Project Management System

A PROJECT REPORT

Submitted by

Aakash S Rathva (22CP326)

In partial fulfillment for the award of the degree of

B. TECH. in COMPUTER ENGINEERING

4CP33: Full Semester External Project (FSEP)

BIRLA VISHVAKARMA MAHAVIDYALAYA
(ENGINEERING COLLEGE)
(An Autonomous Institution)
VALLABH VIDYANAGAR

Affiliated to

GUJARAT TECHNOLOGICAL UNIVERSITY, AHMEDABAD

Academic Year: 2024 – 2025

BVM ENGINEERING COLLEGE, VALLABH VIDYANAGAR-388120

APPROVAL SHEET

The project work entitled “**Project Management System**" carried out by **Aakash S Rathva (22CP326)** is approved for the submission in the course **4CP33, Full Semester External Project** for the partial fulfillment for the award of the degree of B. Tech. in Computer Engineering.

Date:

Place:

Signatures of Examiners:

(Names and Affiliations)

CERTIFICATE

This is to certify that Project Work embodied in this project report titled “**Project Management System**" was carried out by **Aakash S Rathva (22CP326)** under the course **4CP33, Full Semester External Project** for the partial fulfillment for the award of the degree of B. Tech. in Computer Engineering. Followings are the supervisors at the student:

Date:

Place:

Satyam M Raval
Deputy General Manager
Tech Elecon
Anand

Prof**.** Hinaxi M Patel
Assistant Professor
Computer Engineering Department,
BVM Engineering College

Prof.Ami B Parekh
Assistant Professor
Computer Engineering Department,
BVM Engineering College

Dr. Darshak G Thakore
Professor & Head,
Computer Engineering Department,
BVM Engineering College

COMPUTER ENGINEERING DEPARTMENT, BVM ENGINEERING COLLEGE, VALLABH VIDYANAGAR – 388120

CERTIFICATE

This is to certify that Aakash S Rathva (22CP326) of 4th Level, B.Tech. in Computer Engineering of Birla Vishvakarma Mahavidyalaya, Vallabh Vidyanagar has worked on his project under the course (FSEP, 4CP33) titled **“Project Management system”** under the supervision of Satyam Raval at Tech elecon. He has done his work sincerely and has completed his project. We found his work satisfactory.

Place:

Date:

Name and Signature of the External Supervisor

Seal of the company

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this report under the course 4CP33 (Full Semester External Project) and that neither any part thereof nor the whole of the report has been submitted for a degree to any other University or Institution.

I certify that, to the best of my knowledge, the current report does not infringe upon anyone's copyright nor does it violate any proprietary rights and that any ideas, techniques, quotations or any other material from the work of other people included in my report, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the boundary of fair dealing within the meaning of the Indian Copyright (Amendment) Act 2012, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in the current report and have included copies of such copyright clearances to the appendix of this report.

I declare that this is a true copy of report, including any final revisions, as approved by the report review committee.

I have checked write-up of the present report using anti-plagiarism database and it is in permissible limit. However, at any time in future, in case of any complaint pertaining of plagiarism, I am the sole responsible person for the same. I also understand that, in case of such complaints of plagiarism, as per the University Grants Commission (UGC) norms, the University can even revoke the degree conferred to the student submitting such a plagiarized report.

Date:

Institute Code: 007

Institute Name: Birla Vishvakarma Mahavidyalaya (BVM) Engineering College

Signature

Aakash S Rathva

22CP326

ACKNOWLEDGEMENT

I am writing to express my sincere gratitude for the opportunity to participate in the React.js internship program. This experience has been incredibly valuable to my professional development, and I am grateful to the many individuals who contributed to its success.

First and foremost, I would like to thank Dr. Indrajit Patel, Principal and Professor Dr. Darshak G Thakore Head of the computer engineering Department, for their support and guidance throughout my internship. Their willingness to offer their expertise and provide valuable advice significantly shaped the outcome of my work. I am particularly grateful for their encouragement and assistance during the preparation of this internship report.

I would also like to extend my deepest appreciation to my industry mentor, Mr. Satyam Raval. His guidance, knowledge, and unwavering support throughout the internship were invaluable. Mr. Raval's dedication to sharing his expertise and his willingness to help me overcome challenges significantly contributed to the success of the project.

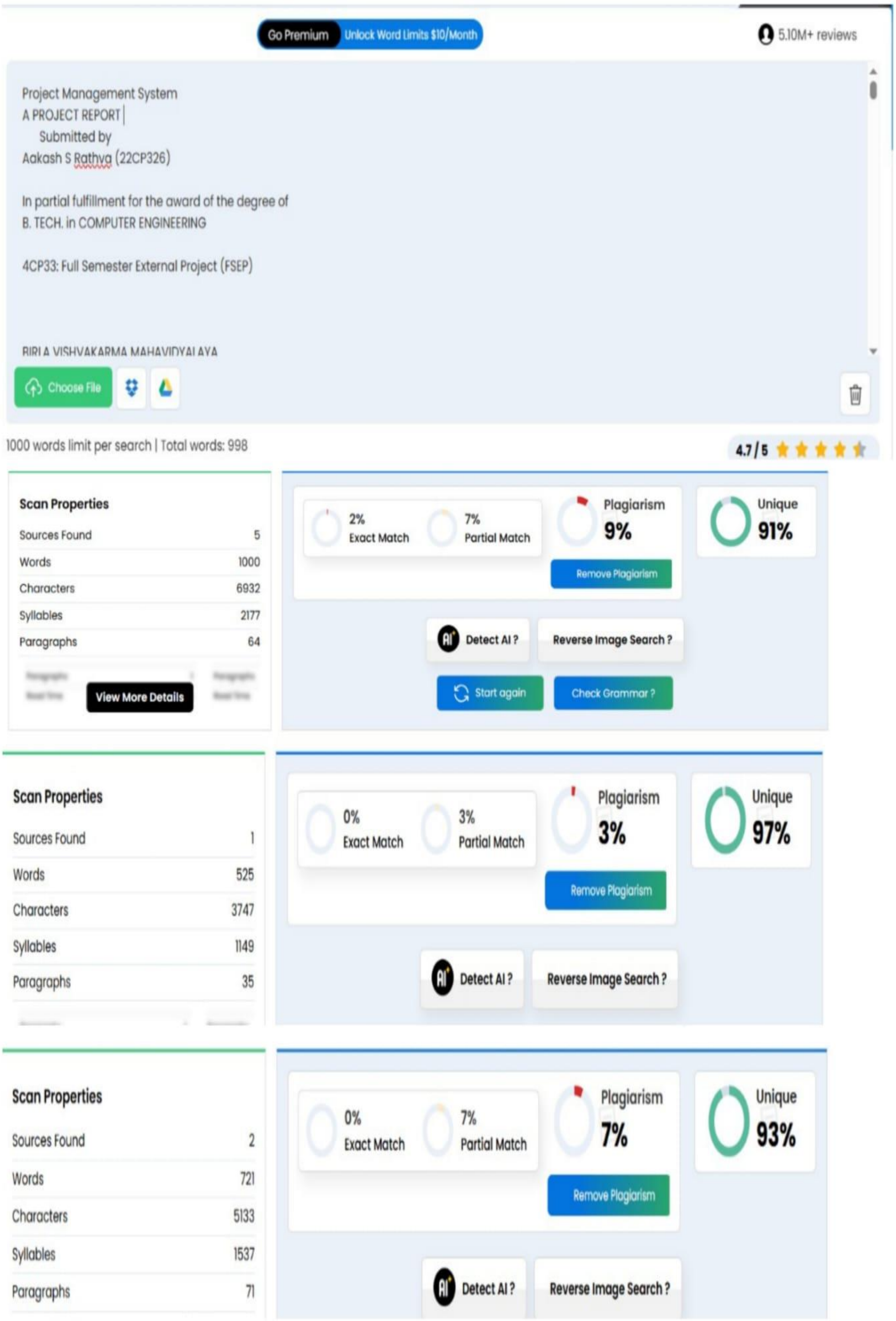
Furthermore, I am thankful to my fellow team members for their unwavering dedication, collaborative spirit, and hard work. Each team member brought unique skills and perspectives to the table, and their contributions played a crucial role in overcoming challenges and achieving project milestones.

Thank you once again to everyone who provided an opportunity to participate in this internship program. Your support and encouragement have been indispensable, and I am grateful for the opportunity to have learned and grown throughout this experience.

Aakash S Rathva

22CP326

Plagiarism Report



Project Management System
A PROJECT REPORT
Submitted by
Aakash S Rathva (22CP326)

In partial fulfillment for the award of the degree of
B. TECH. in COMPUTER ENGINEERING

4CP33: Full Semester External Project (FSEP)

BIRLA VISHVAKARMA MAHAVIDYALAYA

1000 words limit per search | Total words: 998

Scan Properties

Sources Found	5
Words	1000
Characters	6932
Syllables	2177
Paragraphs	64

2% Exact Match
7% Partial Match
Plagiarism 9%
Unique 91%

Scan Properties

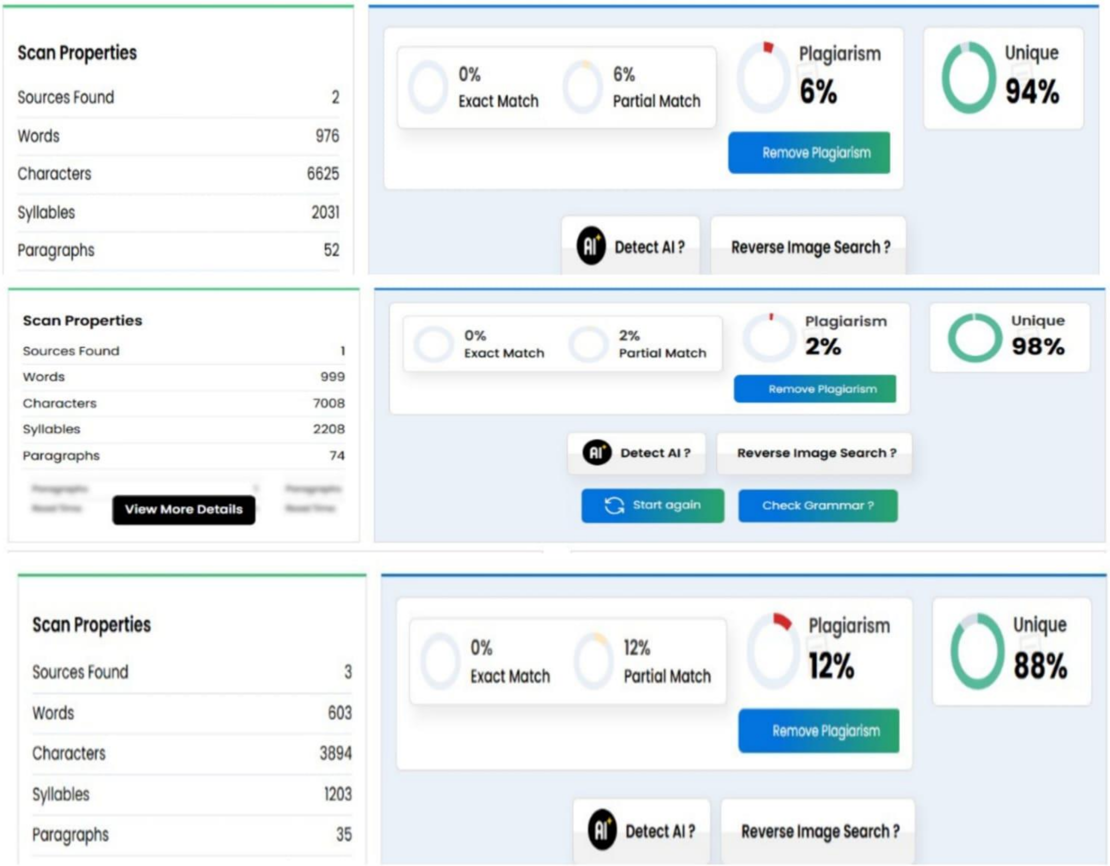
Sources Found	1
Words	525
Characters	3747
Syllables	1149
Paragraphs	35

0% Exact Match
3% Partial Match
Plagiarism 3%
Unique 97%

Scan Properties

Sources Found	2
Words	721
Characters	5133
Syllables	1537
Paragraphs	71

0% Exact Match
7% Partial Match
Plagiarism 7%
Unique 93%



Scan Properties

Sources Found	2
Words	976
Characters	6625
Syllables	2031
Paragraphs	52

0% Exact Match

6% Partial Match

Plagiarism 6%

Unique 94%

Scan Properties

Sources Found	1
Words	999
Characters	7008
Syllables	2208
Paragraphs	74

0% Exact Match

2% Partial Match

Plagiarism 2%

Unique 98%

Scan Properties

Sources Found	3
Words	603
Characters	3894
Syllables	1203
Paragraphs	35

0% Exact Match

12% Partial Match

Plagiarism 12%

Unique 88%

Figure Plagiarism

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, Employee, and Admin, improving productivity and task execution.

Our system leverages React.js for the front-end, Node.js 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 Node.js 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 Member 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.

Table of Contents

LIST OF FIGURES

List of Symbols, Abbreviations and Nomenclature

Term	Description
UI	User Interface
UX	User Experience
API	Application Programming Interface
DB	Database
JWT	JSON Web Token
HTML	HyperText Markup3 Language
CSS	Cascading Style Sheets
SPA	Single Page Application
CRUD	Create, Read, Update, Delete
SQL	Structured Query Language
ER Diagram	Entity Relationship Diagram
PM	Project Manager
TM	Team Member
ID	Identifier

1. Introduction

1.1 Overview

The **Project Management System** is designed to enhance the efficiency of organizations in managing their projects, tasks, and team collaborations. This system operates on a role-based structure, categorizing users into three primary roles:

- **Admin** – Responsible for managing user accounts, system settings, Assign Project and overall system health.
- **Project Manager** – Tasked with creating and assigning task, overseeing task completion, and Managing Team members.
- **Team Member** – Assigned specific tasks, which they can view, update, and collaborate on with other team members.

This system addresses the challenges associated with manual project tracking, such as miscommunication, task mismanagement, and project delays. By offering a centralized platform, users can log in, access project details, track progress, and ensure that all tasks are completed within the assigned timeframes.

A secure authentication system ensures that only authorized individuals can access specific functionalities based on their role, safeguarding data integrity. The system is designed to improve work efficiency through a structured, user-friendly workflow.

1.2 Purpose of the Project

The **Project Management System** aims to streamline the management of projects, tasks, and team collaboration for organizations. By offering a single platform to assign, track, and manage projects, the system simplifies the process of organizing and overseeing tasks.

For **Project Managers**, it provides the ability to assign tasks, set deadlines, and monitor progress, ensuring that projects stay on track.

The system offers a **role-based structure**, where **Admins** manage the system, **Project Managers** oversee the projects, and **Team Members** are responsible for task completion. This clear division of responsibilities minimizes confusion and optimizes work efficiency.

With this system, organizations can enhance productivity, improve communication among teams, and ensure that projects are completed on time and within scope. By eliminating manual tracking and fragmented communication channels, the system makes project management more streamlined and effective.

1.3 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.
  - Ensure each user has specific permissions based on their role.
- **Project Management**
  - Enable **Project Managers** to create, edit, and delete projects tasks
  - Provide clear project overviews, including deadlines, progress tracking, and assigned tasks.
  - Offer easy search and filtering options for efficient management of multiple projects.
- **Task Assignment and Tracking**
  - Allow **Project Managers** to assign tasks to **Team Members** and set deadlines.
  - Enable **Team Members** to update the status of tasks (e.g., In Progress, Completed, Pending).
  - Provide task filtering options based on status, priority, or project.
- **User-Friendly Interface**
  - Ensure a clean, intuitive, and simple user interface.
  - Provide clear navigation for each role to access their respective features.
  - Implement notifications to keep users updated on task progress and deadlines.
- **Secure System and Data Management**
  - Store all project and task data securely in a backend database.
  - Restrict access to authorized users only, ensuring data security.
  - Maintain data consistency and integrity across all user roles and modules.
- **Collaboration and Team Communication**
  - Enable **Team Members** to collaborate on tasks and see who is responsible for what.
  - Allow real-time updates to tasks, improving collaboration.
  - Enhance teamwork by ensuring each member knows their responsibilities within the project.

1.4 Scope of the Project

The **scope** of the **Project Management System** encompasses its functionalities, features, and limitations. It is designed for small to medium-sized teams and organizations seeking to improve the planning, execution, and tracking of their projects.

1.4.1 Features and Functionalities

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

1.4.2 Backend and Database Management

- The system will store all **user**, **project**, and **task** data securely.
- Real-time data updates will ensure accuracy and reliability.
- Role-based authentication ensures privacy and security of user data.

2. Tools And Technology

2.1 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.

2.2 HTML (HyperText 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.

2.3 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.

2.4 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.

2.5 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 `<Browser Router>` 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.

2.6 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.

2.7 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.

2.8 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.

2.9 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.

2.10 Java script:

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.

2.11 GitHub:

GitHub is a web-based platform and version control system that enables developers to collaborate on projects, host code repositories, and manage software development workflows. Below is a brief overview of its core features:

- **Version Control**: GitHub uses Git, a distributed version control system, to track changes to files and manage different versions of a code base. Developers can create branches to work on features or bug fixes independently, merge changes into the main branch, and revert to earlier versions when necessary.
- **Code Hosting**: GitHub offers a centralized platform for hosting Git repositories, making it easy to share code. Repositories can be public or private, and GitHub provides features such as issue tracking, project boards, and wikis to support collaboration and project management.

- **Collaboration**: GitHub supports collaborative development by allowing users to fork repositories, make changes, and submit pull requests for review and integration. Developers can discuss code changes, review proposed modifications, and provide feedback through comments and code reviews.
- **Community and Open Source**: GitHub is a central hub for the open-source community, enabling contributions to public projects, discovery of new tools and frameworks, and interaction with other developers. It hosts millions of public repositories across diverse fields such as web development, data science, machine learning, and gaming.

3. System Analysis

3.1 System Features

3.1.1 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.

3.1.2 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.

3.1.3 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.
  - Passwords will be hashed and stored securely in the MySQL database.

- **Error Checking:**
  - Forms will automatically validate user inputs, checking for missing fields or incorrect data.
  - Instant feedback will help users correct errors before submission.

3.1.4 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.

3.1.5 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.

3.2 Feasibility Study

3.2.1 Technical Feasibility

- **TechnologyStack:**

  The system uses React.js for the front end to create a smooth and interactive user experience. The backend is developed using Node.js to handle server-side logic efficiently. MySQL is used for securely storing and managing project, task, and user data. JWT (JSON Web Token) is implemented for secure user authentication and session management.

- **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 Node.js ensures a fast and stable connection between users and the database.
- **DevelopmentResources:**
  React.js, Node.Js, and MySQL are widely used technologies, making it easier to find developers for future updates. The component-based structure of React.js simplifies maintenance and feature upgrades. Node.Js provides a robust backend solution, ensuring the system runs smoothly even with multiple users.

3.2.2 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. Backend updates in Node.js and database modifications in MySQL can be done without affecting the entire system. Regular updates will ensure bug fixes, performance improvements, and security enhancements.
- **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. Future updates can include an in-app help section or chatbot for real-time assistance.

3.2.3 Economic Feasibility

- **Development**

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

- **Operational**

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

- **Long-Term**

  The system will help increase productivity, reduce miscommunication, and improve project tracking, leading to cost savings in management and efficiency. The automation of task assignments and tracking reduces manual effort, making project execution smoother and reducing operational overhead.

3.3 System Software and Hardware Requirements for Web App

3.3.1 Development Environment

- **Operating System:**
  - Recommended: Windows 10 or 11, Mac OS 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:**
  - **Backend:** Node.Js for handling business logic.
  - **Frontend:** React.js for a responsive user interface.
  - **Database:** MySQL for storing and managing project, task, and user data.
  - **Development Tools:** Visual Studio Code, IntelliJ IDEA,
  - **Package Managers:** NPM (for React.js and node.js)
  - **Version Control:** Git/GitHub for code collaboration and tracking.

3.3.2 Deployment Environment

The deployment environment (where the React application runs after development) will vary depending on the chosen hosting provider. However, some general requirements include:

- **Node.js runtime :** If the application requires server-side rendering or other Node.js functionality

4 System Design

4.1 Data Flow Diagram 0 Level :-

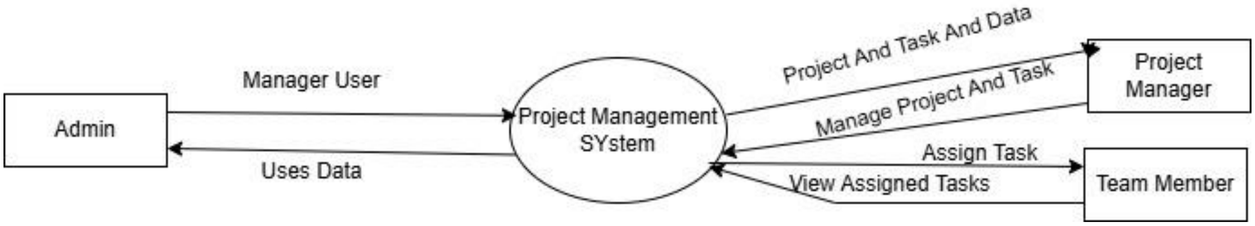


Figure 4.1 DFD 0 Level

This Horizontal DFD Level 0 Shows how Admins And Project Manager, Team Member Interact With Central System

Level 1 DFD

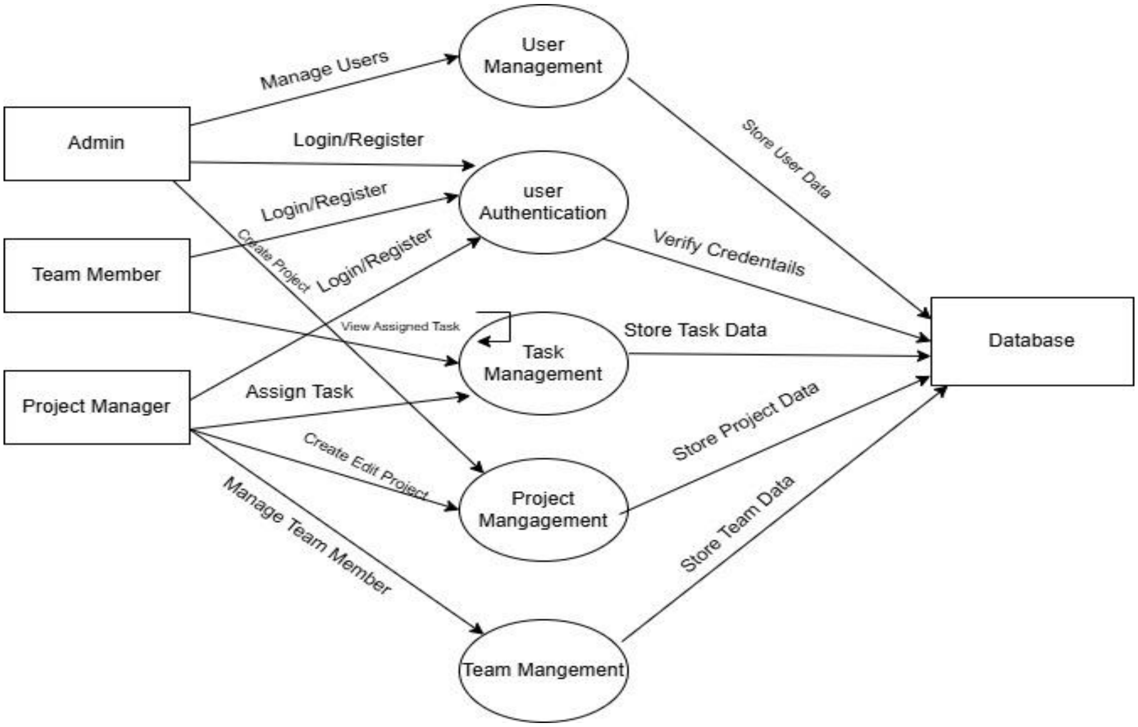


Figure 4.1 DFD 1 Level

4.2 Use Case Diagram

represents the main functionalities of the Project Management System and shows the interactions between different types of users (Admin, Project Manager, Team Member) and the system.

In the diagram:

- **Admin** can manage users, view system analytics, and oversee project and task management.
- **Project Manager** can create projects, assign tasks, manage team members, and track project progress.
- **Team Member** can view assigned tasks, update task statuses, and collaborate with other members.

The diagram helps in identifying all the system requirements from a user's point of view and clearly outlines the various services the system provides to its users.

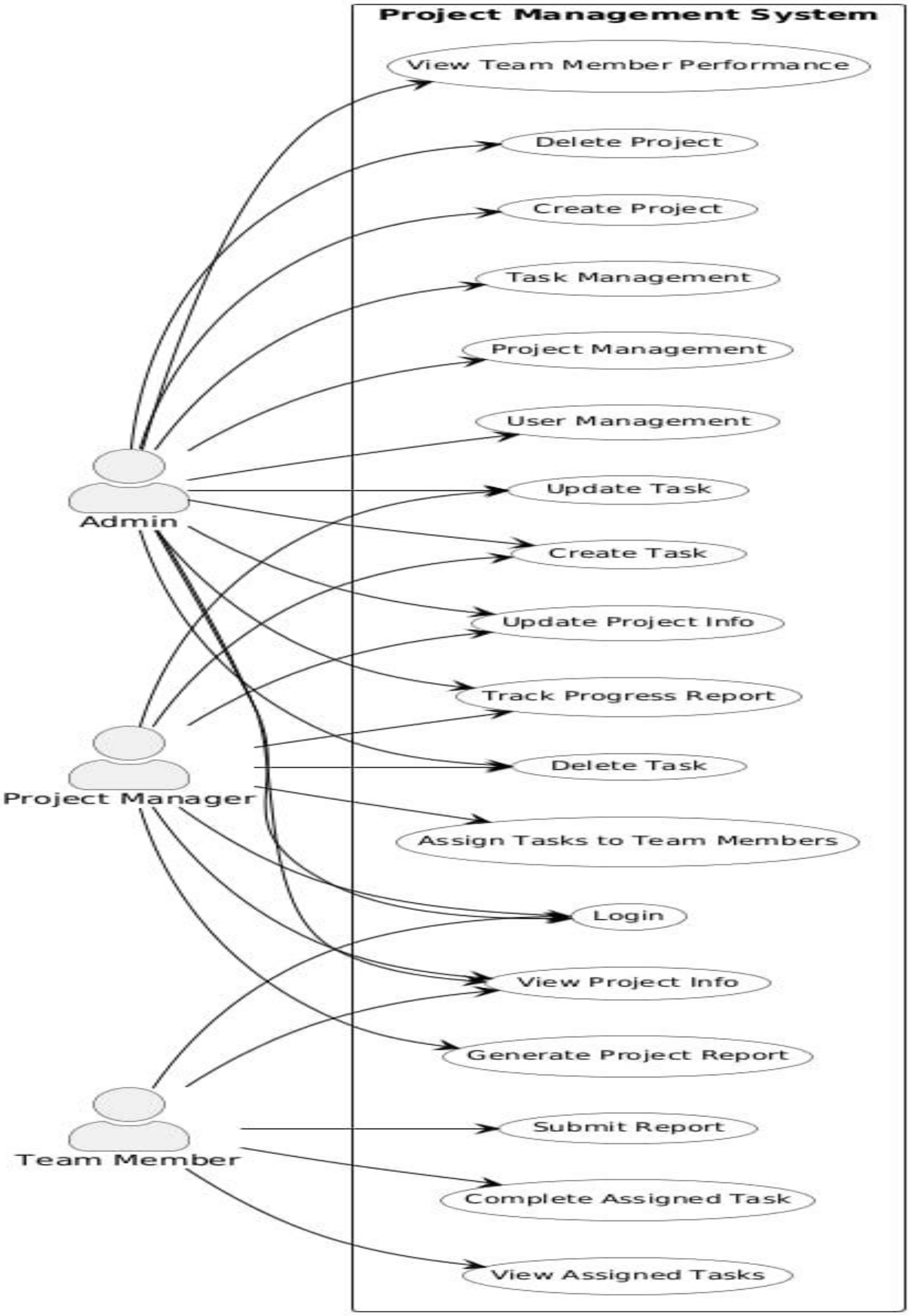


Figure 4.2 Use Case Diagram

4.3 Class Diagram:

The Class Diagram shows the main entities of the Project Management System and their relationships.

- **Users** can create projects, tasks, comments, upload files, and participate in events.
- **Projects** are created by users and contain tasks, events, files, and comments.
- **Tasks** are assigned to users and linked to projects.
- **Team Members** link users to projects, managing team assignments.
- **Task Assginements :-** Task Assginments to Users
- **PM** :- Project Manager
- **Admin** users manage the system separately.

This design ensures smooth project management, user collaboration, and secure data handling

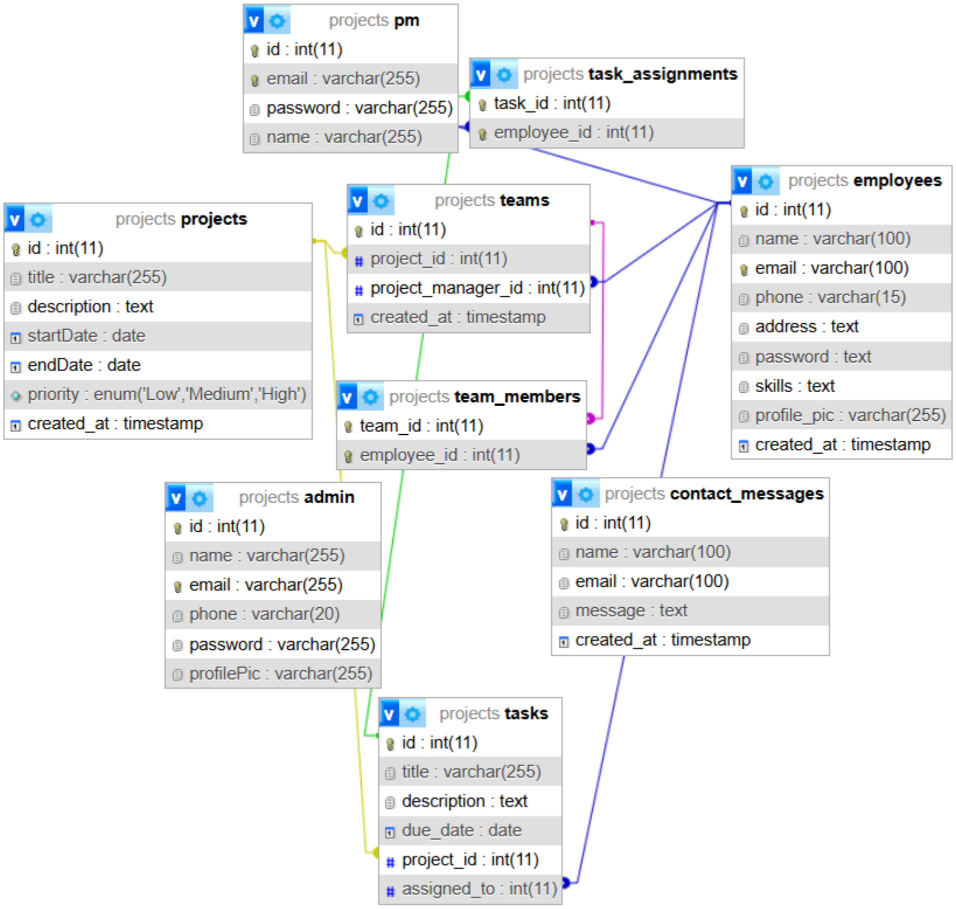


Figure 4.3 Class Diagram

4.4 Activity Diagram :-

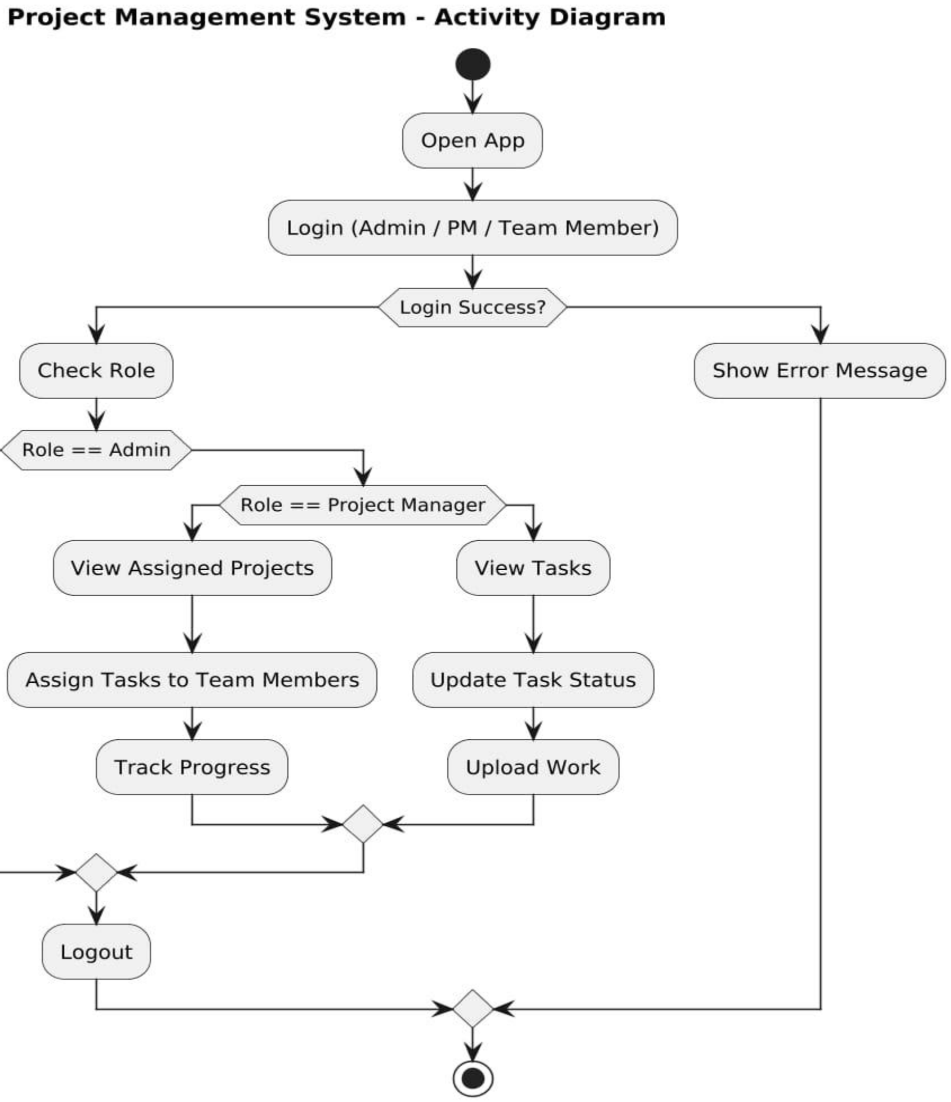
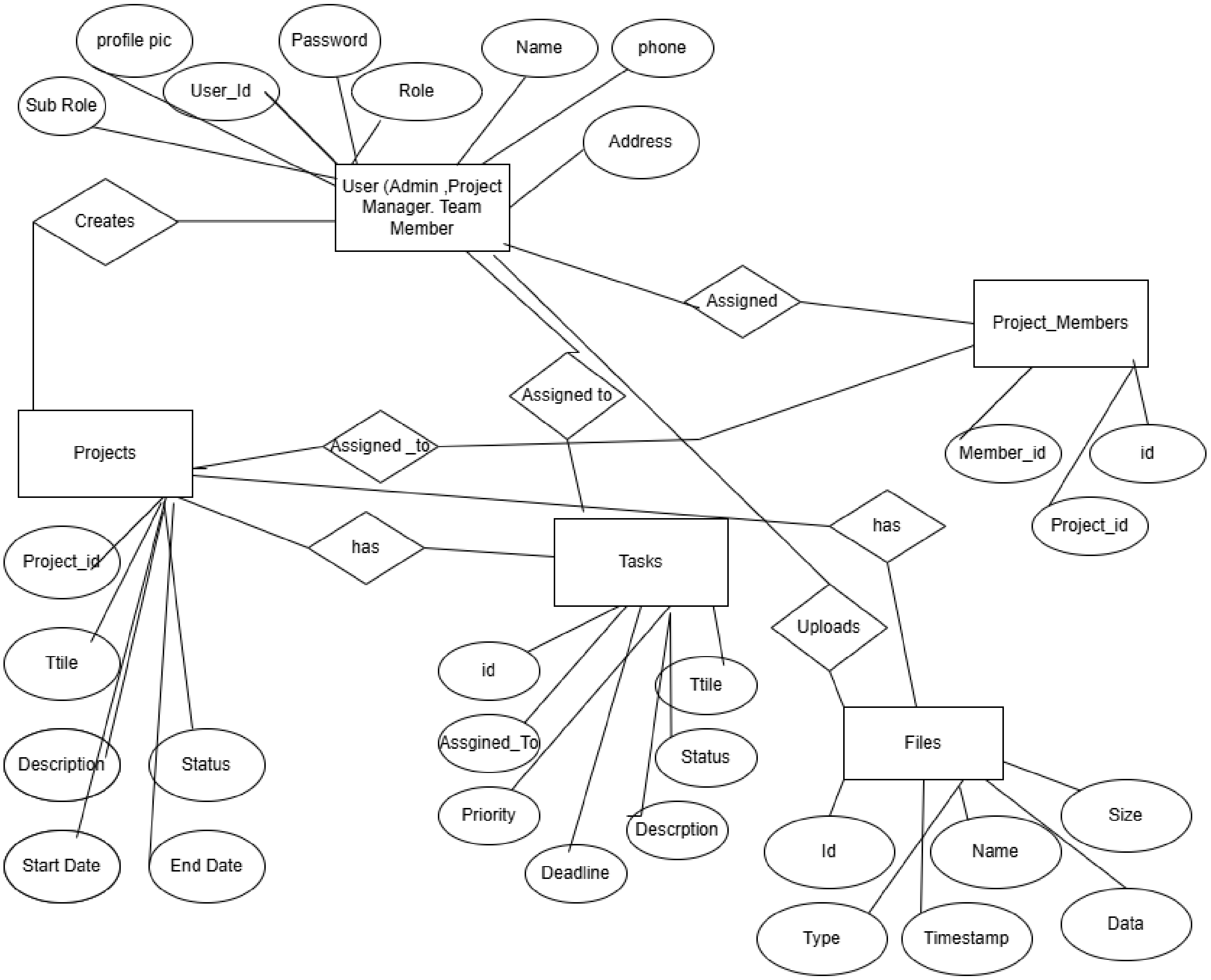


Figure 4.4 Activity Diagram

The activity diagram illustrates the workflow of core operations in the Project Management System, such as project creation, task assignment, and status updates. It visually represents how users (Admin, Project Manager, and Team Member) interact with different system modules

4.5 E-R Diagram :



` Figure 4.4 ER Diagram

The Entity-Relationship (ER) Diagram represents the core structure of the Project Management System, showing the relationships between users, projects, tasks, files, Users can create projects, upload files, Projects are created by users and have multiple tasks, files, associated with them.Tasks are assigned to users and linked to specific projects Files are uploaded by users and linked to projects.Project Members table manages the assignment of users to projects.

This ER diagram ensures clear data management, proper user-project-task relationships, and supports effective project collaboration

4.5 Data Dictionary:

4.5.1 Admin

Table 4.1

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	Admin's email address
Phone	Varchar(255)	Not Null	Encrypted user password
Password	Varchar(255)	Not Null	User Password
Profile_Pic	Varchar(255)	Not Null	Store Image

4.5.2 Employee

Table 4.2

Field Name	Data Type	Constraints	Description
Id	Int	Primary Key, Auto-Increment	Unique identifier
Name	Varchar (100)	Not Null	Name of Employee
Email	Varchar(100)	Null	Employee Email Store
Phone	Varchar(15)	Null	Employee phone Number Store
Address	Text	Null	Store Address

Password	Text	Not Null	Emp Password
Skills	Text	Null	User Skill
Profile_Pic	Varchar(255)	Null	User Profile

4.5.3 Tasks Table

`

Table 4.3

Field Name	Data Type	Constraints	Description
Id	Int	Primary Key, Auto-Increment	Unique identifier for each task
Title	Varchar255)	Not Null	Title or name of the task
Description	Text	NULL	Detailed description of the task
Due_Date	Date	Not Null	Due Date of Task
Poject_Id	Int	Foreign Key	Project id
Assigned_to	Int	Foreign Key (users.id)	User ID to whom the task is assigned

4.5.4 Projects

Table 4.4

Field Name	Data Type	Constraints	Description
Id	INT	Primary Key, Auto-Increment	Unique identifier for each project-member assignment
Title	Varchar(255)	Null	Name of Project
Description	Text	Null	Detail of Project

Start Date	Date	Null	Start Date of Project
End Date	Date	Null	End Date of project
Priority	Enum	Null	Priority of Project
Created_at	Timestamp	Default	Sate and Time of Last Update

4.5.5 Project Manager(PM)

Table 4.5

Field Name	Data Type	Constraints	Description
Id	INT	Primary Key, Auto-Increment	Unique identifier for each event
Email	Varchar(255)	Not Null	PM Email Id
Password	Varchar(255)	Not Null	PM Email Id
Name	Varchar(255)	Not Null	Name Of Pm

4.5.6 Task Assginments

Table 4.6

Field Name	Data Type	Constraints	Description
Task_Id	Int	Primary Key, Auto-Increment	Unique identifier for each comment
Employee_id	Int	Foreign Key (projects.id), NOT NULL	ID of the Employee the comment is associated with

4.5.7 Teams

Table 4.7

Field Name	**Data Type**	**Constraints**	**Decription**
Id	Int	Primary Key, Auto Increment	Unique Identifier
Project_Id	Int	Foreign Key (Not Null)	Id of Project
PM_id	Int	Foreign Key (Not Null)	Project Manager ID
Created_at	Timestamp	Deafult Timestamp	Date And Time Team Update

4.5.8 Team Member

Table 4.8

Field Name	**Data Type**	**Constraints**	**Description**
Team_Id	Int	Primary Key, Auto-Increment	Unique identifier for each Team
Employee_Id	Int	Foreign Key Not Null	Employee Id

4.5.9 Contact

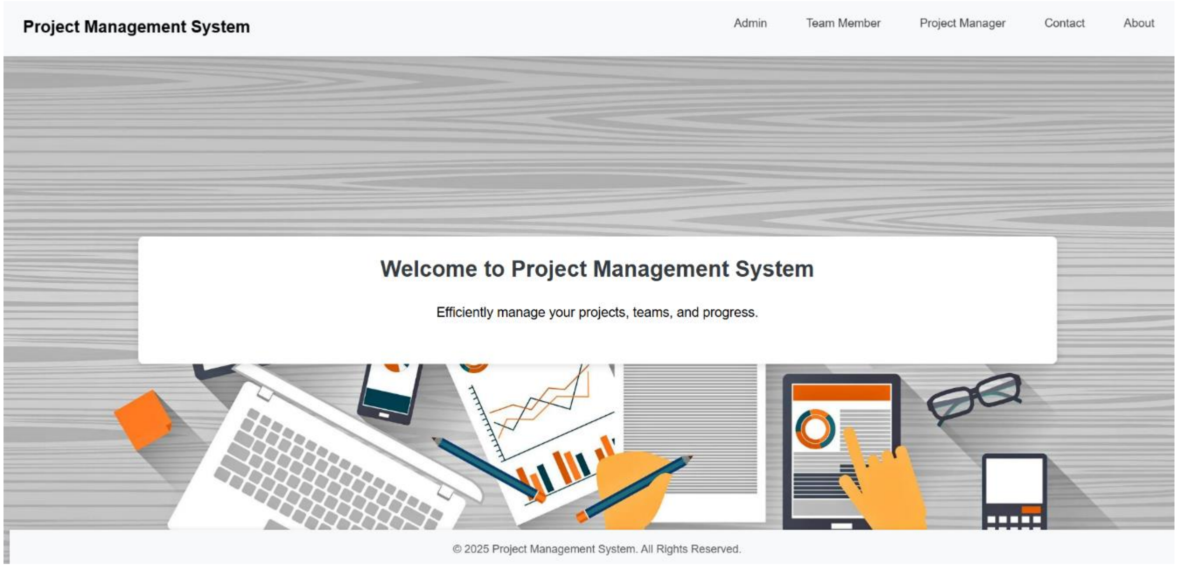
Table 4.9

Field Name	**Data Type**	**Constraints**	**Description**
Id	Int	Primary Key (Not Null)	Unique id
Name	Varchar(100)	Null	Name Of User
Email	Varchar(100)	Null	Email of User
Message	Text	Null	Meessage of User
Created_At	TimeStamp	Deafult_Timestamp	Date And Time of Message

5. Implementation

5.1 Home Page:

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



Welcome to Project Management System

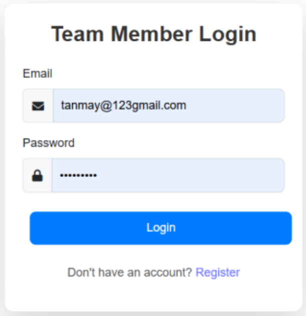
Efficiently manage your projects, teams, and progress.

© 2025 Project Management System. All Rights Reserved.

Figure 5.1 Home Page

5.2 Login Page: Team Member

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



Team Member Login

Email: tanmay@123gmail.com

Password: •••••••••

Figure 5.2 Login Page

5.3 Registration page:

The Registration Page allows new users to create an account. Users will be asked to provide basic information such as name, email, password,

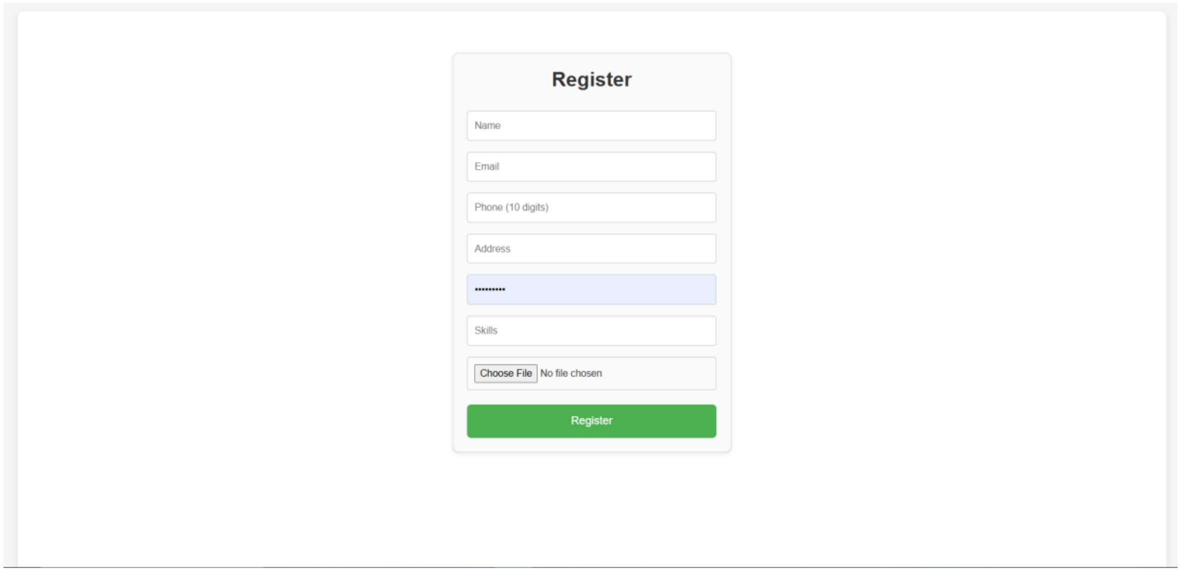
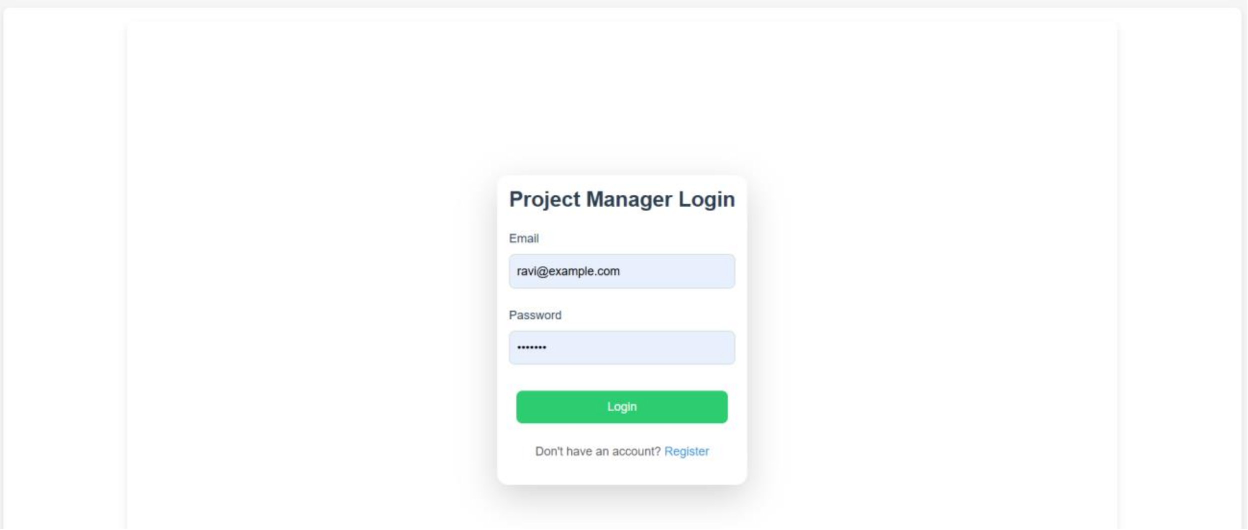


Figure 5.3 Registration Page

5.3.1 Login Page: Project Manager

The Project Manager 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



Project Manager Login

Email

ravi@example.com

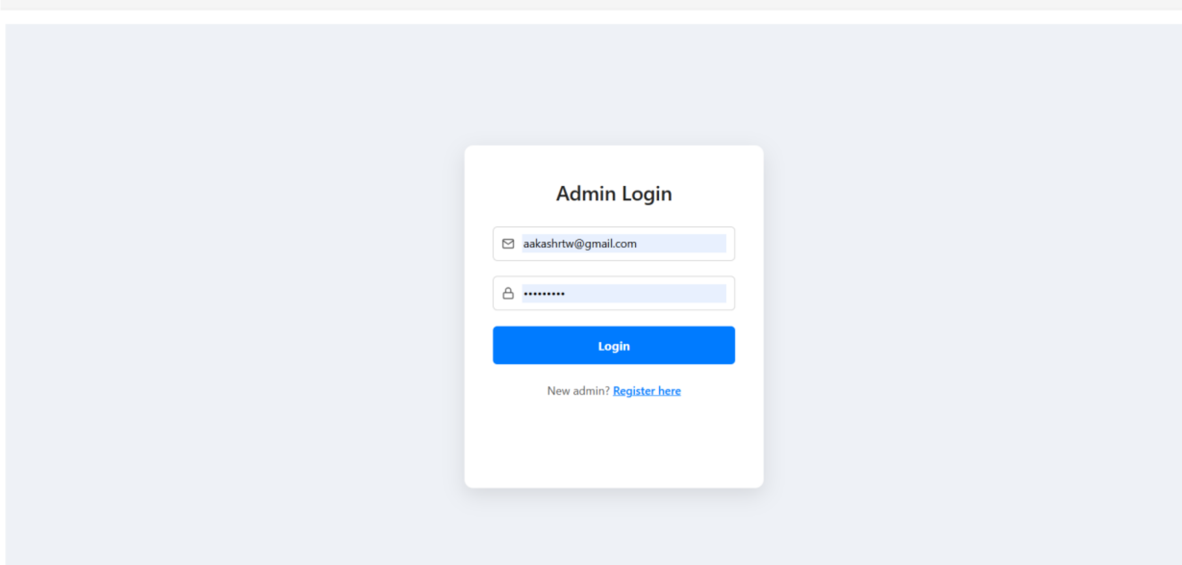
Password

•••••••

Figure 5.3.1 Project Manager Login

5.4 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.



Admin Login

aakashrtw@gmail.com

•••••••••

Figure 5.4 Admin Login Page

5.5 Admin Dashboard :-

The Admin Dashboard serves as the central hub for Admin. On this page, Admin 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.

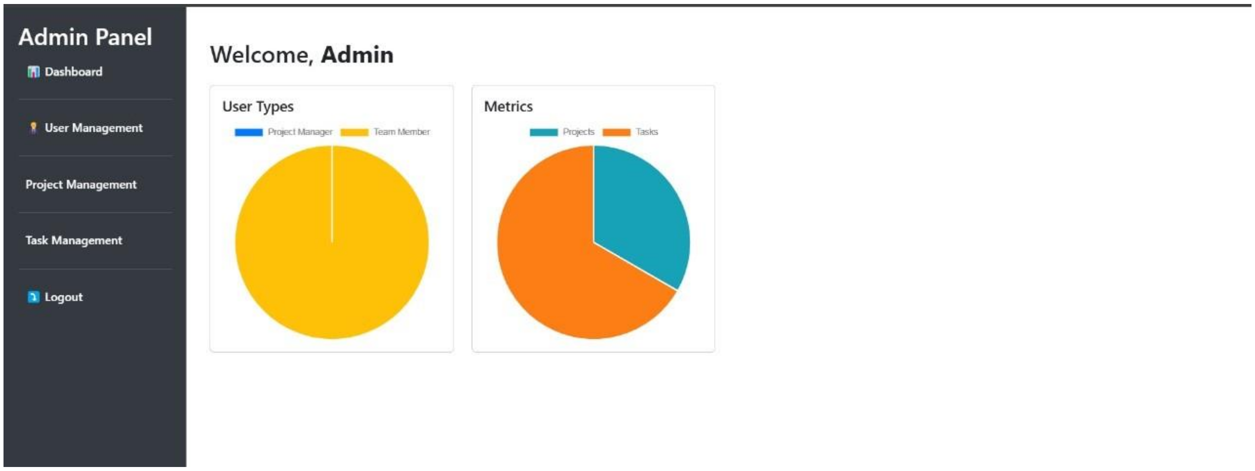
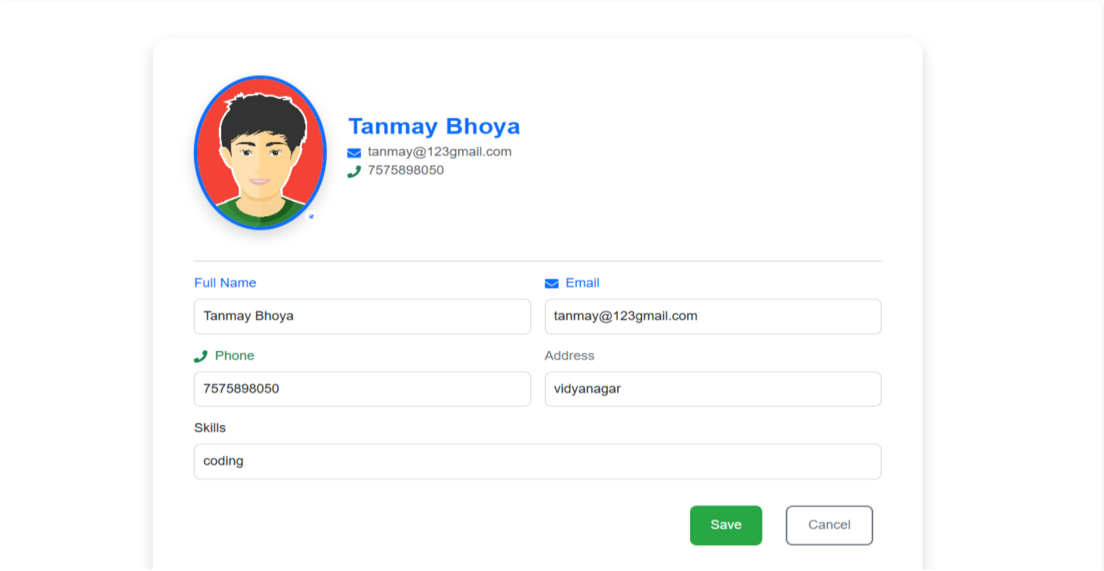


Figure 5.5 Admin Dashboard Page

5.6 Profile web 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.



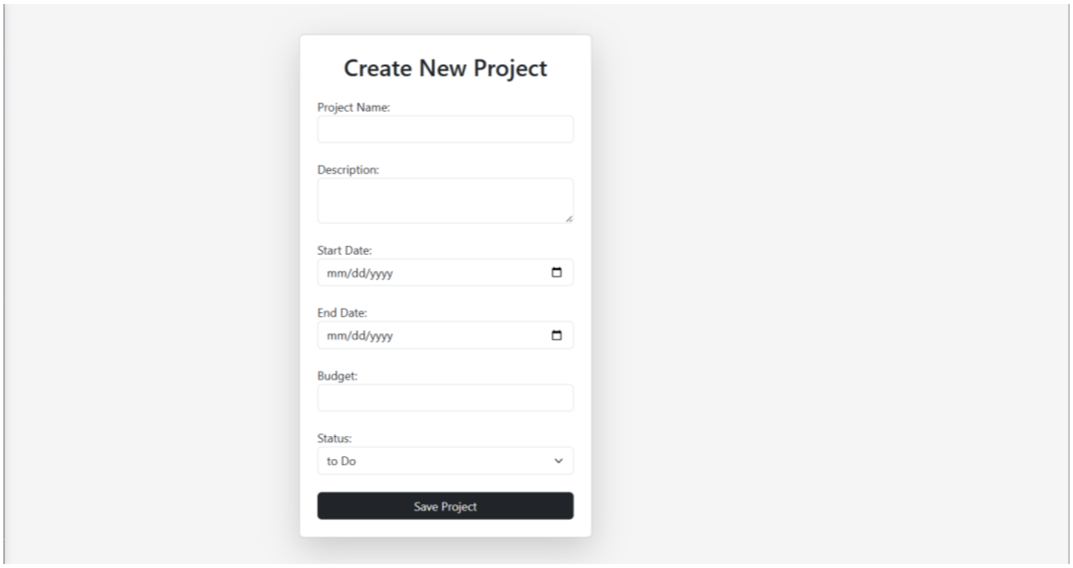
Tanmay Bhoya
✉ tanmay@123gmail.com
☎ 7575898050

✉ Full Name: Tanmay Bhoya
Email: tanmay@123gmail.com
☎ Phone: 7575898050
Address: vidyanagar
Skills: coding

Figure 5.6 Project Manager Profile Page

5.7 Project form 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.



Create New Project

Project Name:

Description:

Start Date: mm/dd/yyyy

End Date: mm/dd/yyyy

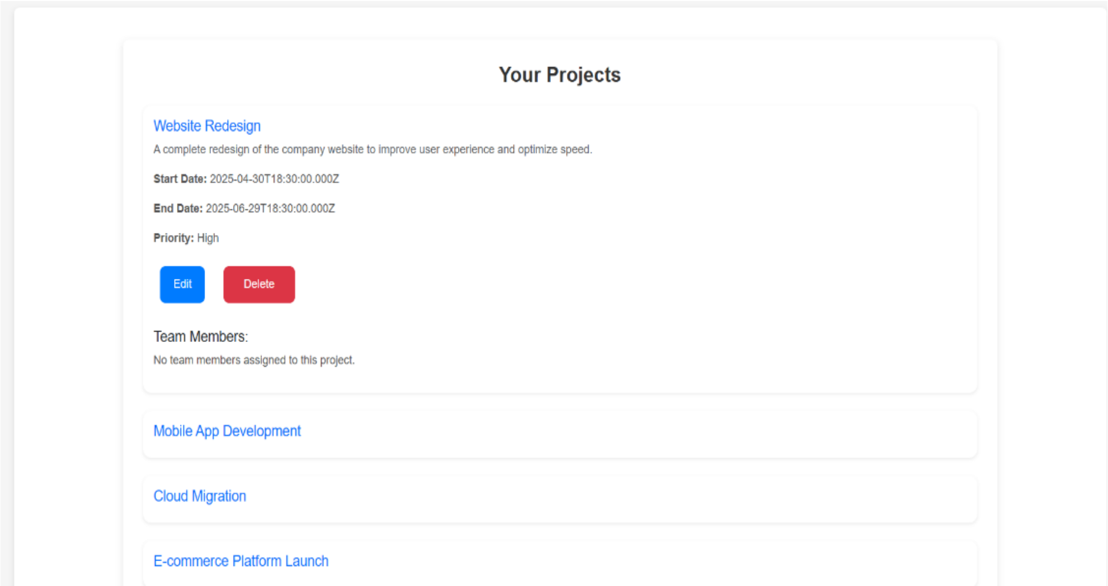
Budget:

Status: to Do

Figure 5.7 Project form Page

5.8 Project list Page:

The Project List Page displays a comprehensive overview of all projects currently being managed. Admin can search, filter, and sort projects based on various criteria such as project status, priority, or deadline.



Your Projects

Website Redesign

A complete redesign of the company website to improve user experience and optimize speed.

Start Date: 2025-04-30T18:30:00.000Z

End Date: 2025-06-29T18:30:00.000Z

Priority: High

Team Members:

No team members assigned to this project.

Mobile App Development

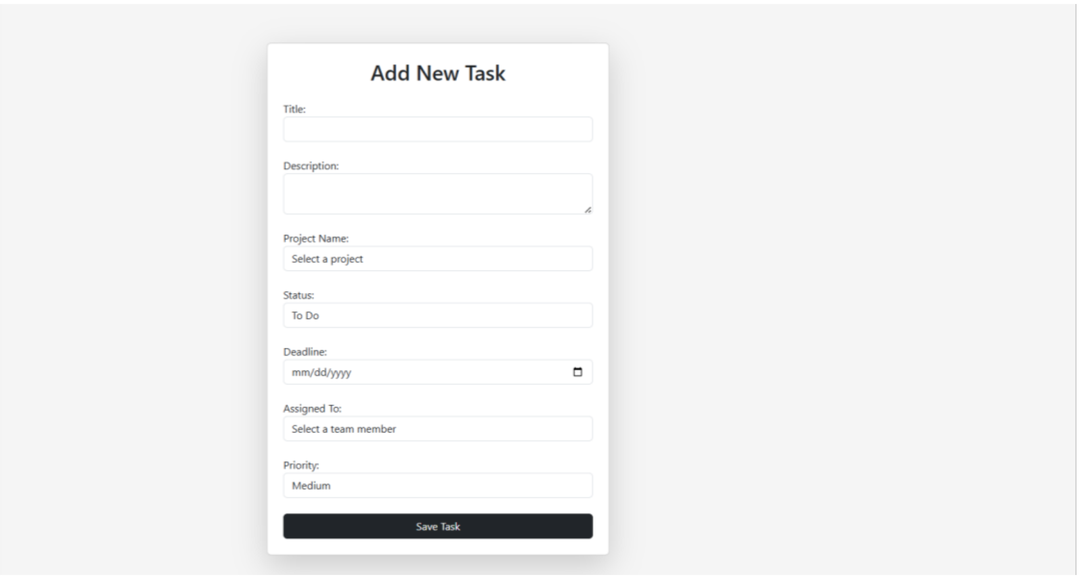
Cloud Migration

E-commerce Platform Launch

Figure 5.8 Project list Page

5.9 Task form 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.



Add New Task

Title:

Description:

Project Name: Select a project

Status: To Do

Deadline: mm/dd/yyyy

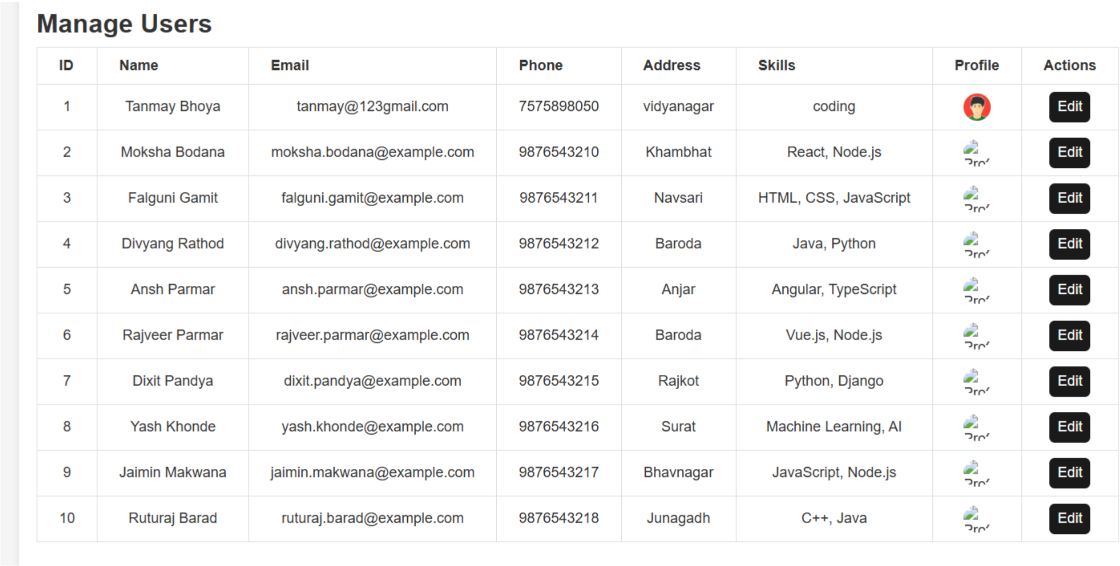
Assigned To: Select a team member

Priority: Medium

Figure 5.9 Task form Page

5.10 User Management page:

In this page all the user listed



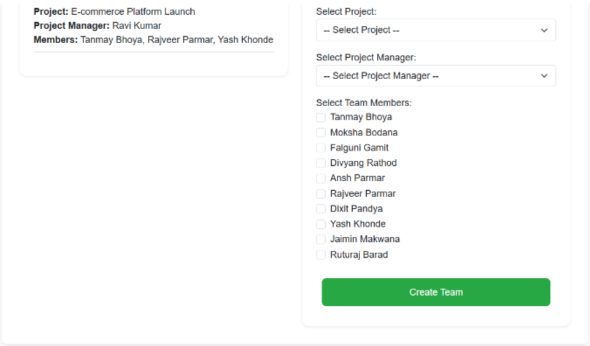
Manage Users

ID	Name	Email	Phone	Address	Skills	Profile	Actions
1	Tanmay Bhoya	tanmay@123gmail.com	7575898050	vidyanagar	coding		Edit
2	Moksha Bodana	moksha.bodana@example.com	9876543210	Khambhat	React, Node.js		Edit
3	Falguni Gamit	falguni.gamit@example.com	9876543211	Navsari	HTML, CSS, JavaScript		Edit
4	Divyang Rathod	divyang.rathod@example.com	9876543212	Baroda	Java, Python		Edit
5	Ansh Parmar	ansh.parmar@example.com	9876543213	Anjar	Angular, TypeScript		Edit
6	Rajveer Parmar	rajveer.parmar@example.com	9876543214	Baroda	Vue.js, Node.js		Edit
7	Dixit Pandya	dixit.pandya@example.com	9876543215	Rajkot	Python, Django		Edit
8	Yash Khonde	yash.khonde@example.com	9876543216	Surat	Machine Learning, AI		Edit
9	Jaimin Makwana	jaimin.makwana@example.com	9876543217	Bhavnagar	JavaScript, Node.js		Edit
10	Ruturaj Barad	ruturaj.barad@example.com	9876543218	Junagadh	C++, Java		Edit

Figure 5.10 Use Page

5.11 Assign Projects To Member

Admin Can Assign Project to Project Manager and Team Member



Project: E-commerce Platform Launch
Project Manager: Ravi Kumar
Members: Tanmay Bhoya, Rajveer Parmar, Yash Khonde

Select Project:
-- Select Project --

Select Project Manager:
-- Select Project Manager --

Select Team Members:
- □ Tanmay Bhoya
- □ Moksha Bodana
- □ Falguni Gamit
- □ Divyang Rathod
- □ Ansh Parmar
- □ Rajveer Parmar
- □ Dixit Pandya
- □ Yash Khonde
- □ Jaimin Makwana
- □ Ruturaj Barad

Figure 5.11 Project Assign Of Page

5.13 Team member’s dashboard:

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.

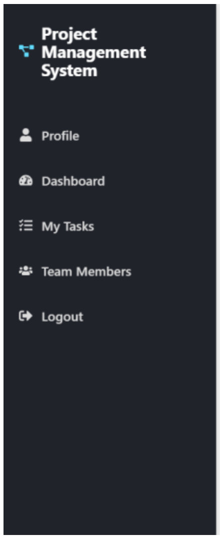
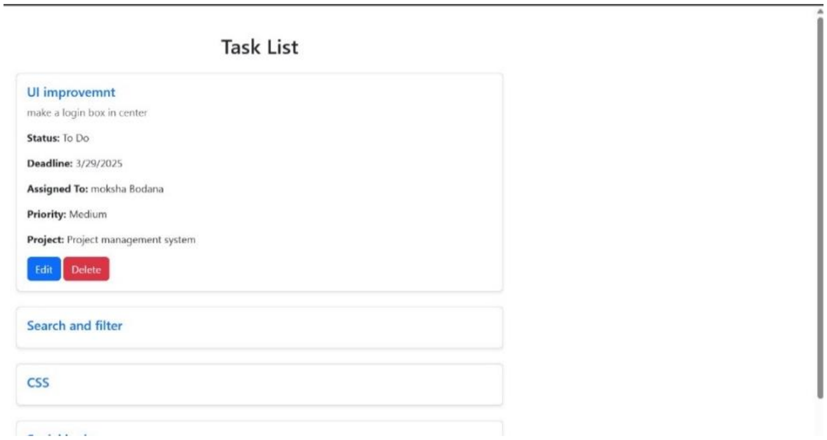


Figure 5.13 Team member Dashboard Page

5.14 Team member’s task list:

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.



Task List

UI improvemnt

make a login box in center

Status: To Do

Deadline: 3/29/2025

Assigned To: moksha Bodana

Priority: Medium

Project: Project management system

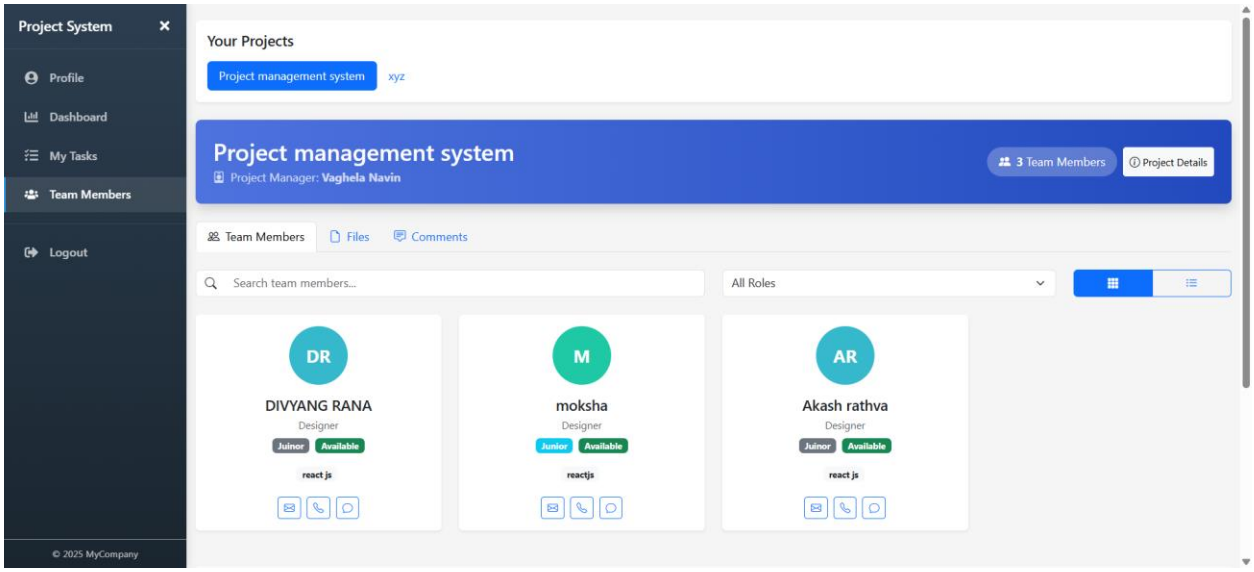
Search and filter

CSS

Figure 5.14 Team member’s task list Page

5.15 Team member’s team:

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



Your Projects

Project management system

Project Manager: Vaghela Navin

DIVYANG RANA
Designer
Juinor · Available
react js

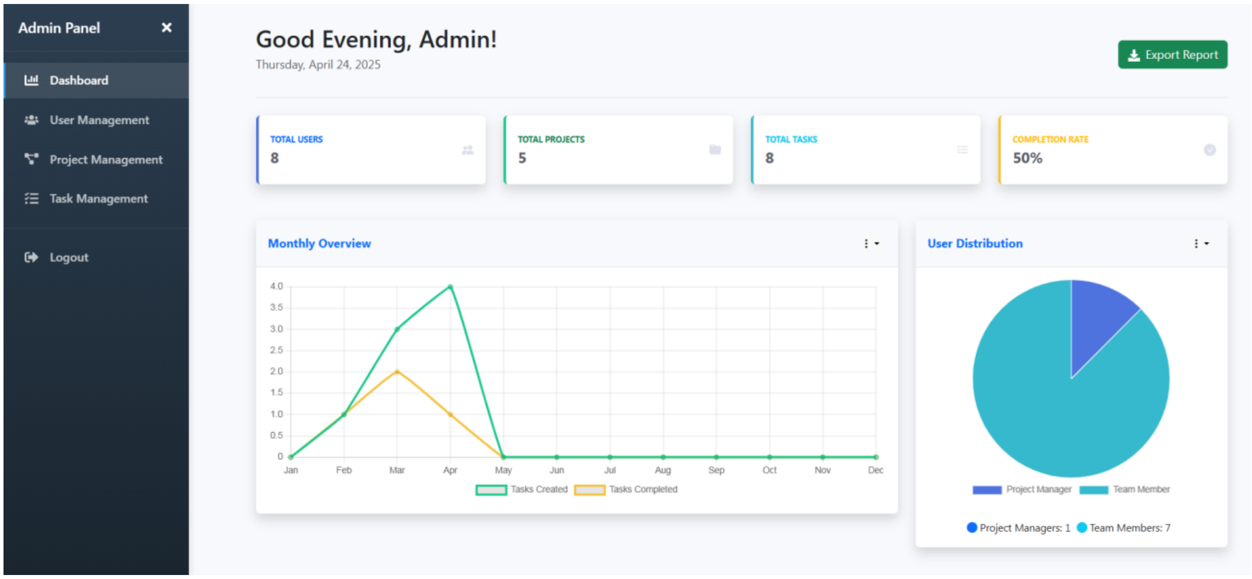
moksha
Designer
Junior · Available
reactjs

Akash rathva
Designer
Juinor · Available
react js

Figure 5.15 Team member’s Team Page

5.16 Admin Dashboard:

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.



Good Evening, Admin!

Thursday, April 24, 2025

TOTAL USERS 8

TOTAL PROJECTS 5

TOTAL TASKS 8

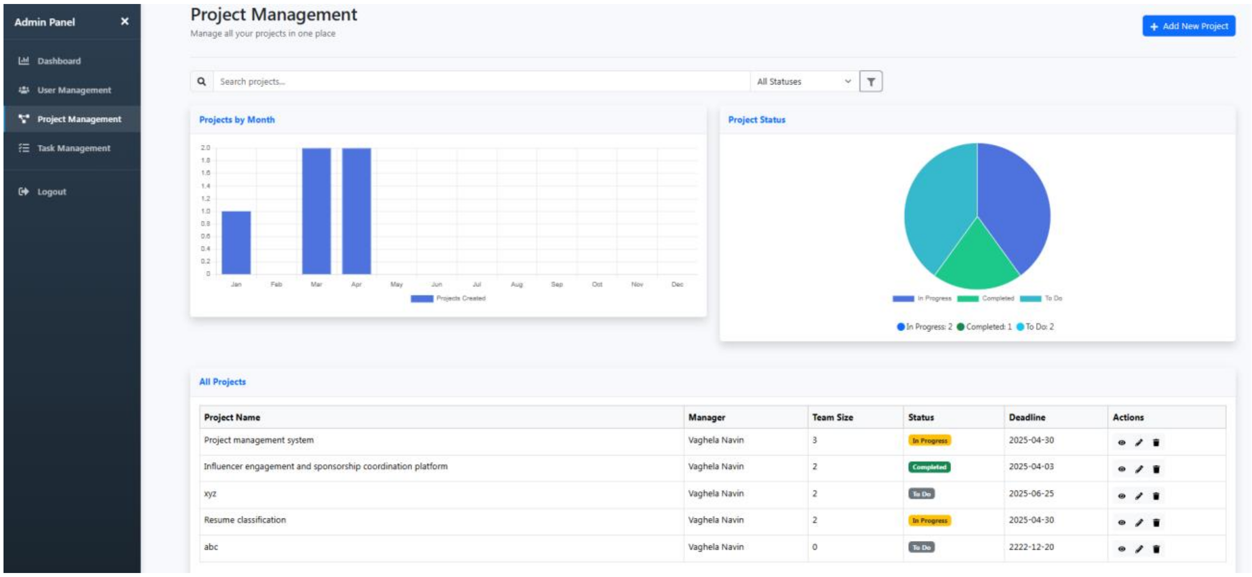
COMPLETION RATE 50%

● Project Managers: 1 ● Team Members: 7

Figure 5.16 Admin Dashboard Page

5.18 Admin’s Project management:

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.



Project Management

Manage all your projects in one place

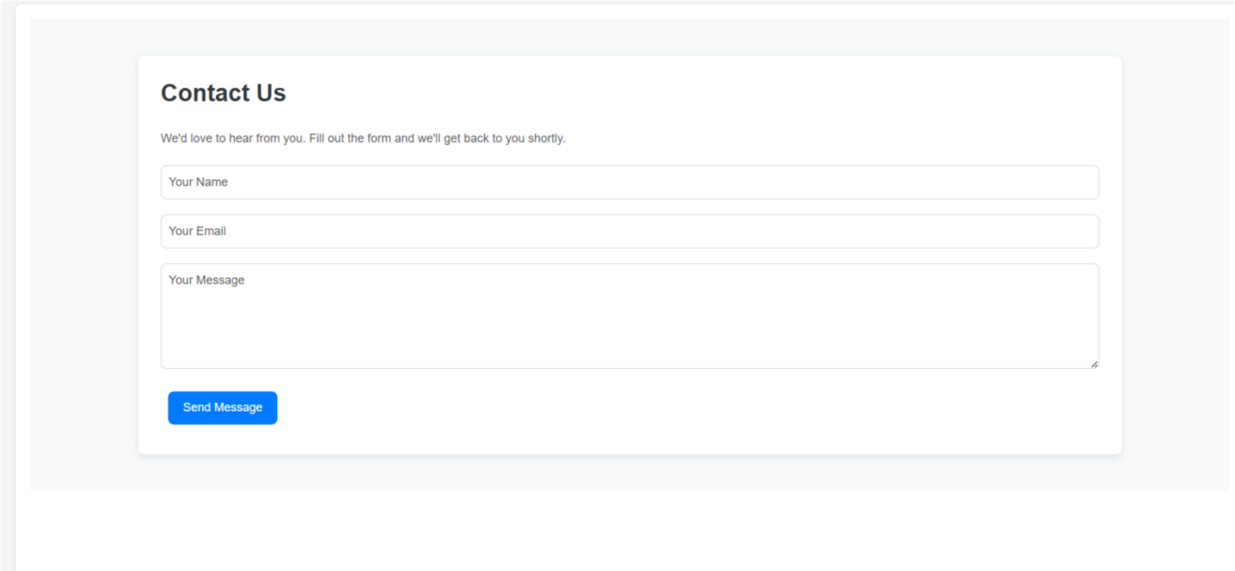
All Projects

Project Name	Manager	Team Size	Status	Deadline	Actions
Project management system	Vaghela Navin	3	In Progress	2025-04-30	👁 ✎ 🗑
Influencer engagement and sponsorship coordination platform	Vaghela Navin	2	Completed	2025-04-03	👁 ✎ 🗑
xyz	Vaghela Navin	2	To Do	2025-06-25	👁 ✎ 🗑
Resume classification	Vaghela Navin	2	In Progress	2025-04-30	👁 ✎ 🗑
abc	Vaghela Navin	0	To Do	2222-12-20	👁 ✎ 🗑

Figure 5.17 Admin Project management Page

5.18 Contact Page :-

The **Contact Page** provides users with a direct way to reach out for support, feedback, or inquiries related to the Project Management System. It typically includes a contact form allowing users to submit their name, email address, subject, and message. This form helps gather feedback or support requests efficiently and can be extended to integrate with email services or admin dashboards..



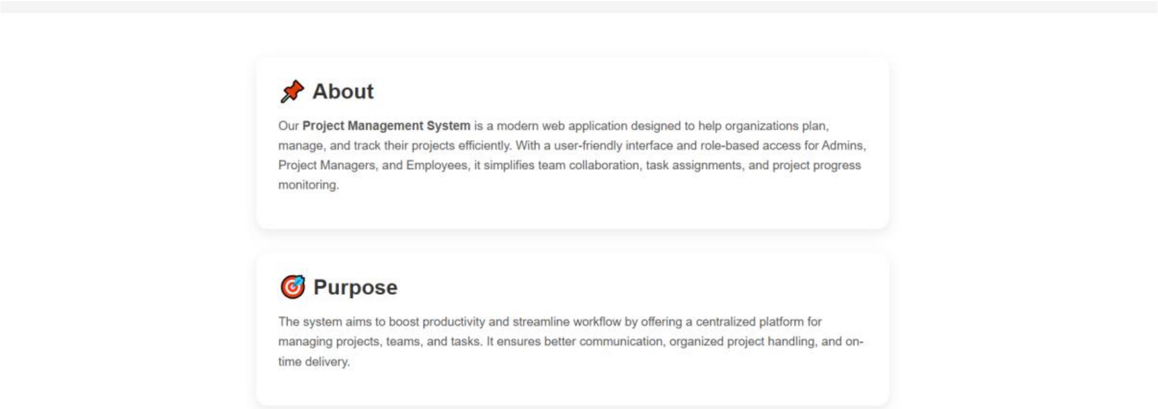
Contact Us

We'd love to hear from you. Fill out the form and we'll get back to you shortly.

Figure 5.18 Contact Page

5.19 About Page

The **About Page** of the Project Management System provides an overview of the platform's purpose, features, and benefits. It serves as an informative section for users and stakeholders to understand the core objectives and functionalities of the system



📌 About

Our **Project Management System** is a modern web application designed to help organizations plan, manage, and track their projects efficiently. With a user-friendly interface and role-based access for Admins, Project Managers, and Employees, it simplifies team collaboration, task assignments, and project progress monitoring.

🎯 Purpose

The system aims to boost productivity and streamline workflow by offering a centralized platform for managing projects, teams, and tasks. It ensures better communication, organized project handling, and on-time delivery.

Figure 5.19 About Page

6. Conclusion And Future Scope

6.1 Conclusion

The Project Management System developed using React.js, Node.js, and MySQL effectively streamlines the management of projects, tasks, teams, and user roles in an organizational environment. With a modern and responsive user interface, robust backend architecture, and secure database integration, the system provides seamless collaboration between administrators, project managers, and employees.

Throughout the development process, key functionalities such as task assignment, team management, progress tracking, and role-based access control were implemented to ensure a productive and secure workflow. The use of modern web technologies ensures scalability, performance, and ease of maintenance.

This system enhances operational efficiency by automating manual processes, reducing communication gaps, and improving task accountability among team members.

Future Scope

1 Real-time Notifications:
Implement WebSocket for real-time task updates, project changes, and team communication.

2 Gantt Chart Integration:
Visualize project timelines and task dependencies using Gantt charts for better project planning.

3 Analytics Dashboard:
Add data visualization tools to show progress metrics, team productivity, and workload distribution

4 One-to-One Messaging:
Enable private conversations between employees, project managers, and administrators.

7. References

1. React.js, "React.js Documentation." Available: https://react.dev/
2. HTML (Markup Language), "HTML Documentation." Available: https://www.w3schools.com/html/
3. CSS, "CSS Documentation." Available: https://www.w3schools.com/css/
4. Tailwind CSS, "Tailwind CSS Installation Guide." Available: https://tailwindcss.com/docs/installation
5. Material UI, "Material UI Documentation." Available: https://mui.com/material-ui/
6. React Router DOM, "React Router Documentation." Available: https://reacttraining.com/react-router
7. React Hooks, "React Hooks Introduction." Available: https://legacy.reactjs.org/docs/hooks-intro.html
8. JavaScript, "JavaScript Documentation." Available: https://www.javascript.com/
9. GitHub, "GitHub Platform." Available: https://github.com/index
10. Font Awesome, "Font Awesome Icons." Available: https://fontawesome.com/
11. Draft.js, "Draft.js Documentation." Available: https://draftjs.org/
12. React-Recharts, "React-Recharts Documentation." Available: https://recharts.org/
13. React Modal, "React Modal GitHub Repository." Available: https://github.com/reactjs/react-modal