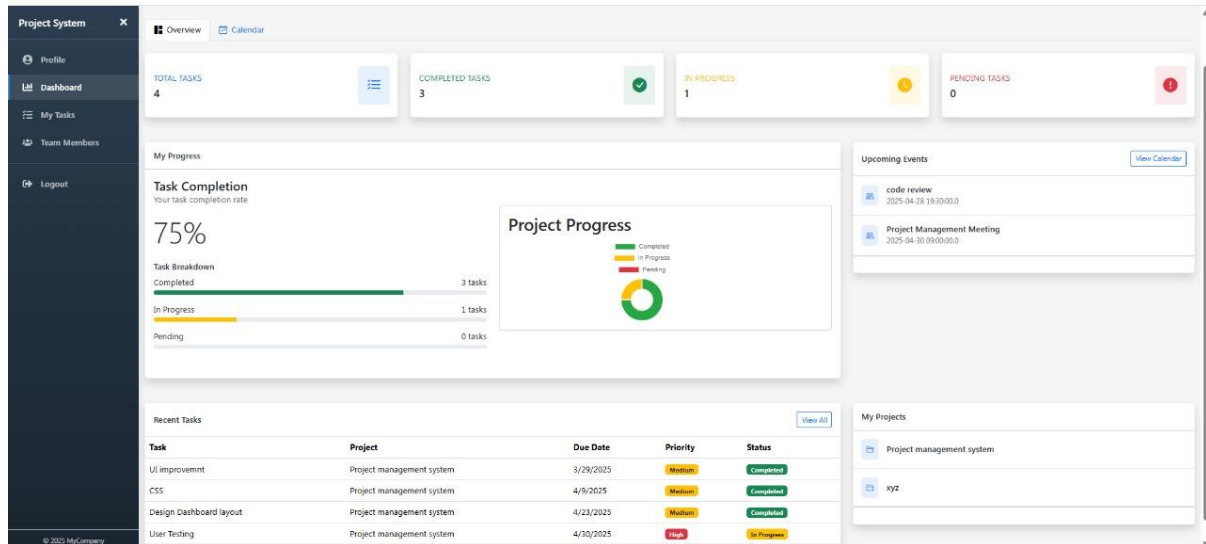


Fig 6.13 Team Member Dashboard

### 6.14 TEAM MEMBER PROFILE PAGE:

The Team Member's Profile Page is where each Team Member can manage their personal information, such as name, email, and role within the project. Team Members can update their contact details and add a short biography. This page serves as a central location for Team Members to keep their information current and relevant to the project.

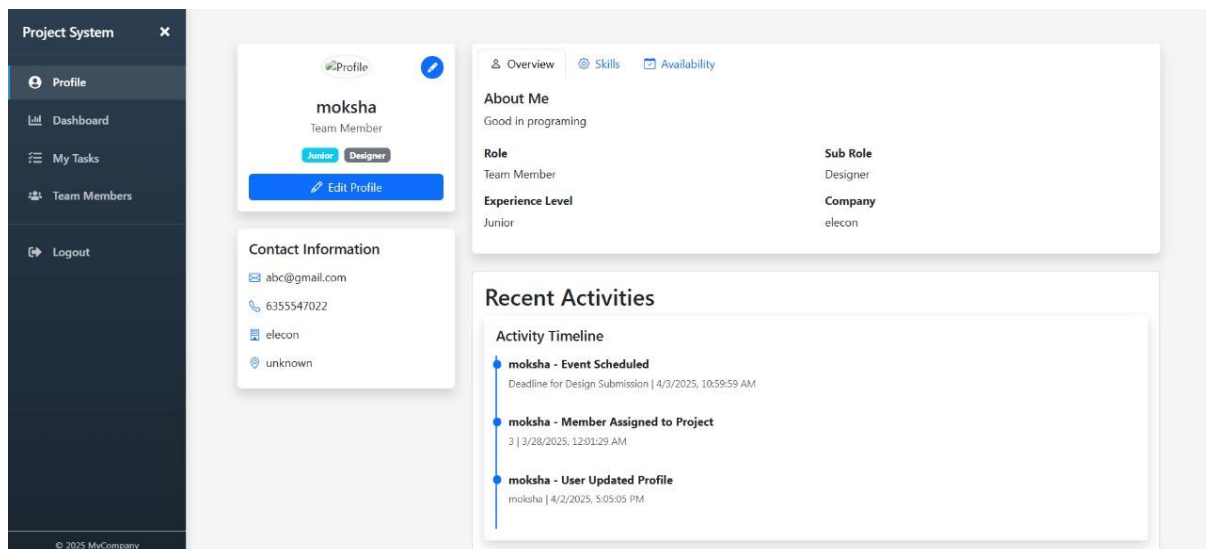


Fig 6.14 Team Member Profile Page



## 6.15 TEAM MEMBER TASKS PAGE:

The Team Member's Task List provides an overview of the tasks assigned to a specific Team Member. This page allows users to view task details, including deadlines, task descriptions, and priorities. Team Members can update their task status (e.g., In Progress, Completed, Pending) directly from this page, keeping everyone updated on their work progress.

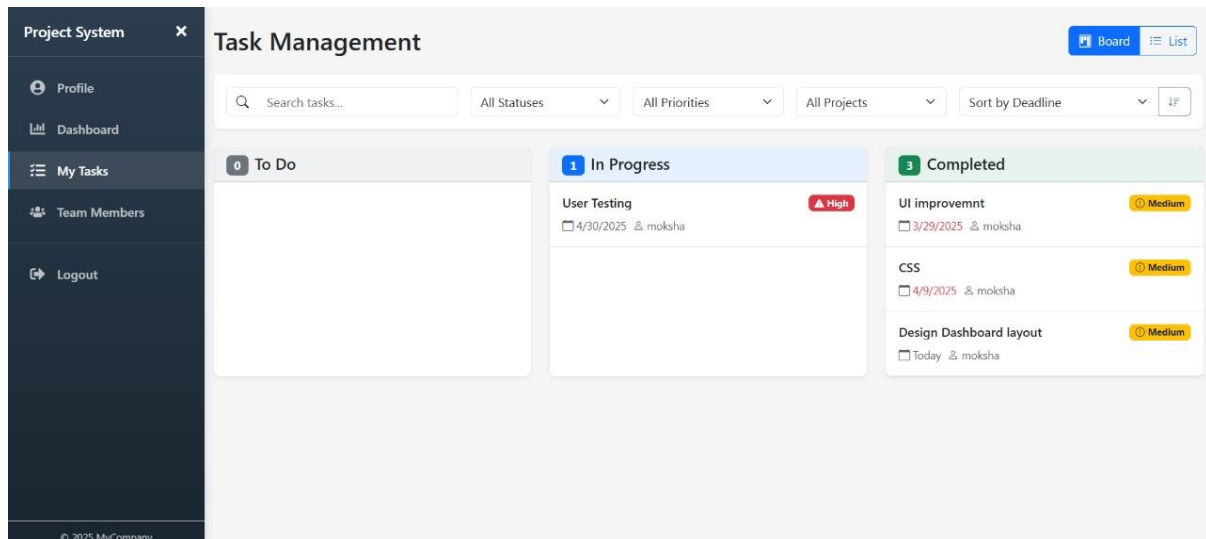


Fig 6.15 Team Member Tasks Page

## 6.16 TEAM MEMBER PAGE:

The Team Member's Team Page shows a list of all team members involved in the same project. This page includes relevant information about each team member, such as their role, tasks, and progress. The Team Member's Team page enables collaboration and communication by giving an overview of who is working on what within the project.

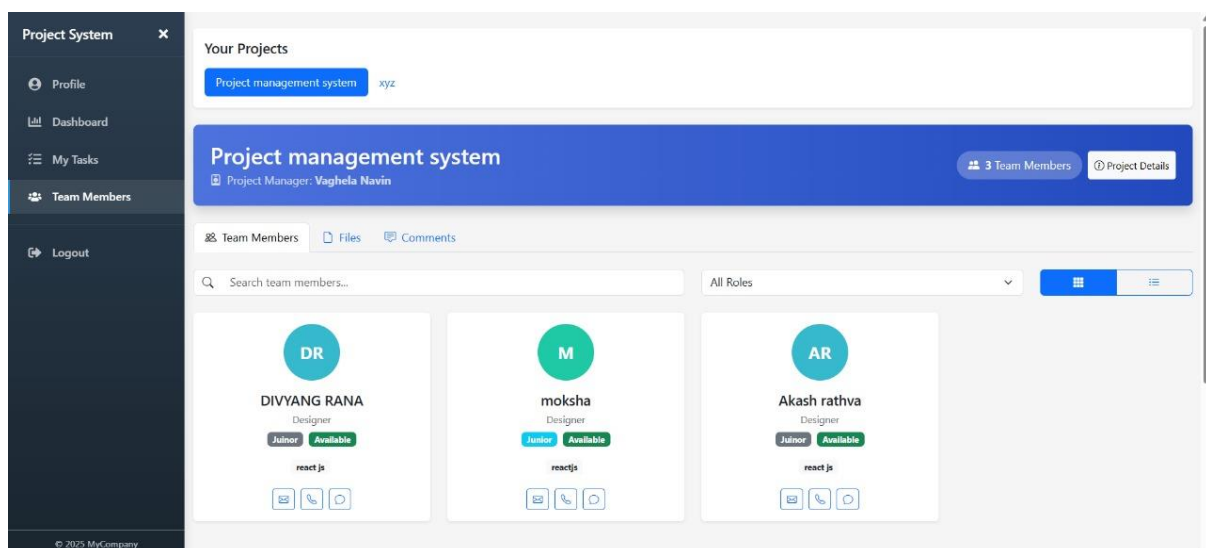


Fig 6.16 Team Member Page



### 6.17 ADMIN LOGIN PAGE:

The Admin Login Page is specifically designed for administrators to access the backend of the platform. The page includes a login form where admins can enter their credentials (email and password). Once logged in, admins gain access to powerful tools to manage user accounts, monitor system performance, and review analytics. The design of this page is simple and secure, focusing on easy access to admin-specific features.

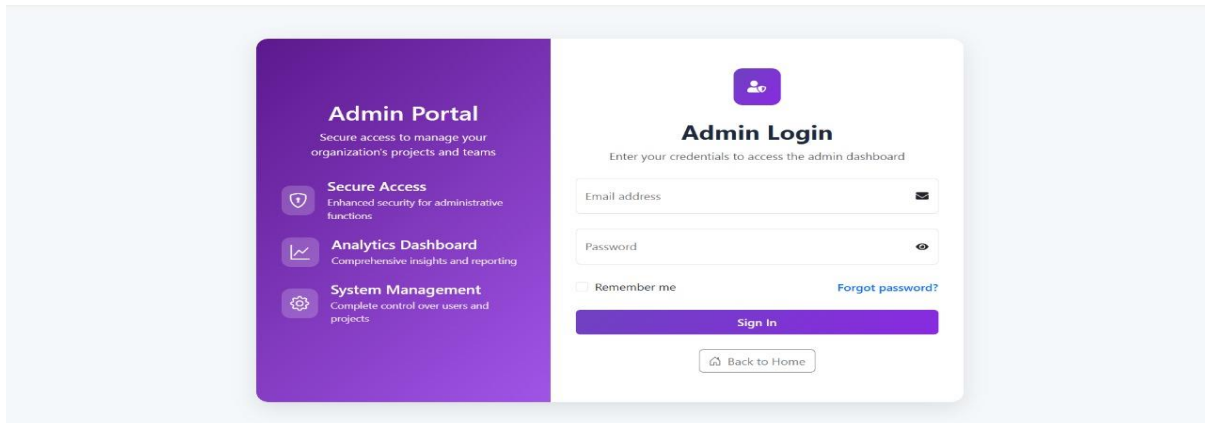


Fig 6.17 Admin Login Page

### 6.18 ADMIN DASHBOARD PAGE:

The Admin Dashboard of the Project Management System provides a quick and clear overview of the platform's key metrics. It displays total users, projects, tasks, and the task completion rate in visually distinct cards. A line chart shows monthly trends of tasks created and completed, helping track progress over time. A pie chart presents user distribution between project managers and team members. The dashboard also includes a sidebar for easy navigation to user, project, and task management sections.

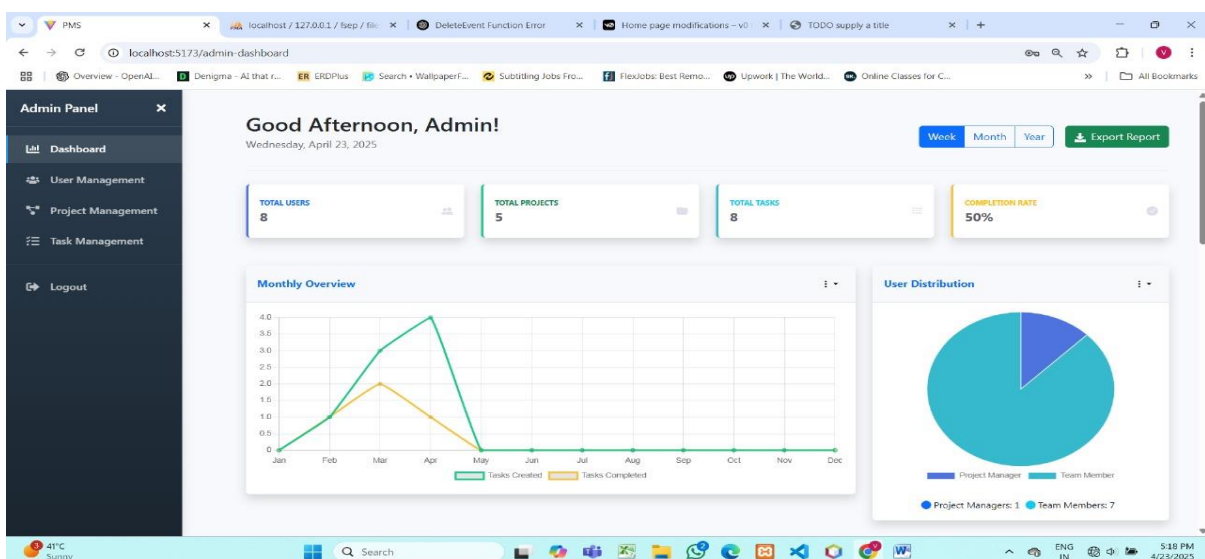


Fig 6.18 Admin Dashboard Page

## 6.19 ADMIN USER MANAGEMENT PAGE:

The User Management page allows the admin to oversee all registered users in the system. It includes a search bar and role filter for easy navigation. A pie chart displays the distribution of users by role— Project Manager and Team Member. The statistics panel shows the total number of users, available users, total assigned projects, and average projects per user. Below, a table lists each user's details, including name, email, role, department, project count, status, and options to view, edit, or delete user profiles. This page helps the admin manage user roles and project involvement efficiently.

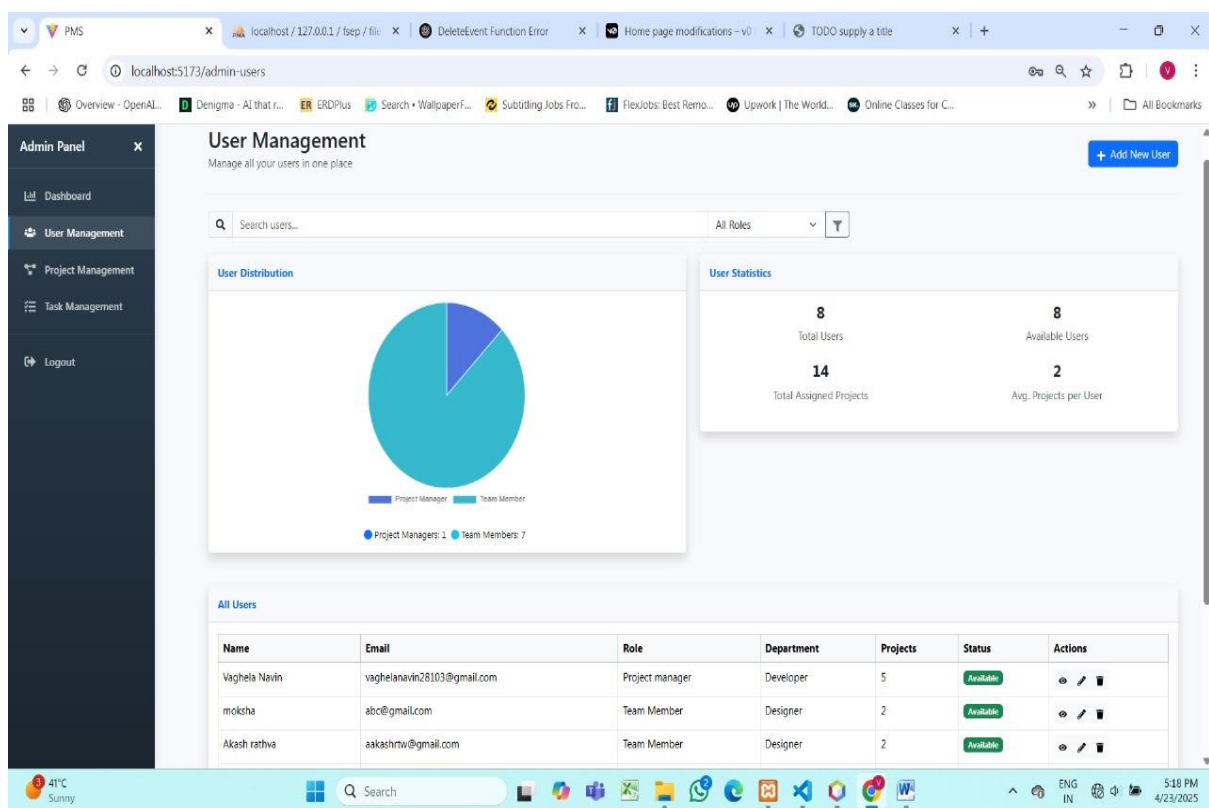


Fig 6.19 Admin User Management Page

## 6.20 ADMIN PROJECT MANAGEMENT PAGE:

The Project Management page in the admin panel of the Project Management System allows administrators to efficiently oversee all ongoing projects. It provides a search bar and status filter to quickly locate projects. A bar chart shows the number of projects created each month, while a pie chart visualizes the distribution of project statuses such as "In Progress," "Completed," and "To Do." Below, a detailed table lists all projects with information like project name, manager, team size, status, deadline, and quick action buttons for view, edit, and delete. This page helps the admin track progress and manage project workflows effectively.

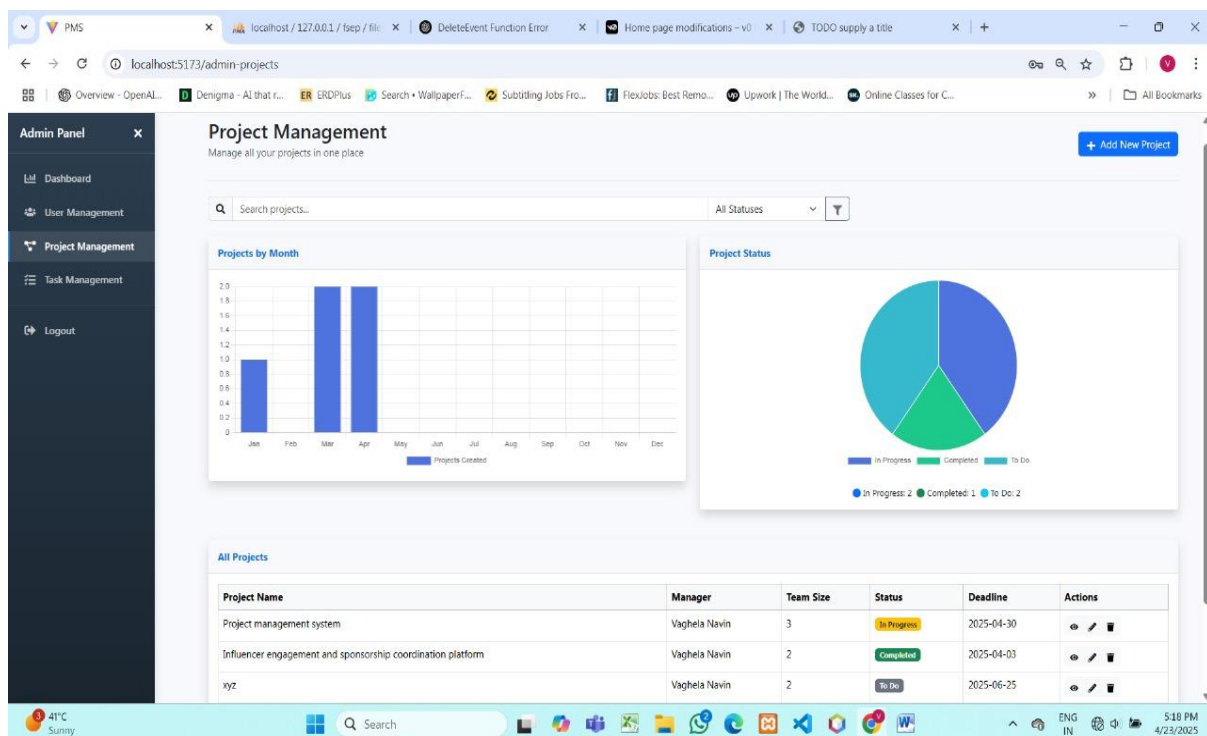


Fig 6.20 Admin Project Management Page

## 6.21 ADMIN TASK MANAGEMENT PAGE:

The Task Management page in the admin panel helps manage and track all tasks within the system. It features a search bar along with filters for status and priority, making it easy to find specific tasks. A pie chart shows the current distribution of task statuses—Completed, In Progress, and To Do. A line graph displays the trend of tasks created versus completed over the months. Below, a task table lists each task with its title, associated project, assignee, status, priority, deadline, and action buttons for viewing, editing, or deleting tasks. This page provides a clear overview of task progress and helps streamline task tracking.

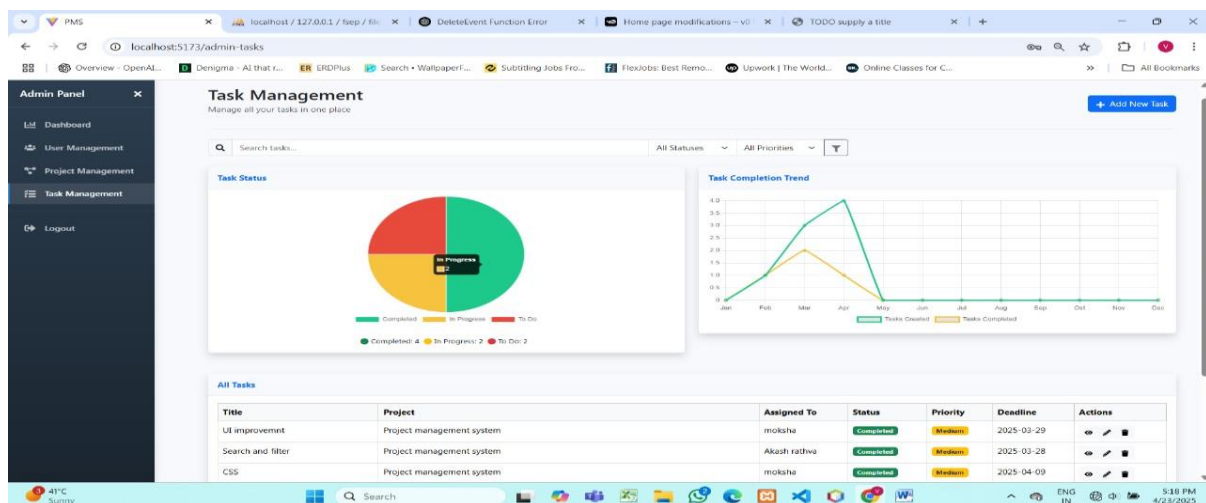


Fig 6.21 Admin Task Management Page

## 7. TESTING

### 7.1 SYSTEM TEST:

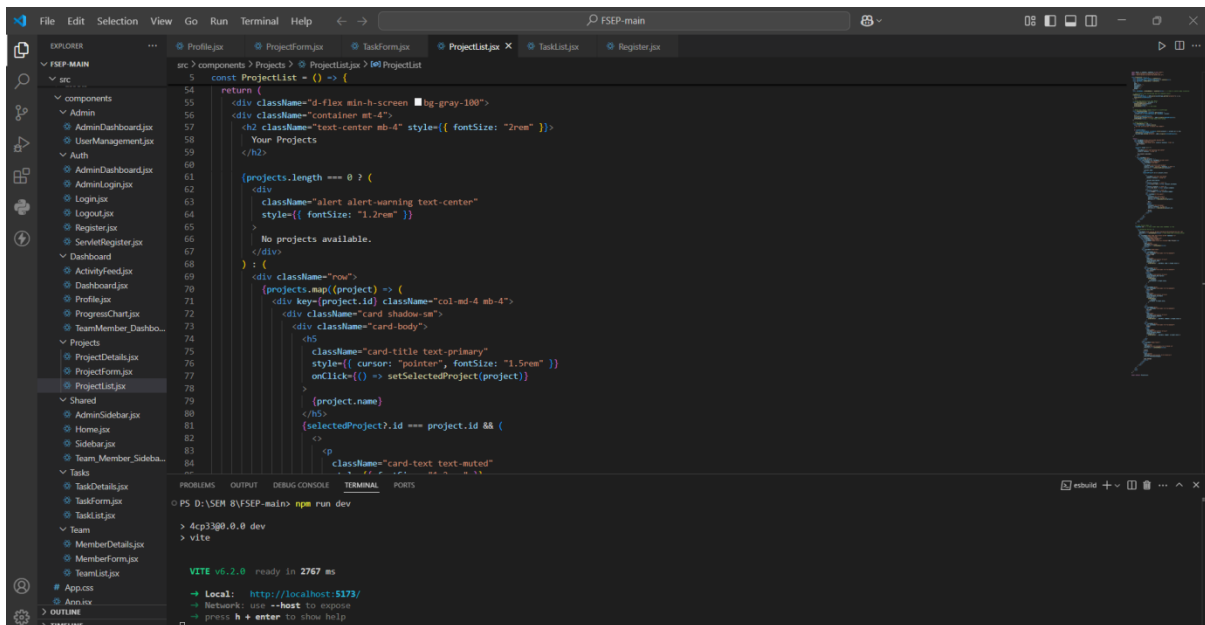


Fig 7.1 System Test

## 8. CONCLUSION

Developing a **Project Management System** as a **single-page web application** using **React.js for the frontend** and **Java with MySQL for the backend** provides multiple benefits. This system is designed to **streamline project planning, task allocation and team collaboration** efficiently. Here's a breakdown of why this technology stack is ideal for our **Project Management System**, focusing on **task management, real-time collaboration, and system scalability**.

### 1. Fast and Responsive Navigation

Our system is built using React.js, ensuring that users can switch between different sections, such as dashboard, projects, tasks, and team members, without experiencing page reloads. This single-page application (SPA) approach enhances user experience by making navigation smooth, reducing load times, and providing a more interactive interface.

### 2. Seamless User Experience

With real-time updates, project managers and team members receive instant notifications about task assignments, deadlines, and project progress. This ensures that all users stay informed, reducing delays and miscommunication. The system also allows easy task tracking, providing visibility into project status and improving collaboration between team members.

### 3. Reusable and Scalable Components

The system is built using a component-based architecture in React.js, meaning UI elements like task lists, project cards, and team member dashboards are reusable across different sections. This modular approach ensures consistency, easier maintenance, and future scalability. Any improvements or updates to components automatically reflect throughout the system, ensuring a uniform user experience.

#### **4. Easy Maintenance and Updates**

Using Java with MySQL for backend development ensures secure and efficient data management. Each system feature is built as an independent module, making it easy to update functionalities without affecting the entire application. Whether it's adding a new project, modifying task priorities, or enhancing security, updates are seamless, ensuring that the system remains reliable and future-proof.

#### **5. Final Thoughts**

Our Project Management System simplifies project tracking, enhances team collaboration, and ensures timely task completion through a structured workflow. By leveraging modern technologies, it provides a secure, user-friendly, and scalable platform for organizations to manage their projects efficiently. Whether you're a Project Manager organizing tasks or a Team Member completing assignments, this system helps keep everything organized and accessible.

## 9. REFERENCES

### 9.1 Technologies and Documentation Links:

- **React.js:**<https://react.dev/>
- **HTML (Markup Language):**<https://www.w3schools.com/html/>
- **CSS:**<https://www.w3schools.com/css/>
- **Tailwind CSS:**<https://tailwindcss.com/docs/installation>
- **Material UI:**<https://mui.com/material-ui/>
- **React Router DOM:**<https://reacttraining.com/react-router>
- **React Hooks:**<https://legacy.reactjs.org/docs/hooks-intro.html>
- **JavaScript:**<https://www.javascript.com/>
- **GitHub:**<https://github.com/index>
- **Font Awesome:**<https://fontawesome.com/>
- **Draft.js:**<https://draftjs.org/>
- **React-Recharts:** <https://recharts.org/>
- **React Modal:**<https://github.com/reactjs/react-modal>