

Project Management Software



By

Muhammad Azhar

Daniyal Ahmed

Department Of Computer Science

Govt Akhter Nawaz Khan (S) Degree College KTS Haripur

May, 2025

Project Management Software



By

Muhammad Azhar, Daniyal Ahmed

21421, 21443

Supervised By:

Mr. Syed Shahid Gillani

Department Of Computer Science

University Of Haripur

May, 2025

Project Management Software



By

Muhammad Azhar

Daniyal Ahmed

A Dissertation Submitted In Partial Fulfillment For Degree Of

BACHELORS

IN

COMPUTER SCIENCE

Department Of Computer Science

University Of Haripur

May 2025

Project Management Software

By

Muhammad Azhar, Daniyal Ahmed



*A thesis submitted in the partial fulfillment of the requirements for the degree of
bachelors in computer science*

COMMITTEE

1. External Examiner : _____
2. Internal Examiner: : _____
3. Supervisor: : _____
4. Head Of Department: : _____

Department Of Computer Science

University Of Haripur

May, 2025

Contents

CHAPTER 1	1
INTRODUCTION	1
1.1 Background of the Study	1
1.2 Problem Statement	1
1.3 Objectives of the Project	1
1.4 Scope of the Project	2
User registration and secure login	2
1.5 Significance of the Study	2
1.6 Project Deliverables	2
CHAPTER 2	3
LITERATURE REVIEW	3
2.1 Introduction	3
2.2 Project Management Tools and Systems	3
2.3 Task Assignment and Tracking	3
2.4 Team Collaboration and Communication	4
2.5 Authentication and Security	4
2.6 Existing Limitations in Current Systems	4
2.7 Contribution of the Proposed System	4
2.8 Summary	5
CHAPTER 3	6
SYSTEM DESIGN	6
3.1 Introduction	6
3.2 System Architecture	6
3.3 Frontend Design (React)	7
3.4 Data Flow Diagrams (DFDs)	10
3.4.1 Level 0 DFD (Context Diagram)	10
3.4.2 Level 1 DFD	11
3.4.3 Level 2 DFD	12
3.5 ERD Diagram	13
3.6 Class Diagram	14
3.7 Sequence Diagram	15
3.8 API Design	16

3.9 Security Design	16
3.10 Summary	16
CHAPTER 4	17
PROPOSED SYSTEM	17
4.1 Introduction	17
4.2 Software Development Methodology	17
4.2.1 Why Kanban?	17
4.3 System Development Tools	18
MongoDB:	18
Express.js.....	18
React.js.....	18
Node.js	18
Postman	19
MongoDB Compass.....	19
Visual Studio Code	19
GitHub	19
4.4 System Architecture	19
1. Presentation Layer (Frontend)	19
2. Application Layer (Backend):	19
3. Database Layer:.....	19
4.5 System Workflow	19
1. Requirement Gathering	20
2. System Design	20
3. Frontend Development.....	20
4. Backend Development.....	20
5. Integration	20
6. Testing and Debugging.....	20
7. Deployment.....	20
4.6 Functional Requirements	20
• User Authentication:.....	20
• Project Management:	20
• Task Management:.....	20
• File Upload:	20

• Read/Unread Message Tracking:	20
4.7 Non-Functional Requirements.....	20
• Usability:	20
• Reliability:.....	21
• Security:	21
• Performance:	21
• Scalability:	21
4.8 System Users.....	21
• Admin:.....	21
• Team Member:	21
CHAPTER 5.....	22
TESTING AND EVALUATION	22
5.1 Introduction.....	22
5.2 Types of Testing	22
5.2.1 Manual Testing through Browser	22
5.2.2 Basic Functional Testing.....	23
5.3 Bug Tracking and Fixes.....	24
5.4 Performance Evaluation.....	24
5.5 Tools Used for Testing	24
5.6 Summary.....	25
CHAPTER 6.....	26
USER MANUAL	26
6.1 Introduction.....	26
6.2 System Requirements	26
6.3 How to Use the System.....	27
6.3.1 Login.....	27
6.3.2 Dashboard	28
6.3.3 Create a Project	28
6.3.4 Add/Assign Tasks	29
6.3.5 Upload Files	30
6.3.6 Use Chat.....	31
6.5 Common Issues & Solutions.....	32
6.6 Logout	32

CHAPTER 7	33
CONCLUSION AND FUTURE WORK	33
7.1 Conclusion	33
6.2 Limitations.....	33
7.3 Future Work.....	34
7.4 Final Remarks	34

Declaration

I hereby declare that this dissertation is neither as a whole or part thereof has been copied out from any source. If any part of this thesis is proved to be copied or found to be a report of some other, we shall bear the consequences. No portion of this work presented in the thesis has been submitted in support of any application from any other degree of qualification or any other university or any other institute of learning.

I further declare that this research and all the associated documents with this, reports and all records are submitted as a partial requirement of the degree of Bachelor of Computer Science. We understand and transfer copyrights for these materials to University of Haripur and we will not sale this research and documents for getting any financial gains. It is further declared that I developed this and also this thesis is entirely on the basis of our personal efforts made under the sincere guidance of the project supervisor Mr. Syed Shaid Gillani Department of Computer Science, Govt Akhtar Nawaz Khan (S) Degree College KTS Haripur.

Muhammad Azhar: _____

Daniyal Ahmed: _____

Plagiarism Certificate

It is certified that the thesis has been checked for plagiarism. I have reviewed the content and found it to be original and in compliance with the university's plagiarism policy.

Supervisor: _____

DEDICATION

- I dedicate this thesis to my beloved parents, whose unwavering support, love, and prayers have been my source of strength throughout this journey, to my teachers, who provided me with guidance and support throughout my educational journey, and to all my friends and classmates, whose motivation and companionship made this experience unforgettable.

Acknowledgement

First and foremost, I am deeply thankful to Almighty Allah for giving me the strength, patience, and ability to complete this thesis successfully. Without His countless blessings, none of this would have been possible.

I would like to express my sincere gratitude to my respected supervisor Mr. Syed Shahid Gillani, for their valuable guidance, support, and encouragement throughout this Project. Their feedback and expertise helped me stay focused and improve the quality of my work.

I am also thankful to the faculty members of the computer Science Department, who provided me with the necessary knowledge and during my academic journey.

A special thanks to my parents, who have always stood by my side with their unconditional love, prayers, and sacrifices. Their support has been a constant source of motivation.

I would also like to thank my friends and classmates for their cooperation, moral support, and the great memories we created during this academic period.

Lastly, I acknowledge everyone who contributed directly or indirectly to this thesis.

Abstract

The rapid advancement of technology in recent years has significantly transformed the way organizations manage projects, collaborate with teams, and handle tasks. The increasing complexity of managing multiple teams and tasks in a streamlined manner has made it crucial for businesses to adopt more efficient project management solutions. This thesis focuses on the development of a **Project Management System (PMS)** designed to provide a comprehensive platform for managing projects, assigning tasks, and facilitating communication between team members.

The system was developed using the **MERN stack** (MongoDB, Express.js, React.js, and Node.js). The primary features of the system include project creation, team management (adding and assigning team members), task assignment, and the ability to upload files related to tasks. Additionally, a **chat feature** is integrated to enable real-time communication between team members, promoting collaboration and faster decision-making.

The system aims to simplify project handling, especially for small teams, academic groups, and startups.. The platform allows project managers to assign tasks to team members, set deadlines, and track the progress of each task, while also providing a centralized location for project-related documents and discussions.

CHAPTER 1

INTRODUCTION

1.1 Background of the Study

In the modern digital era, efficient management of projects and collaboration among teams has become a fundamental need in both academic and professional environments. Traditional project management approaches, such as paper-based or manual tracking systems, are no longer efficient for managing complex workflows, diverse teams, and real-time updates. With the increasing reliance on remote work and distributed teams, there is a growing demand for a centralized digital platform that allows seamless project tracking, task assignment, file sharing, and communication.

This project aims to build a comprehensive Project Management System using the MERN stack (MongoDB, Express.js, React.js, Node.js), which facilitates not only the creation and management of projects and tasks but also enables effective communication through real-time chat and file sharing.

1.2 Problem Statement

Project teams often struggle with disjointed communication, missed deadlines, and inefficient task tracking due to a lack of centralized tools. Existing project management systems either come with high subscription costs or lack flexibility and customization for specific organizational needs.

There is a need for a cost-effective, easy-to-use, and fully integrated solution that allows teams to manage tasks, communicate, and share files within a single application.

1.3 Objectives of the Project

- To develop a web-based application for managing projects and team collaboration.
- To implement user authentication features including login, signup, and password recovery.
- To allow users to create new projects and add team members.
- To enable creation, assignment, and tracking of tasks.

- To provide real-time chat functionality among team members.
- To allow file uploads for each task, facilitating information sharing.

1.4 Scope of the Project

The system is designed for small to medium-sized teams. It allows:

User registration and secure login.

- Password reset via email.
- Creation of multiple projects.
- Addition and management of team members.
- Task assignment with deadlines.
- File upload and management for tasks.
- Real-time chat with individual team members.

1.5 Significance of the Study

This project aims to provide an integrated platform that bridges the gap between task management and communication. It enhances productivity by reducing the need for multiple tools and increases accountability by keeping track of who is doing what and when. Educational institutions, startups, and SMEs can particularly benefit from this open-source, customizable system.

1.6 Project Deliverables

- A functional web-based application developed using the MERN stack.
- A user-friendly interface for managing tasks and chatting.
- Backend APIs for authentication, project, and task management.
- Real-time chat system using WebSockets (Socket.IO).
- Documentation including user manual and developer guide.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter presents a review of the literature relevant to the development of a **Project Management System**. It focuses on project management methodologies, team collaboration tools, task management systems, and real-time communication platforms. The purpose is to understand existing systems, their limitations, and how this project aims to address those gaps.

2.2 Project Management Tools and Systems

Project management tools are essential in ensuring tasks are planned, assigned, and completed within deadlines. Tools like **Asana**, **Trello**, and **Jira** are widely used for managing projects, especially in the software industry. These tools provide functionalities such as task creation, deadline assignment, progress tracking, and team collaboration.

However, many of these tools are either too complex for small teams or lack features like real-time chatting or file uploading integrated with each task. Furthermore, most tools require third-party integrations for communication and file sharing, leading to scattered workflows.

2.3 Task Assignment and Tracking

Task management is a core aspect of any project management system. Research shows that task clarity, accountability, and progress tracking improve productivity and reduce delays (Smith et al., 2019). Most modern systems allow users to assign tasks to team members, set priorities, and deadlines.

Despite this, many platforms do not support file uploads linked directly to tasks, making it difficult to manage task-specific documentation and submissions. This project bridges this gap by allowing team members to upload related files directly within the task.

2.4 Team Collaboration and Communication

Collaboration is a crucial part of project success. Tools like **Slack**, **Microsoft Teams**, and **Zoom** are commonly used, but they are not always integrated within project management tools. Studies (Johnson, 2020) show that switching between apps reduces focus and productivity.

Real-time chat integration inside the project management tool, as implemented in this system, helps team members communicate instantly without switching platforms. This enhances communication and keeps conversations relevant and task-focused.

2.5 Authentication and Security

User authentication is vital for protecting project data. According to a study by Cybersecurity Ventures (2021), poor authentication processes lead to 40% of data breaches in small to mid-sized platforms. Therefore, secure login, signup, and password recovery systems are essential.

This project incorporates secure user authentication using token-based sessions and proper validation. Password reset functionality is also included, offering a user-friendly and secure way to regain account access.

2.6 Existing Limitations in Current Systems

Although various tools offer partial solutions, there is a need for a system that combines all essential features in one platform — from task creation and team collaboration to file management and communication. Most current systems are either too limited or overly complex for small to medium-sized teams.

2.7 Contribution of the Proposed System

This project contributes by offering an all-in-one platform with features like:

- User authentication (Signup/Login/Forgot password)
- Project creation and team member assignment
- Task creation and assignment
- File uploading per task

- Real-time chat between team members

The integration of all these functionalities in one place improves workflow efficiency, team coordination, and project tracking.

2.8 Summary

This chapter reviewed the existing project management tools and their limitations, especially for small teams. It identified the gaps in current platforms and set the foundation for why this project is necessary. The next chapter will discuss the **System Analysis**, including requirements gathering and functional specifications.

CHAPTER 3

SYSTEM DESIGN

3.1 Introduction

This chapter discusses the design of the Project Management System. It covers the system architecture, database schema, component structure, and user interface (UI) design. The goal is to provide a clear understanding of how different components interact and how data flows through the system.

3.2 System Architecture

The system follows a **client-server architecture**. It consists of three main parts:

- **Frontend (Client Side):** Built with React and Vite. It handles user interactions, displays data, and sends requests to the backend.
- **Backend (Server Side):** Built with Node.js and Express.js. It handles business logic, API requests, and database operations.
- **Database:** MongoDB stores users, projects, tasks, and chat messages.

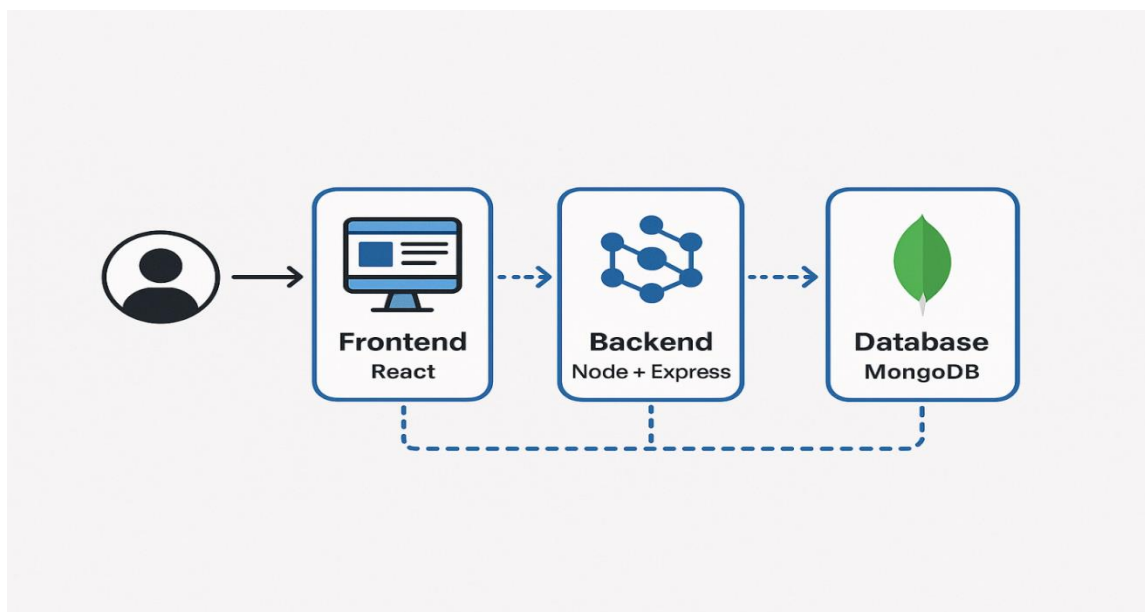
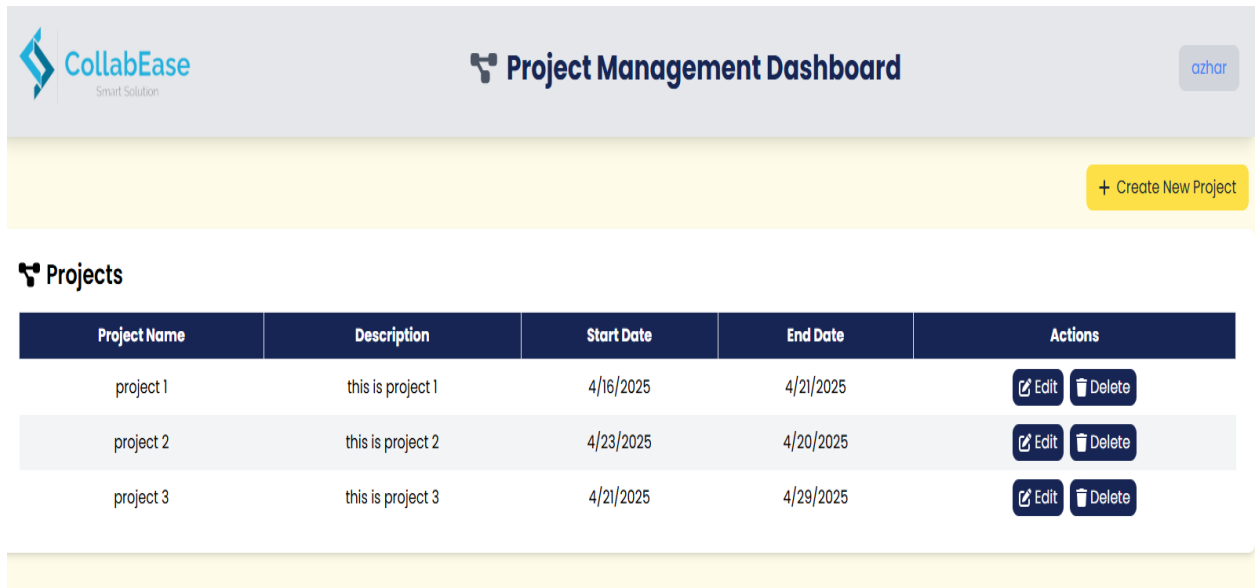


Figure 3. 1: System Architecture

3.3 Frontend Design (React)

The frontend is developed using React and is divided into reusable components. Key design elements include

- **Dashboard:** Displays projects



The screenshot displays the 'Project Management Dashboard' for 'CollabEase Smart Solution'. The dashboard includes a header with the logo, a user profile 'azhar', and a '+ Create New Project' button. Below the header, there is a 'Projects' section containing a table with three projects. Each project row has 'Edit' and 'Delete' action buttons.

Project Name	Description	Start Date	End Date	Actions
project 1	this is project 1	4/16/2025	4/21/2025	Edit Delete
project 2	this is project 2	4/23/2025	4/20/2025	Edit Delete
project 3	this is project 3	4/21/2025	4/29/2025	Edit Delete

Figure 3.1: Dashboard

- **Project details:** Represents project details.

Welcome, azhar

CollabEase
Smart Solution

Project Details

Teams

Project Details

+ Create New Task

Pending

Task Name	Description	Assign
task 3	this is task 3	+ Assign

In Progress

Task Name	Description	Assigned To
task 1	this is task 1	odli
task 2	this is task 2	habib

Review

Task Name	Description	Assigned To
-----------	-------------	-------------

Completed

Task Name	Description	Assigned To
-----------	-------------	-------------

Figure 3.3: Project Details

- **Task details:** Represents individual task details.

Task Details

Status: in-progress

Task Name: task 2

Task Description: this is task 2

Comments

Write a comment...

Post Comment

Uploaded Files

No files uploaded yet.

Choose File

No file chosen

Upload

Team Members

No team members assigned yet.

Move Task To... ▼

Delete Task

Figure 3.4: Task Details

3.4 Data Flow Diagrams (DFDs)

DFDs show how data moves through the Project Management Software. They help visualize how users (Project Manager and Team Members) interact with the system.

3.4.1 Level 0 DFD (Context Diagram)

This gives a high-level view of the system. It shows the main users (Project Manager and Team Member) and how they exchange data with the system.

- Project Manager: creates projects, adds team members, assigns tasks
- Team Member: completes tasks, uploads files, and communicates

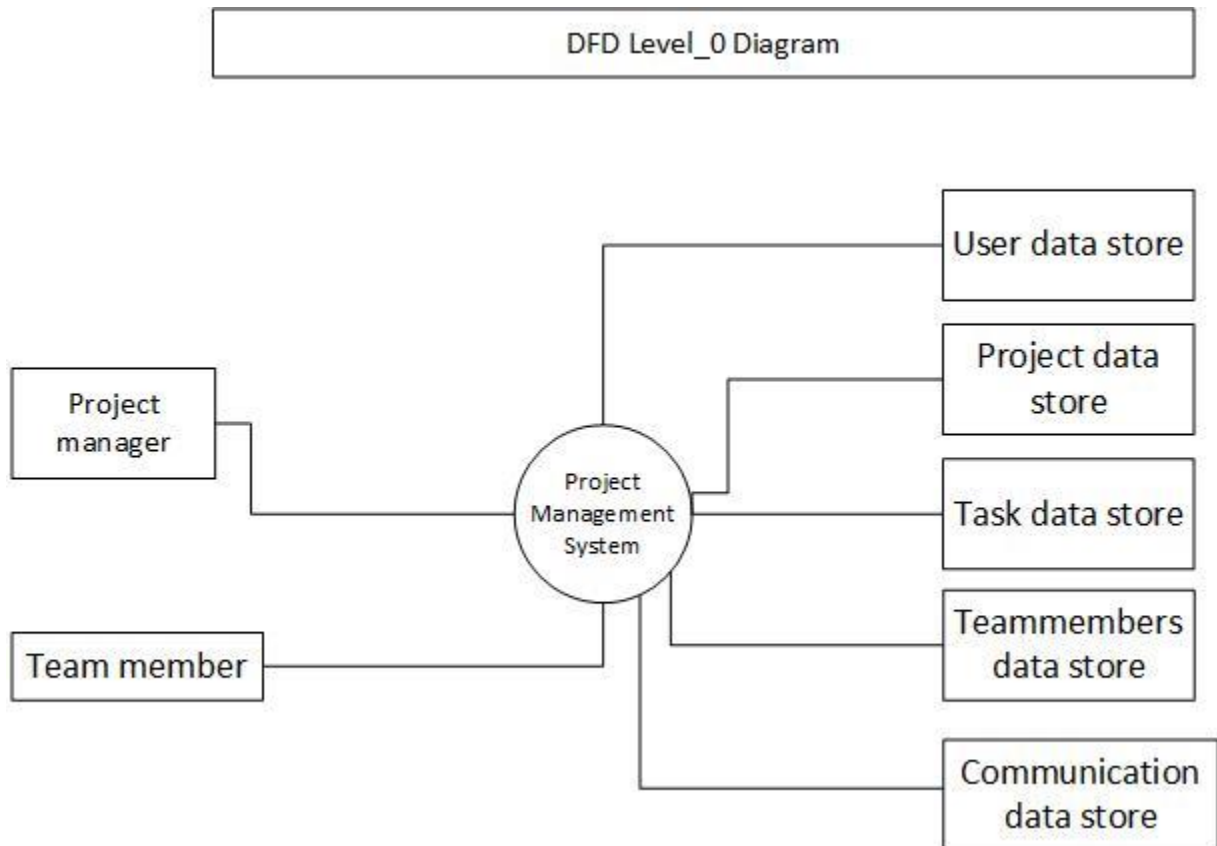


Figure 3.5: DFD Level 0 — Context Diagram of the Project Management System

3.4.2 Level 1 DFD

This breaks the system into key processes:

- Create Project
- Assign Task
- Upload File
- Chat with Team Members

It shows how each function connects to users and data storage.

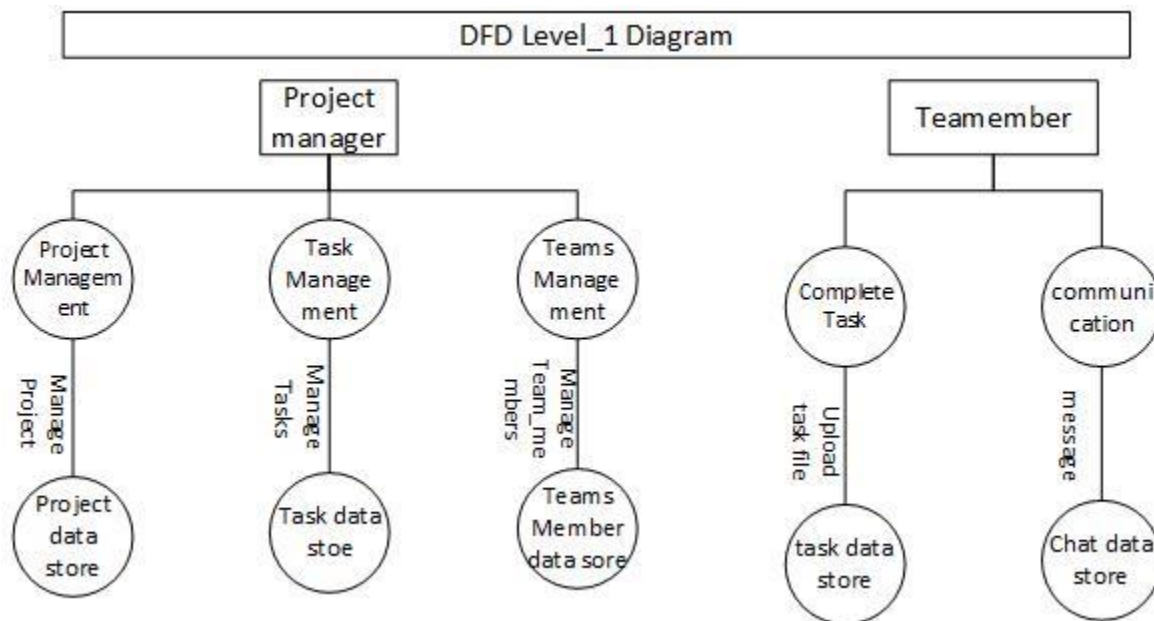


Figure 3.5: DFD Level 1 — High-Level Data Flow of the Project Management System

3.4.3 Level 2 DFD

This focuses on the “Assign Task” process in more detail:

- Select task details
- Choose team members
- Send assignments
- Receive task updates

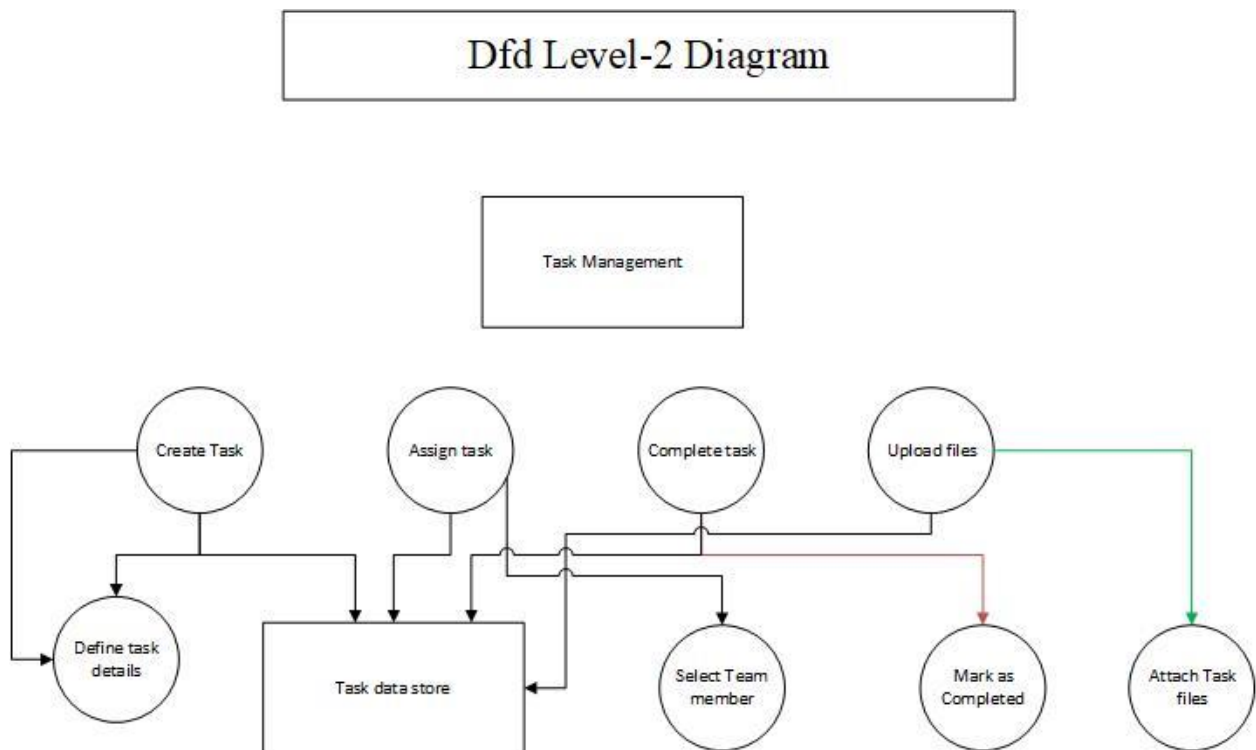


Figure 3.6: DFD Level 2 — Detailed Data Flow of Core Modules

3.5 ERD Diagram

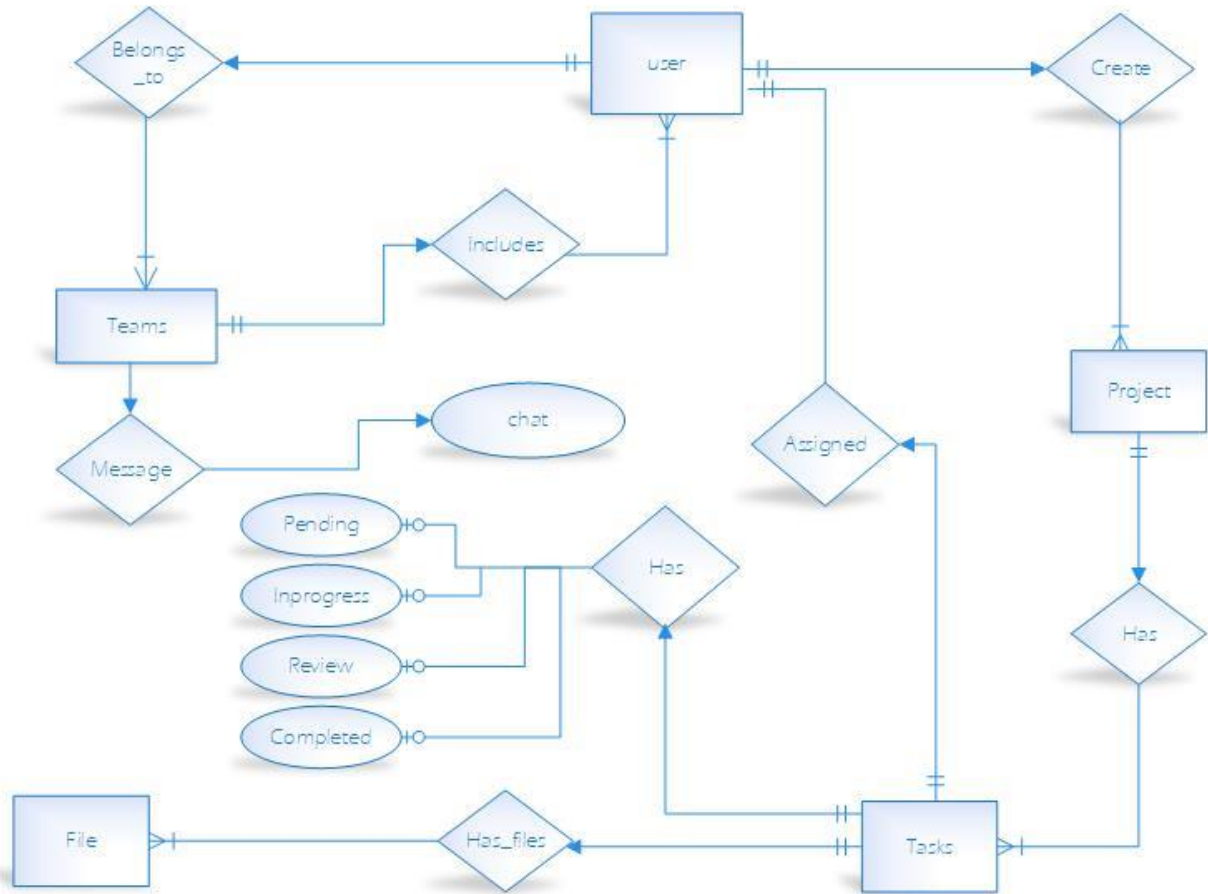


Figure 3.7: Entity-Relationship Diagram of the Project Management System

3.6 Class Diagram

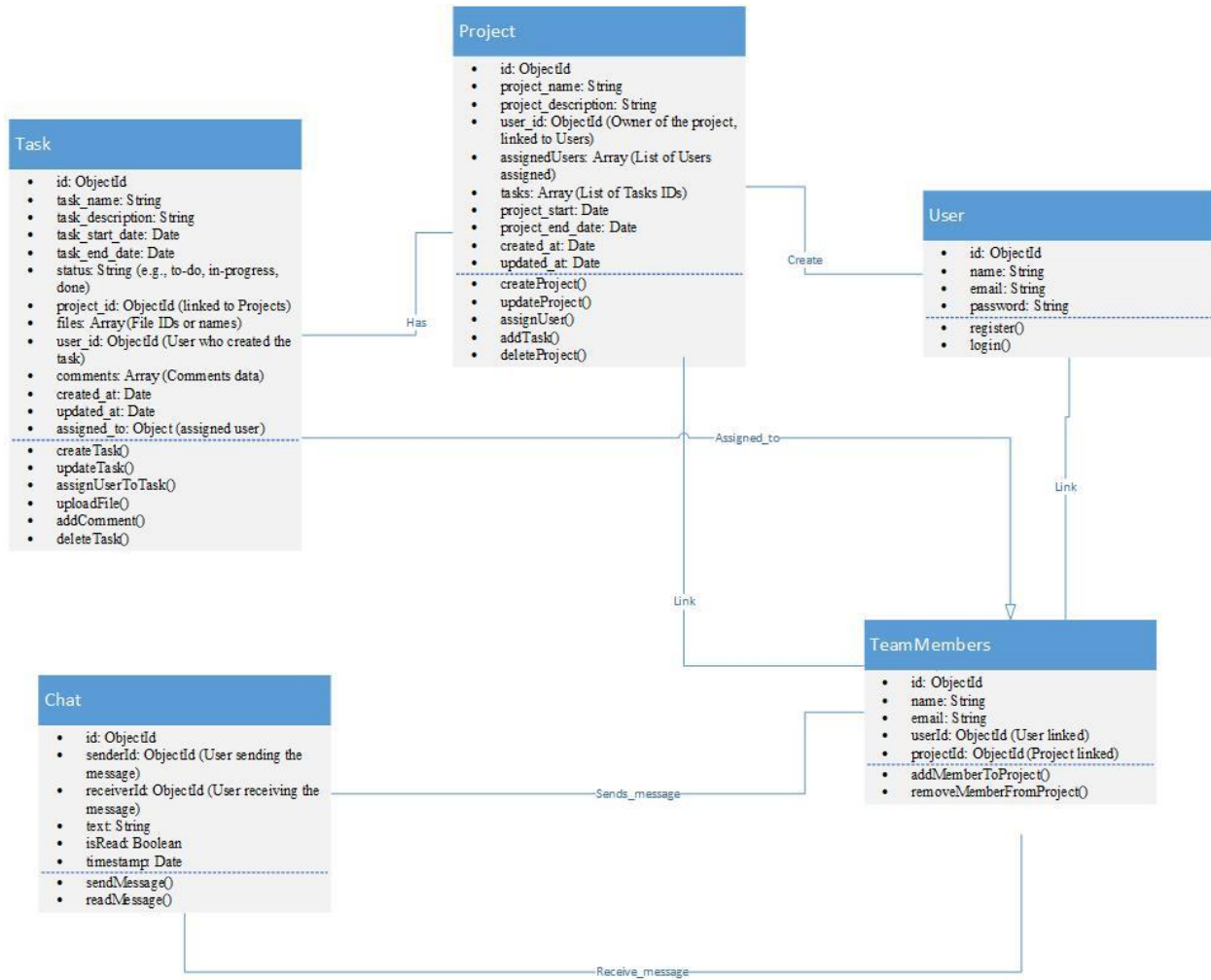


Figure 3.6: UML Class Diagram of the Project Management System

3.7 Sequence Diagram

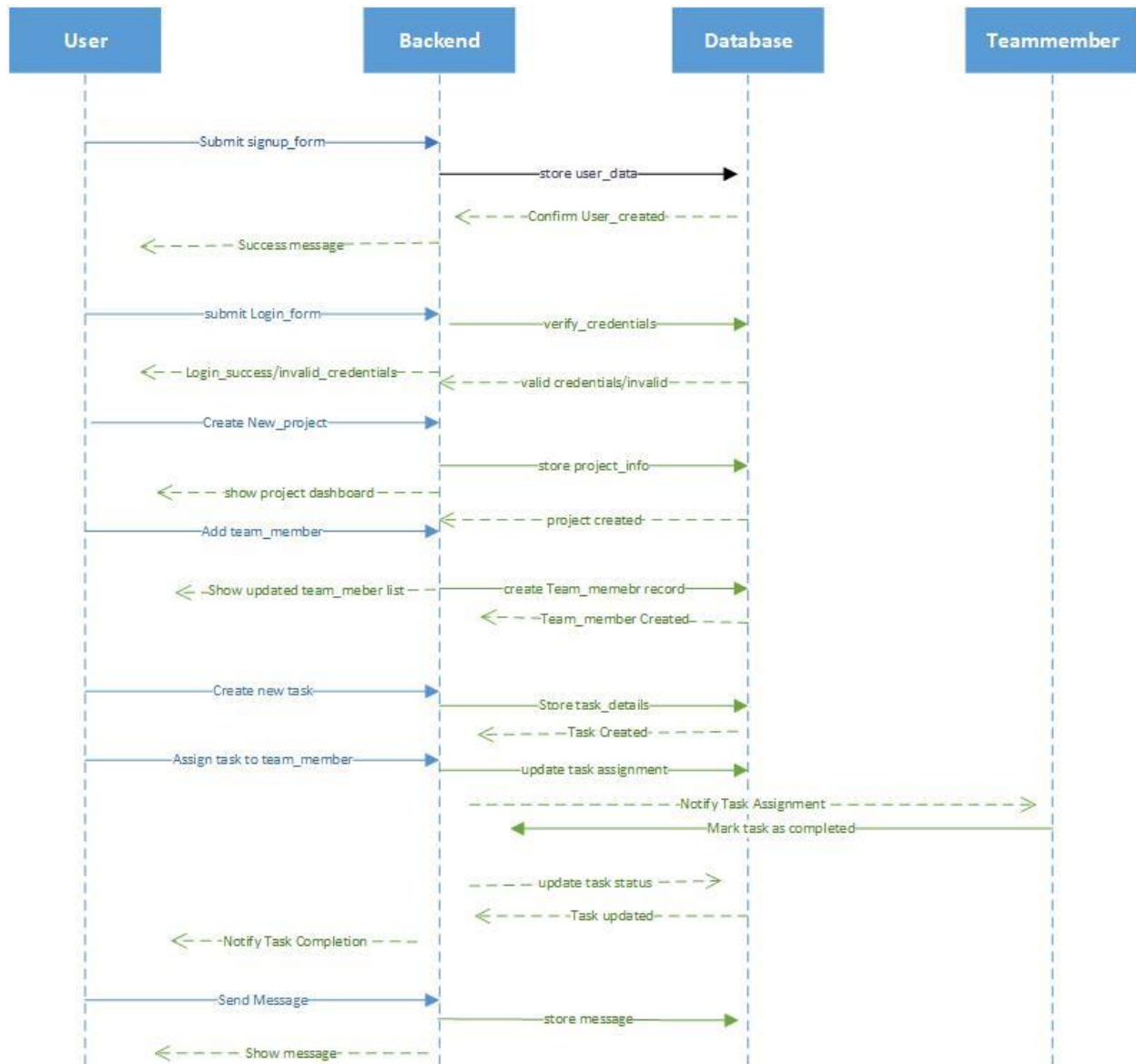


Figure 3.7: UML Sequence Diagram of the Project Management System

3.8 API Design

The backend exposes RESTful APIs:

- **Auth APIs:** /api/auth/login, /api/auth/register, /api/auth/forgot-password
- **Project APIs:** /api/projects/create, /api/projects/:id
- **Task APIs:** /api/tasks/create, /api/tasks/:id
- **Message APIs:** /api/messages/send, /api/messages/get/:userId

3.9 Security Design

Security features include:

- Password hashing using **bcrypt**
- **JWT** for session handling and authorization
- Validation of inputs on both frontend and backend
- Restriction of access based on user roles (admin/member)

3.10 Summary

This chapter explained the technical design of the system including its architecture, UI, database schema, and APIs. In the next chapter, we will go through the **Implementation and Coding** details of the system.

CHAPTER 4

PROPOSED SYSTEM

4.1 Introduction

This chapter outlines the methodology adopted during the development of the proposed system. It includes the software development model used, tools and technologies, system architecture, and the overall workflow followed to build the system. The selected methodology ensures that the system is developed in a structured, efficient, and organized manner.

4.2 Software Development Methodology

The system was developed using the **Kanban methodology**, which is a flexible and visual method for managing tasks and tracking project progress. Kanban helps in maintaining a continuous flow of work and supports incremental system development.

4.2.1 Why Kanban?

- It allows flexibility and adaptation during development.
- Visual boards help track task status (To Do, In Progress, Done).
- Suitable for projects where tasks can be continuously added or modified.

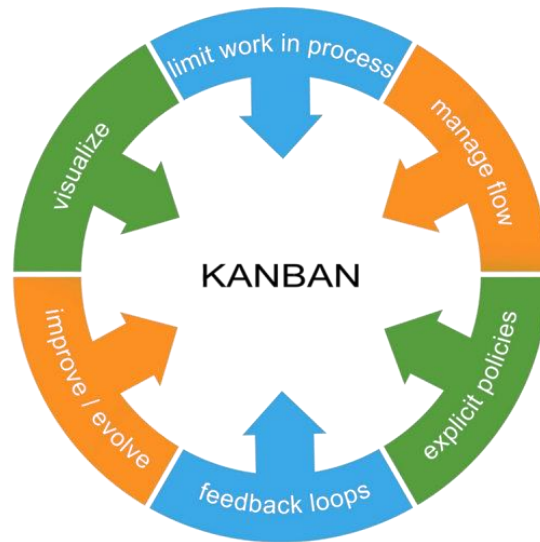


Figure 4.1: Kanban Board

4.3 System Development Tools

The following tools and technologies were used in the development of the system:

MongoDB:

MongoDB is a NoSQL database used to store project data, tasks, users, and files in a flexible, document-oriented structure. It handles large volumes of unstructured data efficiently.

Express.js

Express.js is a lightweight backend framework for Node.js. It simplifies server-side development and is used to build APIs and handle HTTP requests.

React.js

React.js is a JavaScript library used on the frontend to build fast, dynamic, and responsive user interfaces. It helps create reusable components for the web app.

Node.js

Node.js is a JavaScript runtime environment that runs server-side code. It powers the backend and allows using JavaScript for both client and server.

Postman

Postman is a tool for testing APIs. It helps developers send requests to endpoints, check responses, and debug backend functionality.

MongoDB Compass

MongoDB Compass is a GUI tool for interacting with MongoDB databases. It allows visualizing collections, running queries, and managing records without writing code.

Visual Studio Code

Visual Studio Code is a powerful and lightweight code editor. It supports many extensions and features for easier coding, debugging, and project management.

GitHub

GitHub is a platform used for version control and collaboration. It allows storing code, tracking changes, and working with teams on the same project.

4.4 System Architecture

The system follows a **three-tier architecture**:

1. **Presentation Layer (Frontend):**

Developed using React.js. Handles the user interface and interaction.

2. **Application Layer (Backend):**

Developed using Node.js and Express.js. Handles routing, logic, and APIs.

3. **Database Layer:**

Managed using MongoDB. Stores data such as users, tasks, projects, and comments.

4.5 System Workflow

The development process followed these steps:

1. **Requirement Gathering**

Understanding user needs and defining functional and non-functional requirements.

2. **System Design**

Creating wireframes, database schema, and architecture diagrams.

3. **Frontend Development**

Building interactive components using React and Tailwind CSS.

4. **Backend Development**

Creating RESTful APIs, handling authentication, and data storage using MongoDB.

5. **Integration**

Connecting frontend and backend using Axios for HTTP requests.

6. **Testing and Debugging**

Using Postman for API testing and browser dev tools for UI testing.

7. **Deployment**

Hosting the system (MongoDB Atlas for database).

4.6 Functional Requirements

Functional requirements describe what the system should do. Key functional requirements of this system include:

- **User Authentication:** Users can register, login, and reset passwords.
- **Project Management:** Admin users can create new projects and add team members.
- **Task Management:** Admin can create tasks and assign them to specific members.
- **File Upload:** Team members can upload files as task deliverables.
- **Chat System:** Team members can chat in real time.
- **Read/Unread Message Tracking:** Users can see which messages are unread.

4.7 Non-Functional Requirements

Non-functional requirements define how the system performs:

- **Usability:** The interface should be easy to use for all users.

- **Reliability:** The system should function correctly without crashes.
- **Security:** User data and files must be protected using secure authentication and file storage practices.
- **Performance:** Real-time chat and file upload should work smoothly with minimal delay.
- **Scalability:** The system should support multiple users and tasks as the team grows.

4.8 System Users

There are two main types of users in the system:

- **Admin:**
 - Can create projects.
 - Can add team members to a project.
 - Can create and assign tasks.
- **Team Member:**
 - Can view assigned tasks.
 - Can upload files to tasks.
 - Can chat with team members.

CHAPTER 5

TESTING AND EVALUATION

5.1 Introduction

This chapter discusses the testing approaches used to ensure that the Project Management System works correctly and meets user expectations. Since the main focus was on developing the system and checking it manually through the browser, the testing involved real-time interaction with the application to verify its functionality, fix errors, and validate user experience.

5.2 Types of Testing

5.2.1 Manual Testing through Browser

Most of the testing was carried out manually by running the project in the browser. The system was interacted with as an end-user would — creating projects, adding team member assigning tasks, sending messages, and uploading files — to confirm that each feature worked correctly.

Key areas tested manually:

Component	Description	Status
Login Page	Verified login with correct and incorrect credentials	Passed
Registration Page	Tested new user registration functionality	Passed
Project Management	Created new projects and edited existing ones	Passed
Task Assignment	Assigned tasks to users and checked assignment accuracy	Passed

Chat System	Sent and received real-time messages	Passed
File Upload	Uploaded and downloaded files successfully	Passed

5.2.2 Basic Functional Testing

Basic functional testing was performed by manually trying out each major feature. This included checking the flow from login to dashboard, creating and managing projects, assigning tasks, chatting with users, and handling file uploads.

Feature	Test Scenario	Result
Project Creation	Admin created a project with all details	Success
Task Assignment	Assigned task to team members	Success
Chat Feature	Real-time chat messages appeared correctly	Success
File Upload	Files uploaded and accessible for download	Success
Forgot Password Flow	Password reset link and token tested	Success

5.3 Bug Tracking and Fixes

While testing manually, a few issues were noticed and fixed immediately:

Bug Found	Fix Applied
Chat messages not updating	Refreshed chat component state after sending
File upload failing for empty files	Added validation to prevent empty file uploads
Dashboard not updating instantly	Added page refresh and API calls after changes

5.4 Performance Evaluation

The system was tested informally for performance by observing its behavior during manual use:

- **Load Time:** Pages loaded within 2–3 seconds on average.
- **Real-time Chat:** Messages appeared almost instantly (under 1 second).
- **Database Response:** Data retrieved quickly without noticeable delay.
- **Mobile and Desktop View:** Application was responsive and worked well on different screen sizes.

5.5 Tools Used for Testing

- **Browser (Chrome/Firefox):** Used for live testing and checking UI/UX.
- **Developer Tools:** Inspected API calls, console errors, and network performance.
- **Postman:** Used occasionally for checking API endpoints separately.
- **Console Logs:** Monitored application flow and debugged issues during development.

5.6 Summary

This chapter explained the different testing approaches and evaluations used during development. The Project Management System was thoroughly tested, and all major bugs were resolved. The system performs well under normal usage conditions and meets the functional requirements effectively.

CHAPTER 6

USER MANUAL

6.1 Introduction

This user manual provides step-by-step guidance for end-users to use the Project Management Software. The goal is to help users easily navigate, create projects, assign tasks, upload files, and communicate through the chat feature.

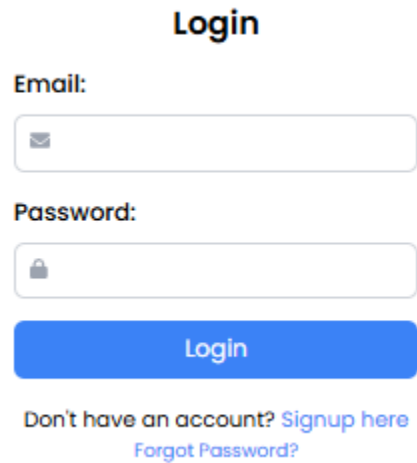
6.2 System Requirements

- **Browser:** Google Chrome / Mozilla Firefox
- **Device:** PC, Laptop, or Mobile (responsive)
- **Internet:** Required
- **Login Credentials:** Provided by the administrator

6.3 How to Use the System

6.3.1 Login

- Enter your email and password.
- Click "Login" to access the dashboard.



The login form is centered on the page. It features a title 'Login' in bold black text. Below the title, there are two input fields: 'Email:' with an envelope icon and 'Password:' with a lock icon. A blue 'Login' button is positioned below the password field. At the bottom, there are two links: 'Don't have an account? Signup here' and 'Forgot Password?'.

Login

Email:

Password:

Login

Don't have an account? [Signup here](#)

[Forgot Password?](#)

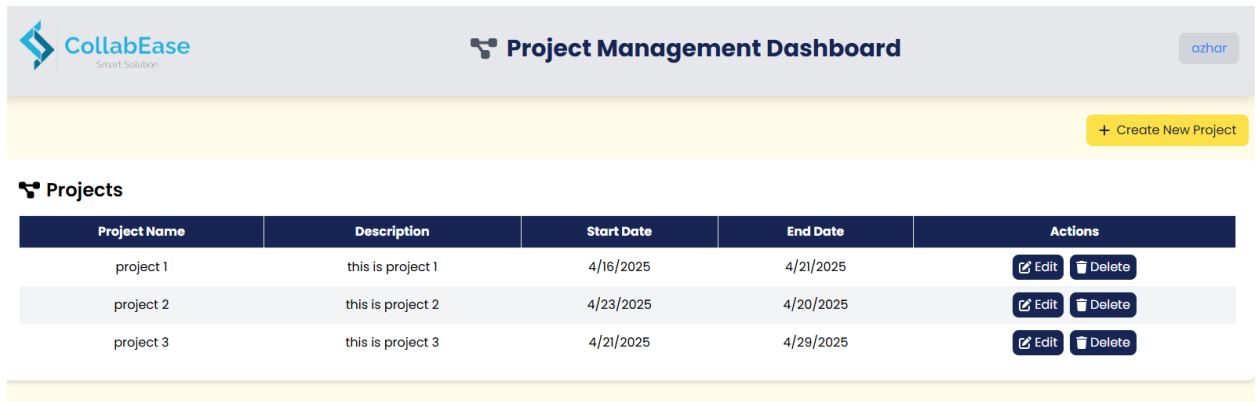
Figure 6.3.1: Login Page

6.3.2 Dashboard

- View all your projects, assigned tasks, task status, chat notifications.

6.3.3 Create a Project

- Click on "Create Project" button
- Fill in project name, description, and deadline



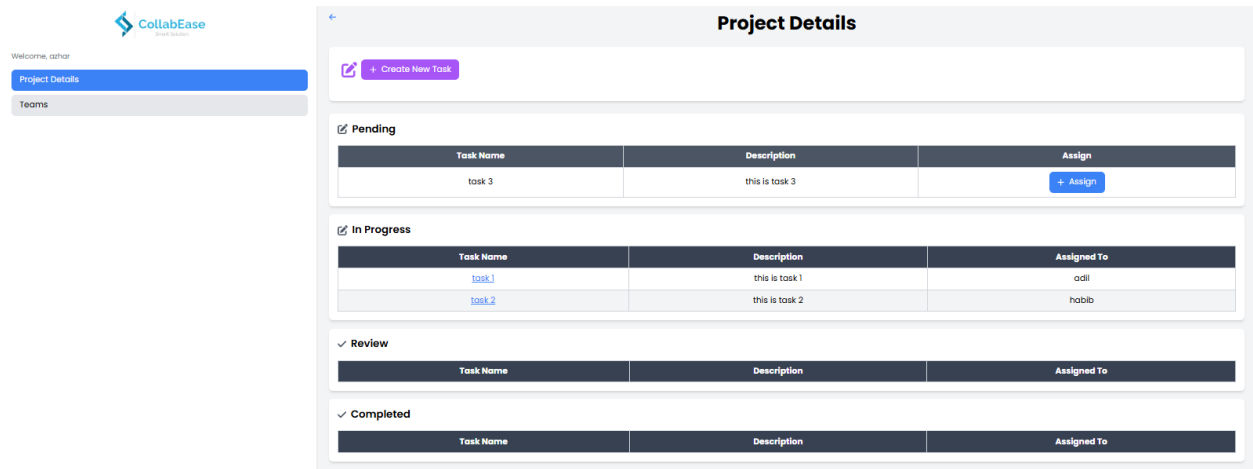
The screenshot displays the CollabEase Project Management Dashboard. At the top, the CollabEase logo is on the left, the dashboard title is in the center, and the user name 'azhar' is on the right. A yellow banner below the header features a '+ Create New Project' button. The main section, titled 'Projects', contains a table with the following data:

Project Name	Description	Start Date	End Date	Actions
project 1	this is project 1	4/16/2025	4/21/2025	Edit Delete
project 2	this is project 2	4/23/2025	4/20/2025	Edit Delete
project 3	this is project 3	4/21/2025	4/29/2025	Edit Delete

Figure 6.3.2: First screen after login

6.3.4 Add/Assign Tasks

- Open any project
- Click "Create new task"
- Fill in task details,
- Assign it to a teammember



The screenshot displays the CollabEase application interface. On the left, a sidebar shows the user's name 'Welcome, oshar' and navigation links for 'Project Details' (highlighted) and 'Teams'. The main content area is titled 'Project Details' and features a '+ Create New Task' button. Below this, tasks are organized into four status sections: Pending, In Progress, Review, and Completed. Each section contains a table with columns for Task Name, Description, and Assign/Assigned To.

Pending

Task Name	Description	Assign
task 3	this is task 3	+ Assign

In Progress

Task Name	Description	Assigned To
task 1	this is task 1	odil
task 2	this is task 2	habib

Review

Task Name	Description	Assigned To
-----------	-------------	-------------

Completed

Task Name	Description	Assigned To
-----------	-------------	-------------

Figure 6.3.3: Login Page

6.3.5 Upload Files


- Go to a task
- Click "Upload File"
- Select a file from your system

Task Details

Status: in-progress


Task Name: task 2

Task Description: this is task 2

 **Comments**

Write a comment...

Post Comment

 **Uploaded Files**

No files uploaded yet.

Choose File

No file chosen

Upload

Team Members

No team members assigned yet.

Move Task To... ▼

Delete Task

Figure 6.3.4: Upload task files

6.3.6 Use Chat

- Open the “Team member” tab within a project
- Select team member for chat
- Type a message and press save button to send message

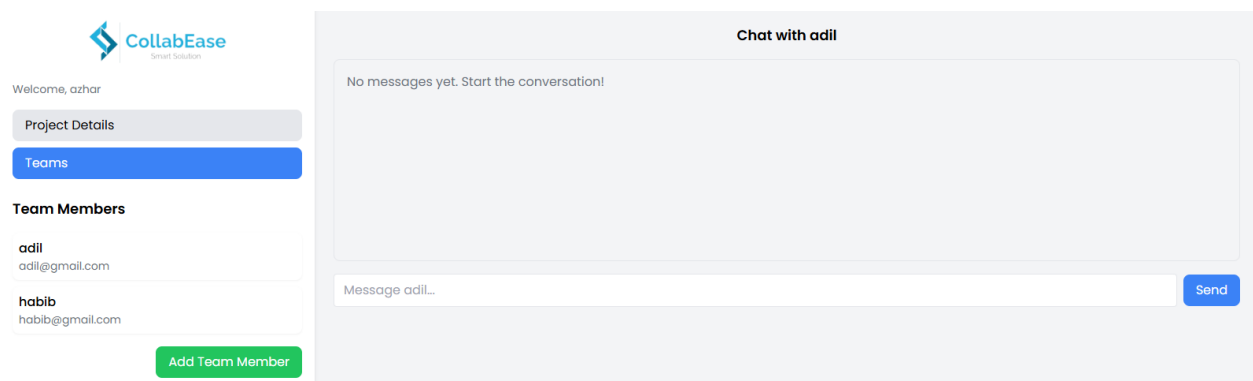


Figure 6.3.5: chat with team members

6.5 Common Issues & Solutions

Issues	Solution
Cannot login	Check credentials or reset password
Task not saving	Ensure all required fields are filled
Chat not working	Check internet connection or reload page

6.6 Logout

Click the profile and select “Logout” to safely exit the system.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

This thesis presented the design and development of a web-based **Project Management System** that enables project creation, task management, team collaboration, file sharing, and real-time communication. The system was developed using modern web technologies including the MERN stack (MongoDB, Express.js, React.js, Node.js) and Socket.IO for real-time features.

Key functionalities implemented include:

- **User Authentication:** Secure login, registration, and password reset.
- **Project Management:** Admins can create projects and add team members.
- **Task Management:** Tasks can be created, assigned to team members, and updated.
- **File Upload:** Users can upload files related to specific tasks.
- **Real-Time Chat:** Team members can communicate instantly.
- **Notification System:** Chat notifications and unread message indicators improve communication.

All features were tested thoroughly and the system demonstrated good performance, usability, and reliability. The interface is responsive and user-friendly, making it suitable for student groups, small teams, and educational environments.

This project highlights the importance of structured project management and effective team communication in modern software development practices.

6.2 Limitations

While the system achieves its primary objectives, some limitations were observed:

- Lack of push notifications .
- Limited file type support in the upload module.
- No visual analytics or reporting features for projects/tasks.

- Requires manual deployment and lacks CI/CD integration.

7.3 Future Work

To enhance the Project Management System, several features can be added in future versions:

1. **Mobile App:** Develop a cross-platform mobile application using React Native or Flutter for on-the-go access.
2. **Notification System:** Implement push notifications for updates and deadlines.
3. **Task Progress Visualization:** Add Kanban boards, Gantt charts, and pie charts for better visualization of project status.
4. **Calendar Integration:** Allow users to view tasks and deadlines in a calendar view.
5. **Voice and Video Calls:** Enable more effective remote communication within the team.
6. **Activity Logs:** Maintain logs of user actions for better accountability and tracking.
7. **AI-Powered Recommendations:** Suggest task assignments or priorities using AI.
8. **Dark Mode and Theme Customization:** Enhance user experience with UI personalization.

7.4 Final Remarks

The development of this system not only fulfilled academic goals but also provided practical experience in full-stack development, teamwork, and software engineering practices. It serves as a strong foundation for building more advanced management tools and demonstrates the potential of using real-time web technologies for productivity and collaboration.

