Deliverable E – SE D3

Final Report

ProjectPulse

Group Members:

Tauha Imran	22i-1239,
Minahil Ali	22i-0849,
Nabeed Haider	22i-0871

Project Title: **Project Pulse**

Submission Date: 28th April, 2025

Contents

1. Project Introduction

ProjectPulse is a MERN stack-based Task Management System developed to streamline task delegation, team coordination, and project monitoring. It offers a comprehensive dashboard where users can manage teams, tasks, schedules, timelines, documents, and track overall project health efficiently. The system emphasizes usability, real-time notifications, and effective collaboration to enhance productivity for small to medium teams.

2. Functional and Non-Functional Requirements

2.1 List of functional requirements.

1. Team Management

- The system shall allow users to **create**, **edit**, and **delete teams**.
- The system shall allow users to **manage team memberships** (adding and removing members).
- The system shall implement **Role-Based Access Control** with two roles:
  - **Admin**
  - **Member**

2. Task Management

- The system shall allow users to **create**, **edit**, and **delete tasks**.
- The system shall allow users to **assign tasks** to specific users or teams.
- The system shall allow users to **set task priority**:
  - High
  - Medium
  - Low
- The system shall allow users to **set task status/stage**:
  - Todo
  - In Progress
  - Completed
  - On Hold
- The system shall allow users to **set due dates and deadlines** for tasks.
- The system shall allow users to **add task descriptions and attachments**.

3. Project Creation

- The system shall enable users to **create new projects** by specifying project details and initial settings.

4. Task Commenting

- The system shall allow **Admins** to **add**, **edit**, and **delete comments** on tasks.

5. Recurring Task Scheduling

- The system shall allow users to **schedule tasks to recur** at specified intervals:
    - Daily
    - Weekly
    - Monthly

6. User Profile Management

- The system shall allow users to **update their profile information**, including:
    - Name
    - Email
    - Profile Picture

7. User Addition

- The system shall allow **Administrators** to **add new users** by specifying:
    - User roles
    - User credentials

8. Project Organization

- The system shall support **project categorization** through:
    - Tags
- The system shall allow **creation of subtasks/checklists** inside a project.

9. Project Timeline Updates

- The system shall allow users to **update and adjust project timelines**, including milestones and deadlines.

10. Project Status Analysis

- The system shall provide a **dashboard** that:
    - Analyzes and displays the **current status**, **progress** of the projects.

11. File Uploading on Tasks

- The system shall allow users to **upload and attach file links** to specific tasks.

12. Notification System

- The system shall send **real-time notifications** to users regarding:
    - Task updates
    - Project changes
    - New comments

2.2 List of non-functional requirements.

1. Performance and Responsiveness

- The system shall provide **fast API response times**, aiming for **under 200ms** for core actions like login, task creation, and updates.
- The client interface shall use **Vite** for optimized bundling and **TailwindCSS** for efficient rendering, ensuring **fast load times** across devices.
- The system shall be **responsive**, delivering a seamless experience on **both mobile and desktop** screens.

2. Security and Data Integrity

- The system shall implement **secure authentication** using **JWT (JSON Web Tokens)**.
- Sensitive information (such as API keys, database URLs) shall be secured via **environment variables (.env)**.
- The **MongoDB database** shall be **securely hosted**, ensuring protection against unauthorized access and data breaches.
- Input validation and sanitization shall be enforced to maintain **data integrity** and prevent injection attacks.

3. Scalability, Reliability, and Availability

- The system shall be designed for **horizontal scalability**, allowing smooth growth in the number of users, teams, and projects without impacting performance.
- **High availability** shall be ensured, with the system able to **handle concurrent operations** without downtime or data corruption.
- The backend shall be modular (controllers/, routes/, models/) to support future expansions (e.g., adding new modules like analytics, reporting).

4. Maintainability and Version Control

- The system shall follow a **modular architecture** separating frontend (components/, pages/, redux/) and backend (controllers/, middleware/, routes/) for better maintainability.
- **Version control** shall be maintained through **GitHub**, allowing collaborative development, code review, and tracking of changes.

5. Usability and User Experience

- The application shall offer an **intuitive, user-friendly UI** with consistent interaction patterns (reusable components like Button, Table, ModalWrapper, NotificationPanel).
- Features such as **real-time notifications**, **task filtering**, **user avatars**, and **dashboard charts** shall enhance overall **ease of use**.

6. Backup, Recovery, and Fault Tolerance

- The system shall be prepared for **database backups** and shall aim for **fault tolerance**, ensuring minimal disruption in case of server crashes or data loss.

- Key data like tasks, users, and notifications shall be recoverable with minimal downtime.

3. User Stories

ID	User Story	Acceptance Criteria
1	As a Manager, I want to create new teams so that I can group users under projects.	Team created successfully and members assigned.
2	As a Manager, I want to manage tasks by assigning users and setting deadlines.	Tasks visible under assigned users.
3	As a User, I want to create new projects to organize work better.	New project appears in dashboard.
4	As a User, I want to update my profile information.	Updated details reflected immediately.
5	As an Admin, I want to add a new user to the system.	New user receives login credentials via email.
6	As a Manager, I want to assign tags to tasks.	Tags appear under task listings.
7	As a User, I want to comment on tasks to communicate with teammates.	Comment appears instantly under task.
8	As a User, I want to set the priority level for tasks.	Priority levels visible in task list.
9	As a User, I want to schedule recurring tasks for routine activities.	Recurrence schedule generates next tasks automatically.
10	As a Manager, I want to update the project timeline when changes occur.	Updated timeline visible on Gantt Chart.
11	As a User, I want to upload files related to tasks.	Files downloadable from task details page.
12	As a Manager, I want to track project status on the dashboard.	Progress bars and analytics charts update live.
13	As a User, I want to receive notifications when a task is updated.	In-app notification pops up immediately.

14	As a Manager, I want to analyze task priority charts.	Chart visualization available in dashboard.
15	As an Admin, I want all changes logged securely for audits.	Audit logs are retrievable by Admin.

4. Product Backlog

Feature	**Priority** **(1-High, 2-Medium, 3-Low)**
Manage teams	1
Manage tasks	1
Create projects	1
Update profile	2
Add new user	2
Assign tags to tasks	2
Set task priority	2
Comment on tasks	2
Schedule recurring tasks	2
Track priority chart	3
Upload files on tasks	2
Update project timeline	3
Analyze project status dashboard	3
Receive notifications	2

5. Sprint 1 and Sprint 2 Backlogs

5.1 Sprint 1: Core Features (Completed by March 11, 2025)

- Manage Teams
- Manage Tasks
- Create Projects

- Set Task Priority
- Update Profile

Goal: Build the foundational components of the system including core entity management.

User Story	Feature	Estimated Time	Assigned To
US-1	Manage Teams	2 days	Tauha Imran
US-2	Manage Tasks	3 days	Nabeed Haider
US-3	Create Projects	2 days	Minahil Ali
US-8	Set Task Priority	1 day	Nabeed Haider
US-4	Update Profile	2 days	Tauha Imran

5.2 Sprint 2: Intermediate Features (Completed by March 23, 2025)

Major Tasks:

- Assign Tasks
- Comment on Tasks
- Schedule Recurring Tasks
- Add New User
- Upload Files on Tasks

Goal: Add user collaboration and extended task management capabilities.

User Story	Feature	Estimated Time	Assigned To
US-2	Assign Tasks	1.5 days	Nabeed Haider
US-7	Comment on Tasks	2 days	Minahil Ali
US-9	Schedule Recurring Tasks	2 days	Tauha Imran
US-5	Add New User	1.5 days	Nabeed Haider
US-11	Upload Files on Tasks	2 days	Minahil Ali

6. Project Plan

6.1 Work Breakdown Structure (WBS)

6.1.1 WBS Overview

The Work Breakdown Structure (WBS) is a foundation for project execution by dividing the Task Management System into manageable phases and tasks. It outlines a hierarchical structure that breaks down the entire project scope into defined activities, helping the team manage complexity, monitor progress, and allocate resources efficiently.

This WBS is designed around six key phases:

1. Initial project planning and setup
2. Three development sprints (D1, D2, D3) aligned with grouped use cases
3. Final testing and deployment
4. Comprehensive documentation

Each sprint focuses on a specific set of use cases to ensure modular development and continuous integration. Dependencies are clearly identified to ensure sequential flow and avoid bottlenecks.

6.1.2 Hierarchical WBS Diagram

A visual format showing project decomposition

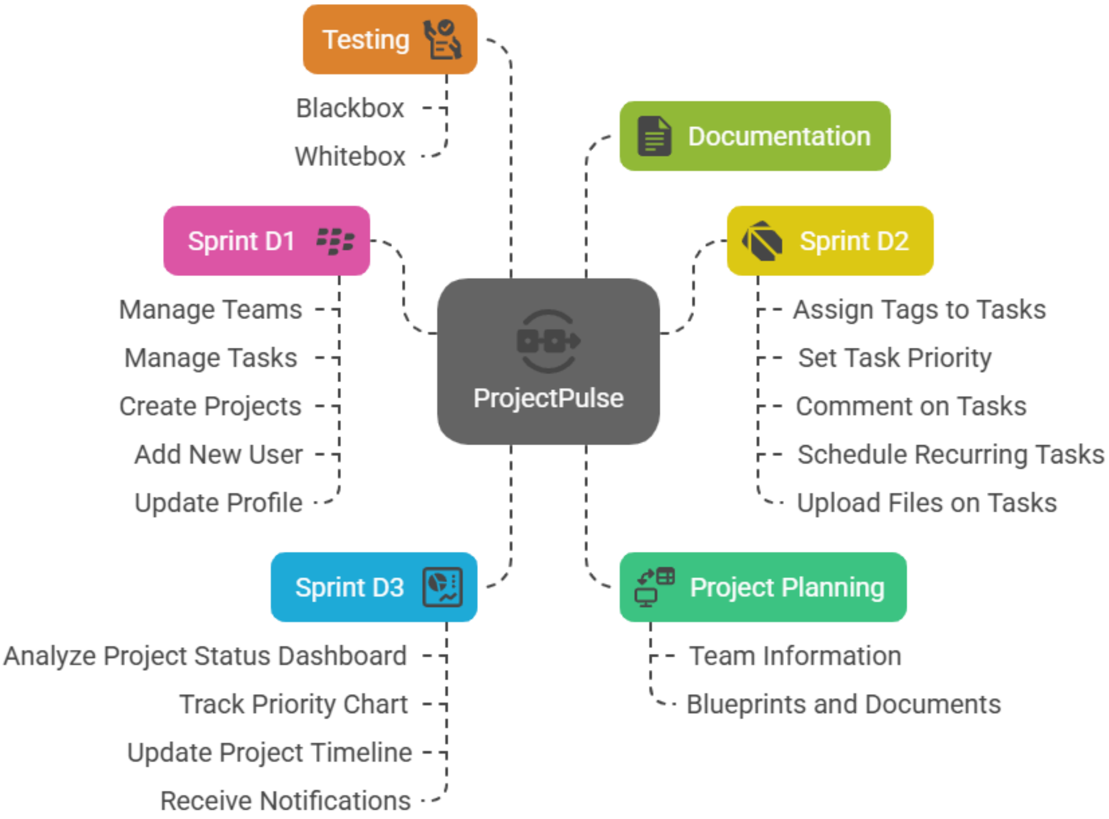


fig: Hierarchical WBS diagram.

6.1.3 Tabular WBS with Details

WBS ID	**Task Name**	**Description**	**Estimated Duration**
1.0	Project Planning	Requirement gathering, initial repo setup	21 days
2.0	Sprint D1 – Core Features	Team mgmt, task creation, project setup	10 days
3.0	Sprint D2 – Intermediate	Time tracking, recurring tasks, file uploads	10 days

4.0	Sprint D3 – Advanced	Gantt charts, timeline updates, document versioning	18 days
5.0	Testing & Deployment	Test cases, bug fixing, deployment	8 days
6.0	Documentation	SRS, API docs, user manuals	8 days

6.2 Project Gantt Chart

The Gantt chart illustrates the overall timeline and task distribution for developing the Task Management System. The project spans from early February to the end of April 2025, segmented into six primary phases.

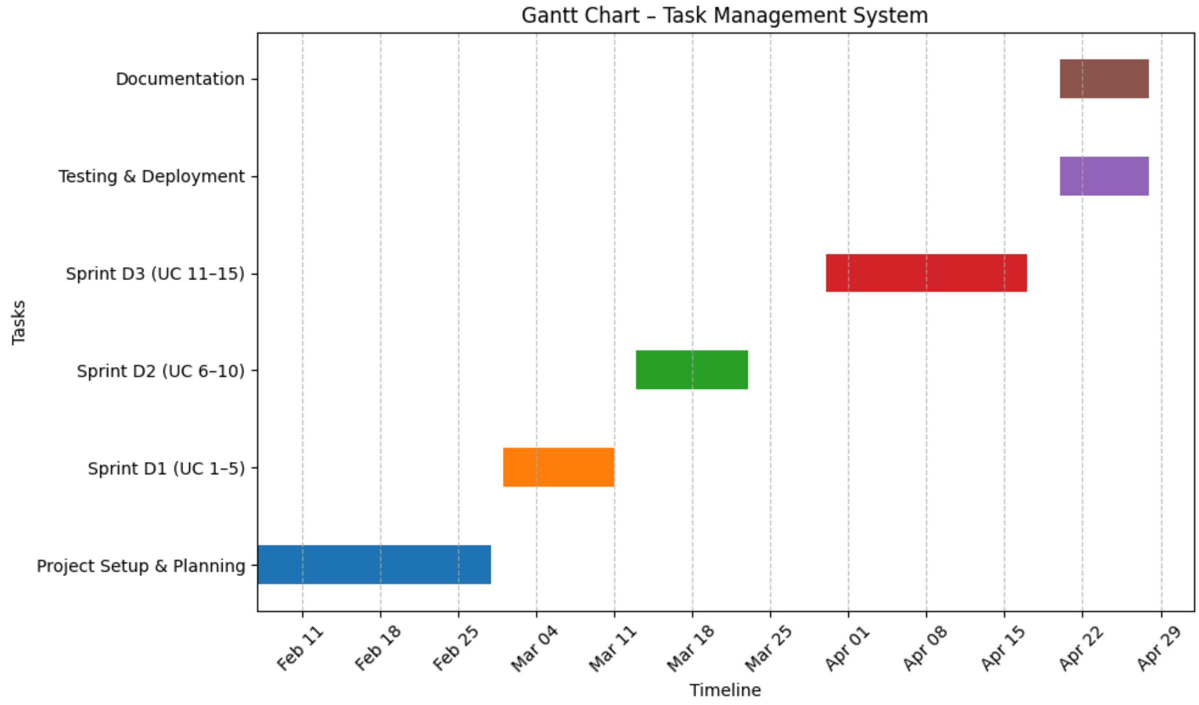


fig: Gantt chart

7. Architecture Diagram

7.1 Use Case Diagram of the system

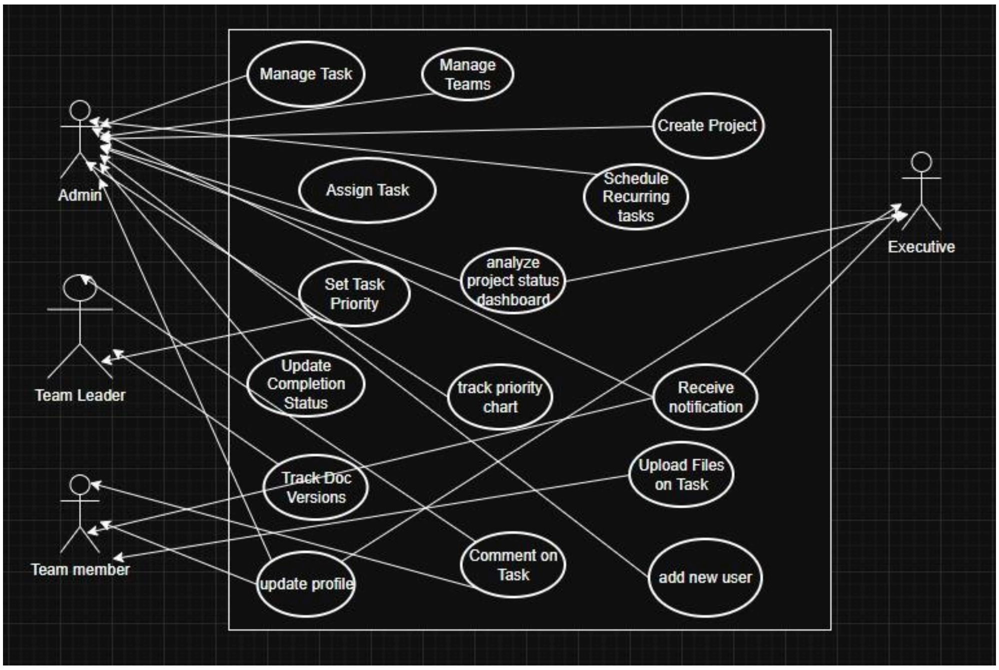


fig: use case diagram.

7.2 UML Package Diagram of Subsystems

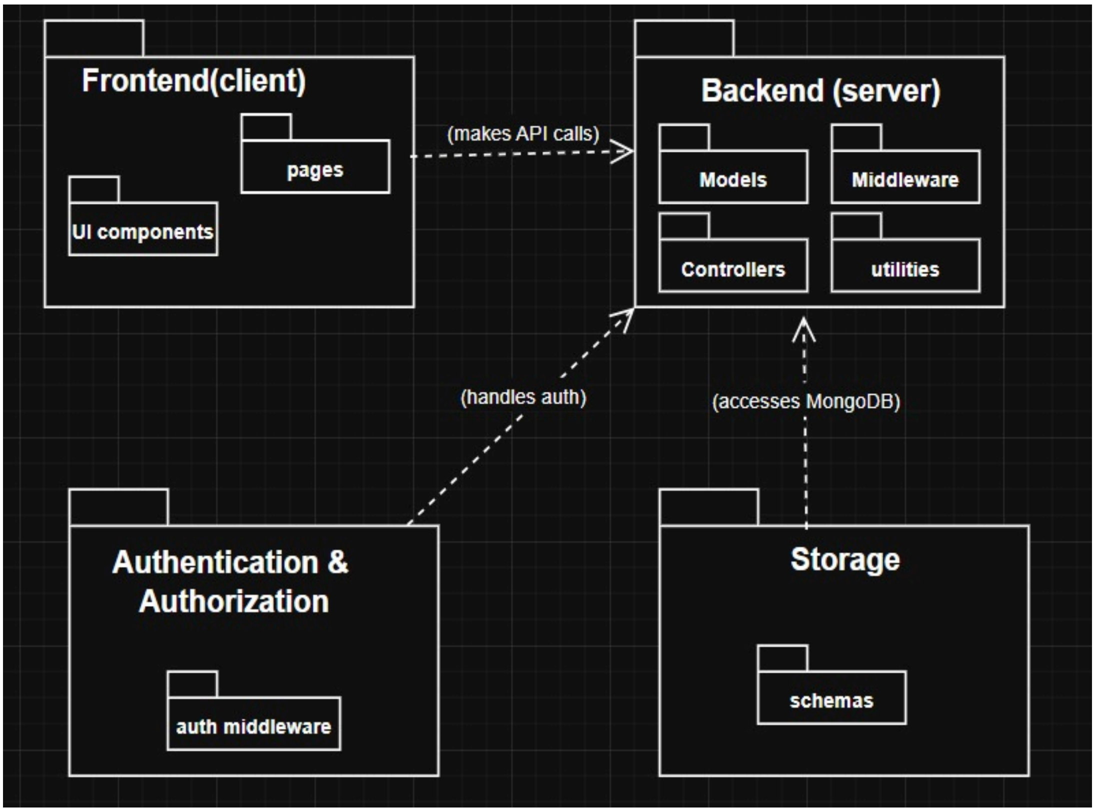


fig: package diagram.

7.3 Deployment Diagram for Client Deployments

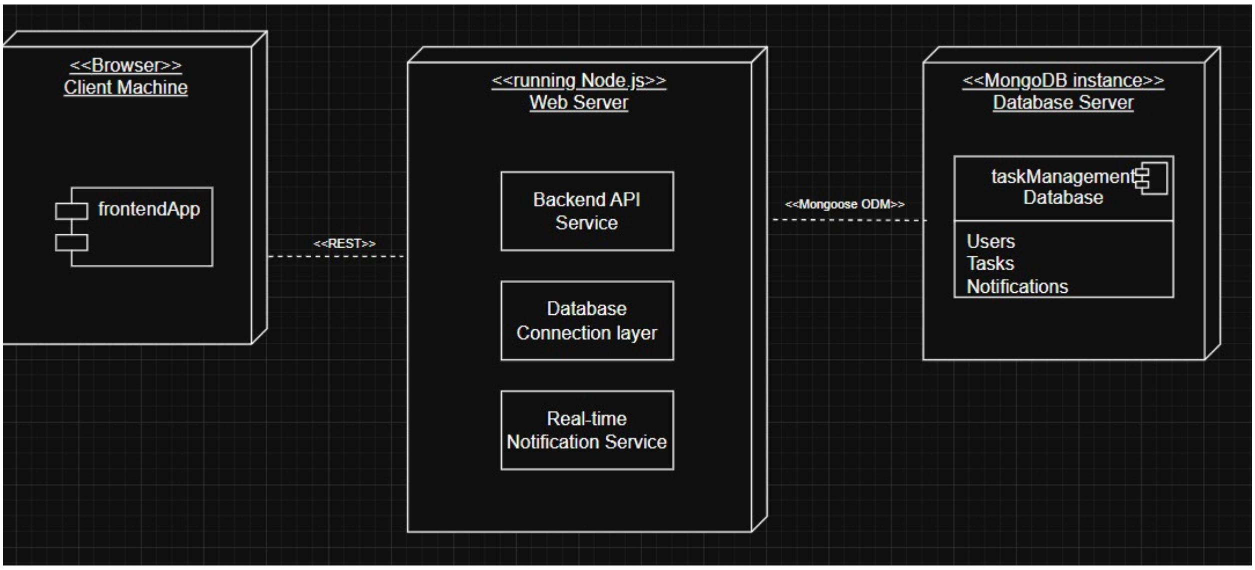


fig: deployment diagram.

7.4 Component Diagram

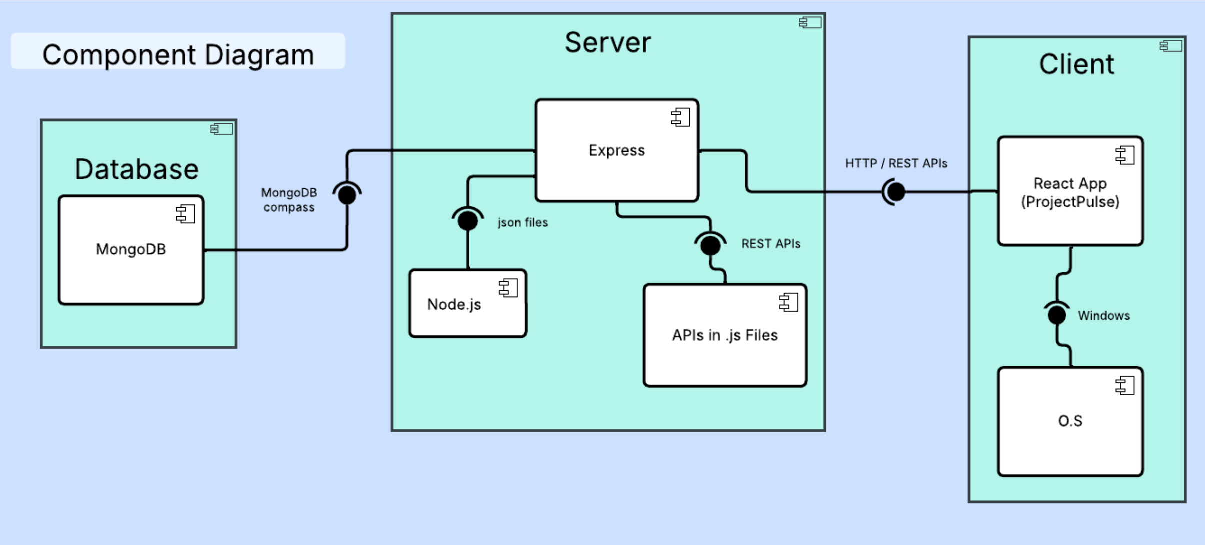


fig: component diagram.

8. Sprint 3 Design Items

8.1 Data Flow Diagram

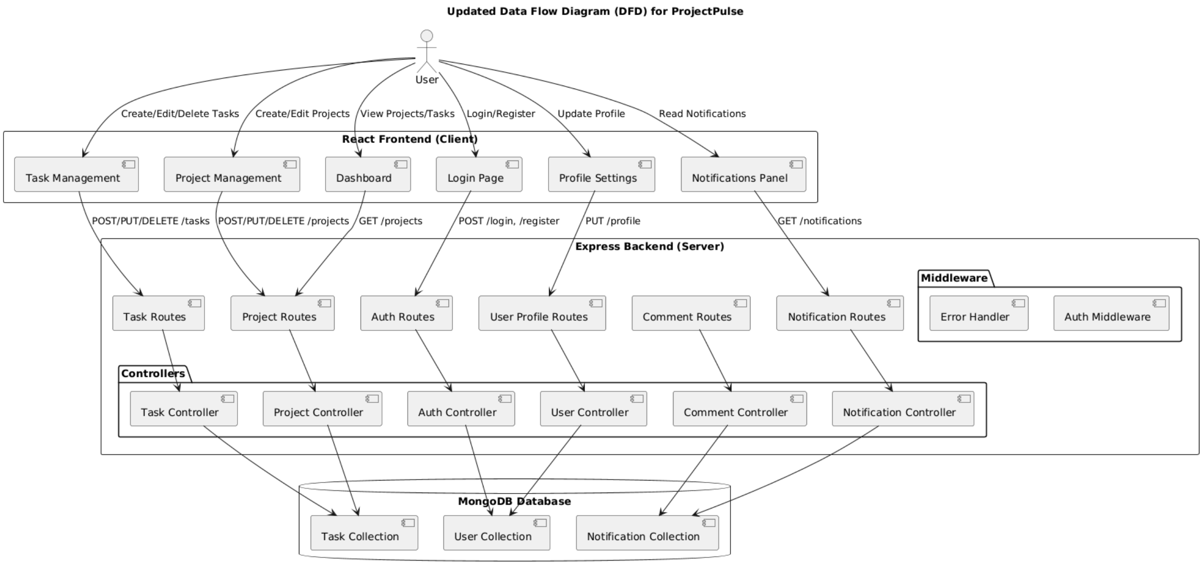


fig: data flow diagram.

8.2 A Sample Sequence Diagram

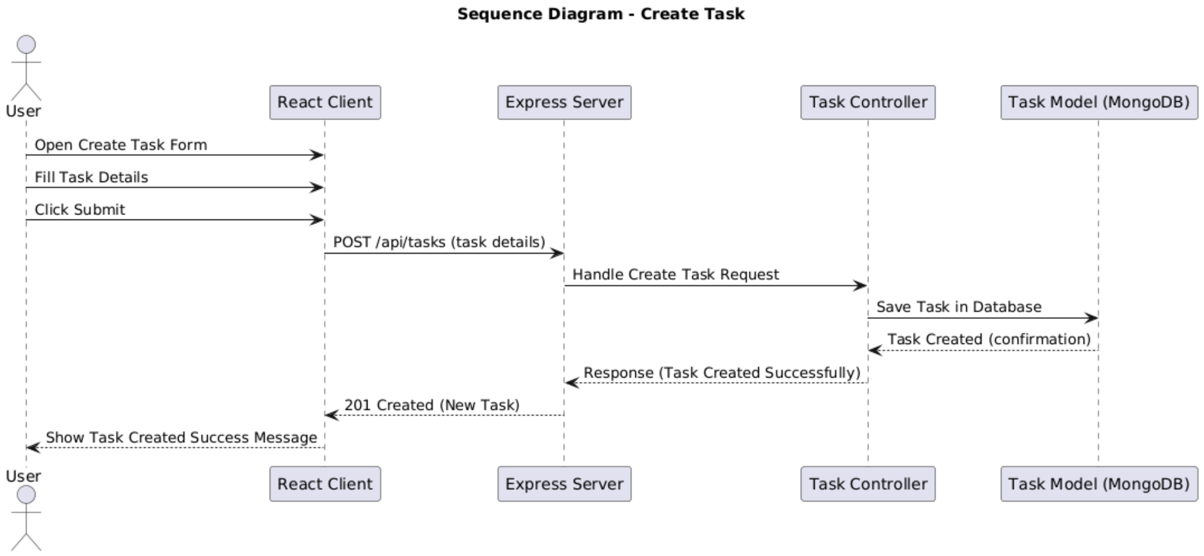
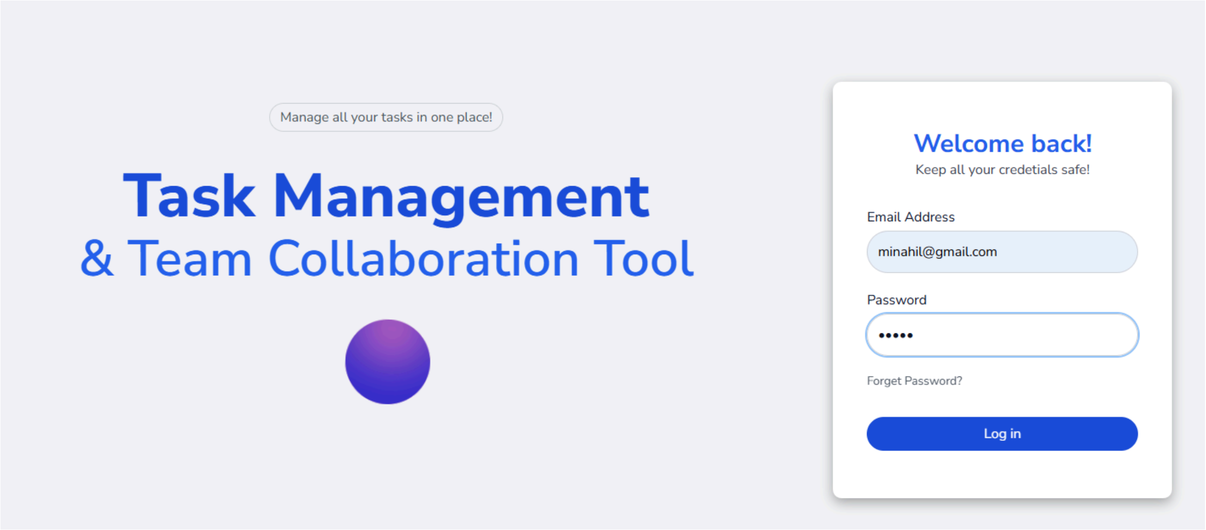


fig: a sample sequence diagram.

9. Implementation Screenshots



Manage all your tasks in one place!

Task Management
& Team Collaboration Tool

Welcome back!
Keep all your credetials safe!

Email Address: minahil@gmail.com

Password: •••••

fig: sign in page

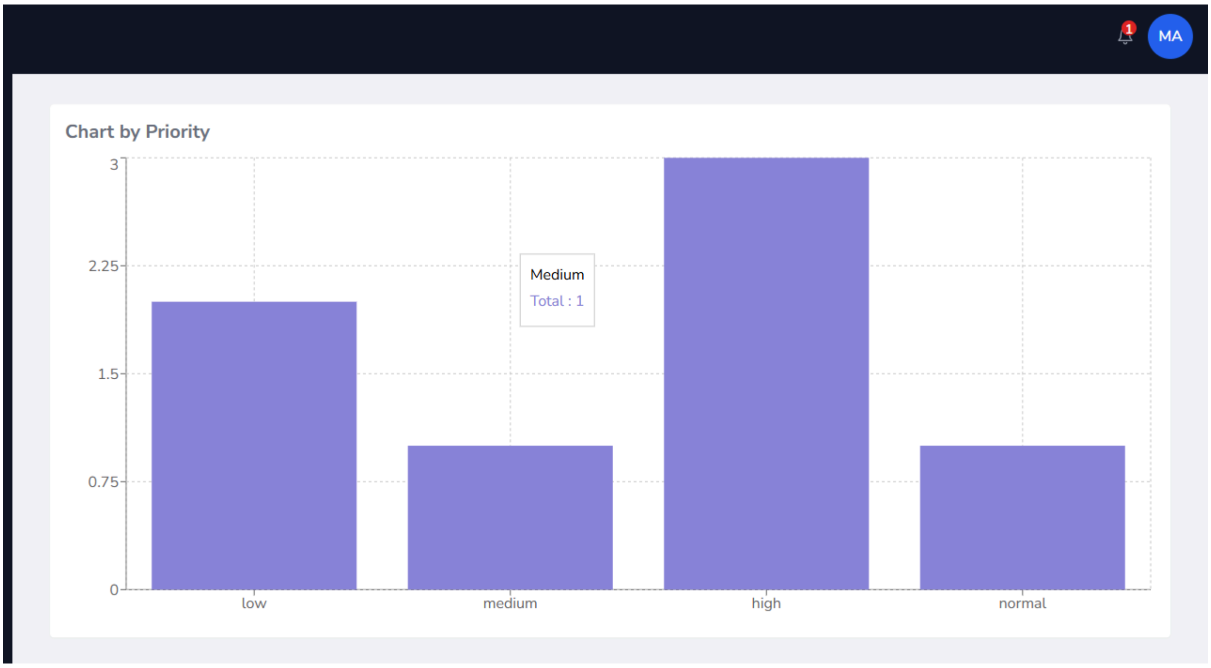
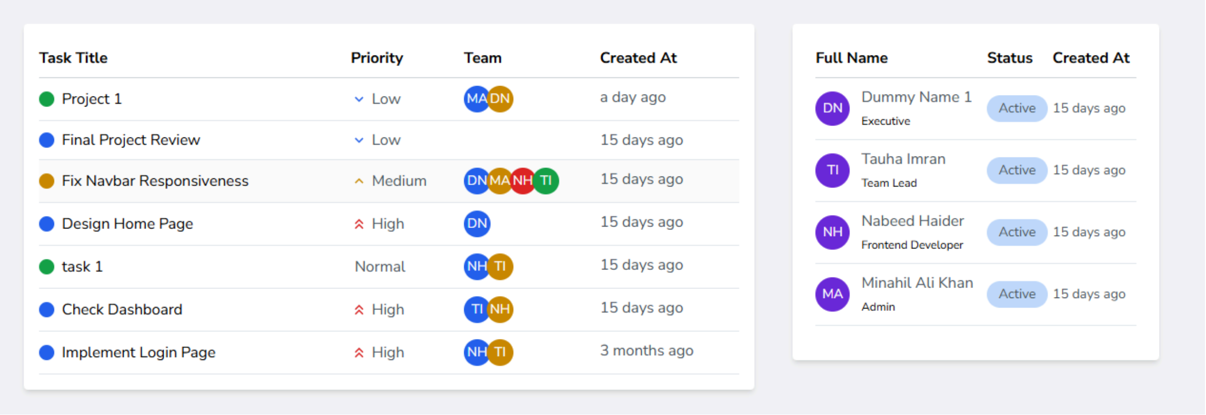


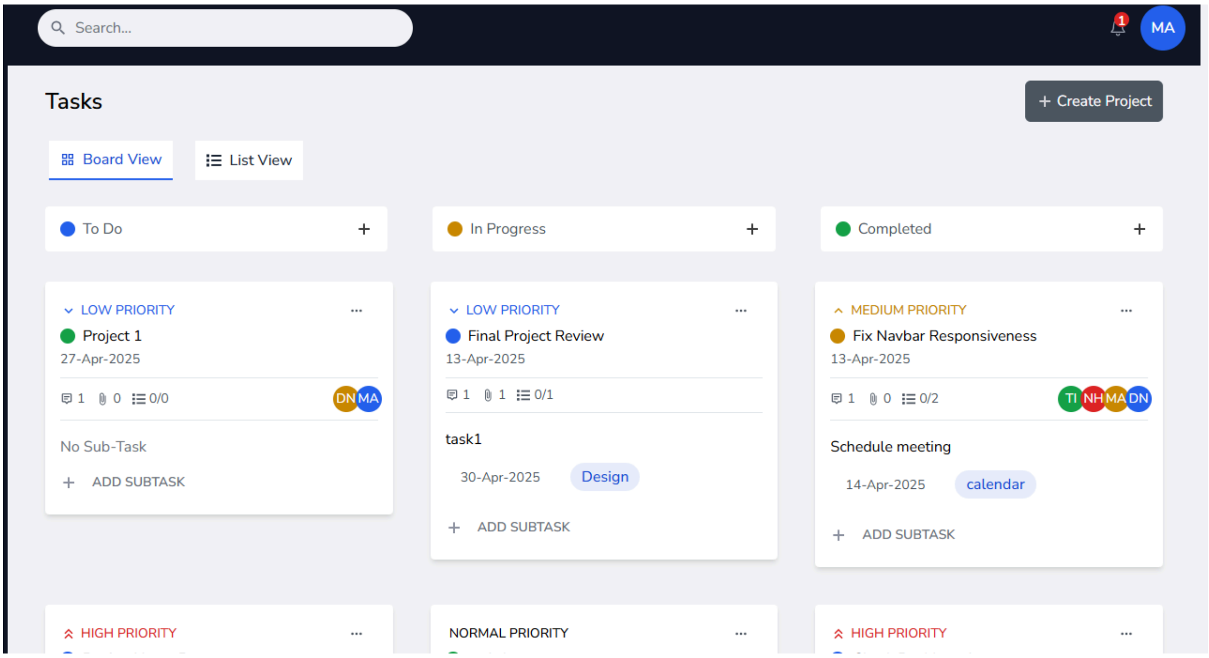
Fig: dashboard: priority chart



Task Title	Priority	Team	Created At
● Project 1	⌄ Low	MA DN	a day ago
● Final Project Review	⌄ Low		15 days ago
● Fix Navbar Responsiveness	⌃ Medium	DN MA NH TI	15 days ago
● Design Home Page	⌃⌃ High	DN	15 days ago
● task 1	Normal	NH TI	15 days ago
● Check Dashboard	⌃⌃ High	TI NH	15 days ago
● Implement Login Page	⌃⌃ High	NH TI	3 months ago

Full Name	Status	Created At
DN Dummy Name 1 Executive	Active	15 days ago
TI Tauha Imran Team Lead	Active	15 days ago
NH Nabeed Haider Frontend Developer	Active	15 days ago
MA Minahil Ali Khan Admin	Active	15 days ago

Fig: dashboard: Admin view of teams and projects



Tasks

⌄ LOW PRIORITY ···
● Project 1
27-Apr-2025
1 0 0/0 DN MA
No Sub-Task
+ ADD SUBTASK

⌄ LOW PRIORITY ···
● Final Project Review
13-Apr-2025
1 1 0/1
task1
30-Apr-2025 Design
+ ADD SUBTASK

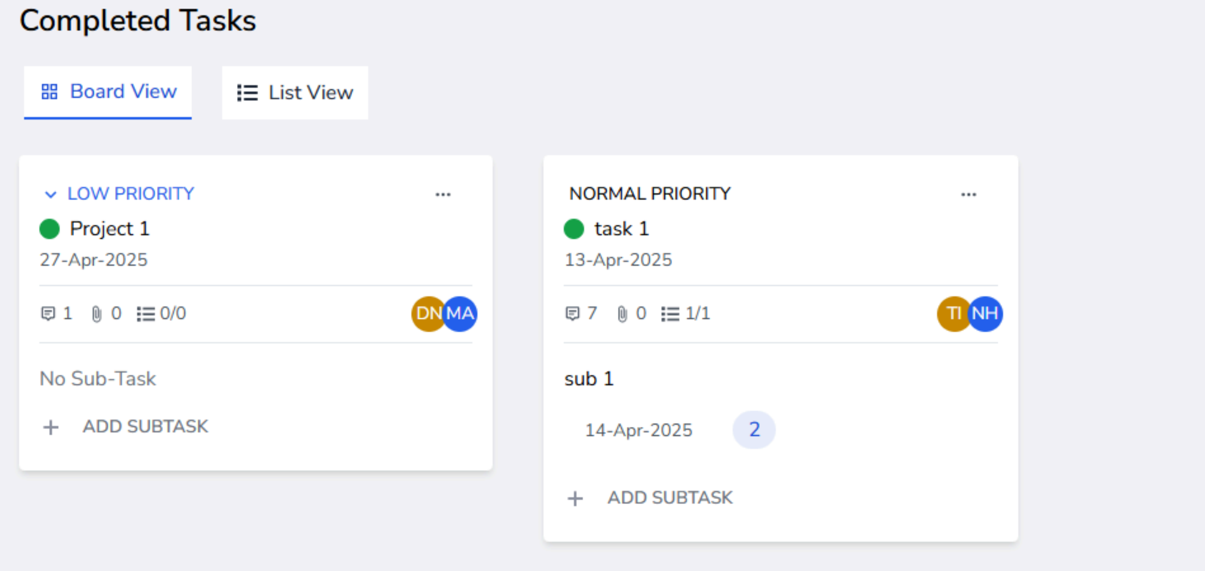
⌃ MEDIUM PRIORITY ···
● Fix Navbar Responsiveness
13-Apr-2025
1 0 0/2 TI NH MA DN
Schedule meeting
14-Apr-2025 calendar
+ ADD SUBTASK

⌃⌃ HIGH PRIORITY ···

NORMAL PRIORITY ···

⌃⌃ HIGH PRIORITY ···

Fig: projects and tasks CURD



Completed Tasks

Board View | List View

LOW PRIORITY ...

● Project 1

27-Apr-2025

1 0 0/0 DN MA

No Sub-Task

+ ADD SUBTASK

NORMAL PRIORITY ...

● task 1

13-Apr-2025

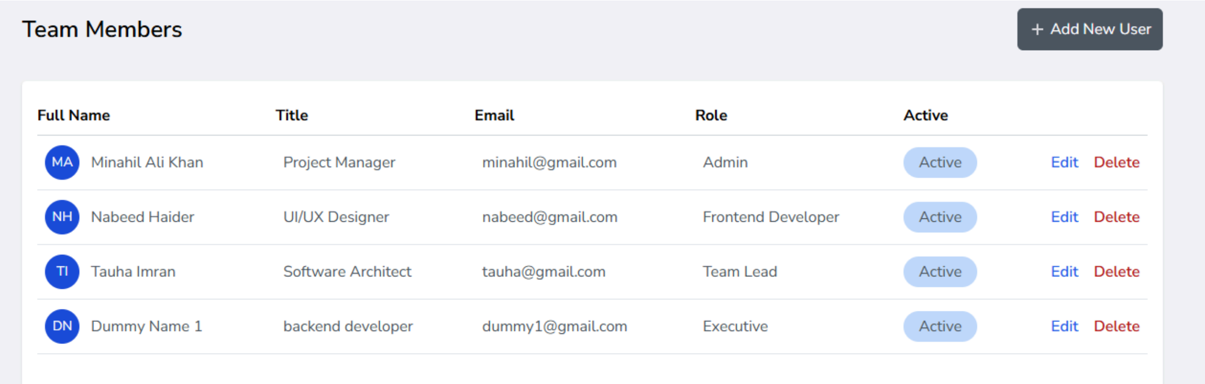
7 0 1/1 TI NH

sub 1

14-Apr-2025 2

+ ADD SUBTASK

Fig: sorted by progress

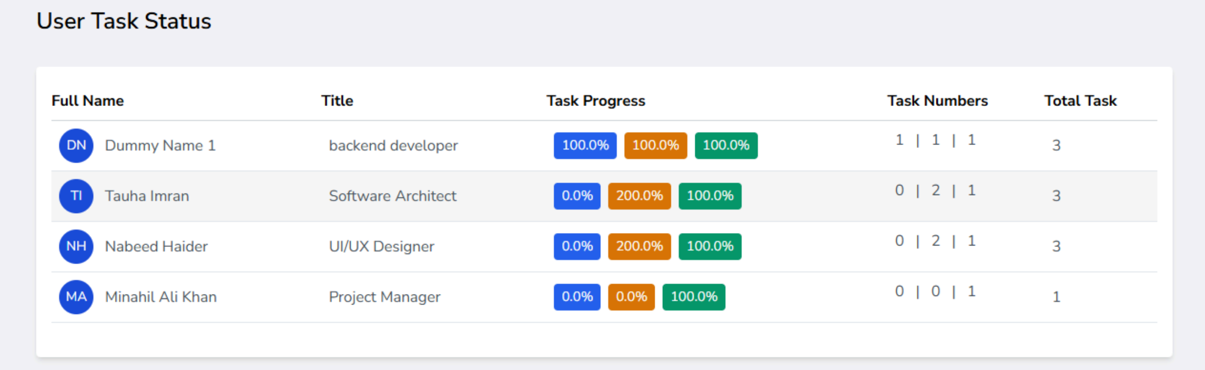


Team Members

+ Add New User

Full Name	Title	Email	Role	Active	
MA Minahil Ali Khan	Project Manager	minahil@gmail.com	Admin	Active	Edit Delete
NH Nabeed Haider	UI/UX Designer	nabeed@gmail.com	Frontend Developer	Active	Edit Delete
TI Tauha Imran	Software Architect	tauha@gmail.com	Team Lead	Active	Edit Delete
DN Dummy Name 1	backend developer	dummy1@gmail.com	Executive	Active	Edit Delete

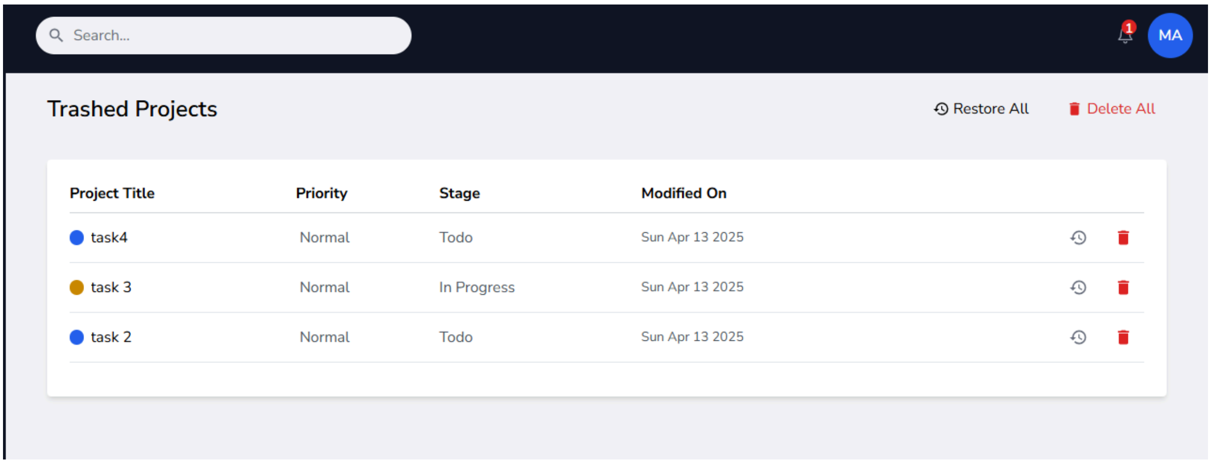
Fig: team CURD



User Task Status

Full Name	Title	Task Progress	Task Numbers	Total Task
DN Dummy Name 1	backend developer	100.0% 100.0% 100.0%	1 | 1 | 1	3
TI Tauha Imran	Software Architect	0.0% 200.0% 100.0%	0 | 2 | 1	3
NH Nabeed Haider	UI/UX Designer	0.0% 200.0% 100.0%	0 | 2 | 1	3
MA Minahil Ali Khan	Project Manager	0.0% 0.0% 100.0%	0 | 0 | 1	1

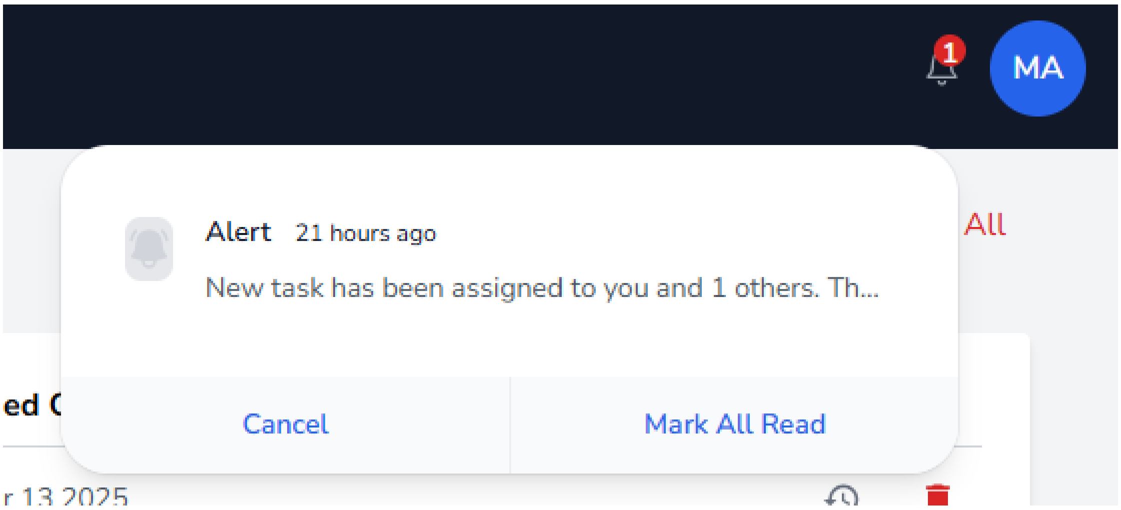
Fig: user task status



Trashed Projects

Project Title	Priority	Stage	Modified On
● task4	Normal	Todo	Sun Apr 13 2025
● task 3	Normal	In Progress	Sun Apr 13 2025
● task 2	Normal	Todo	Sun Apr 13 2025

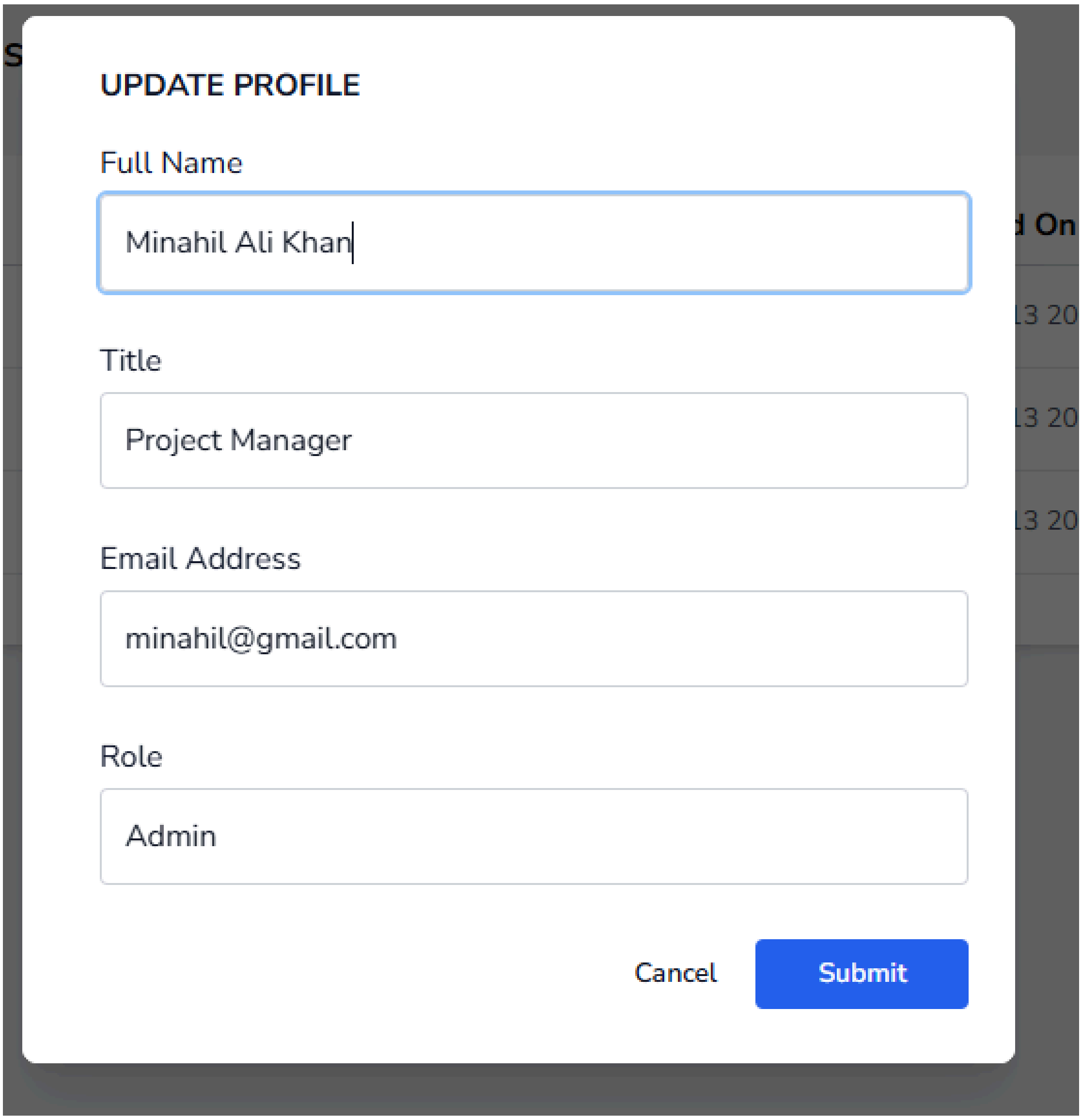
Fig: trashed projects



Alert 21 hours ago

New task has been assigned to you and 1 others. Th...

Fig: notification panel



UPDATE PROFILE

Full Name

Minahil Ali Khan

Title

Project Manager

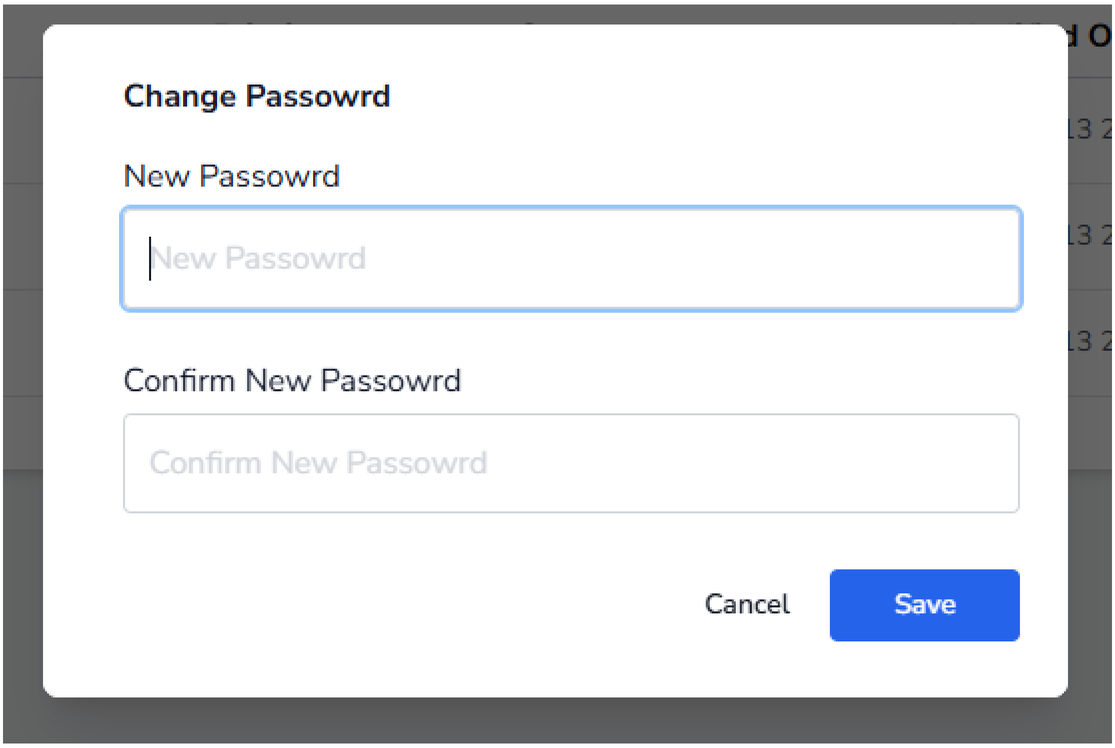
Email Address

minahil@gmail.com

Role

Admin

Fig: update profile

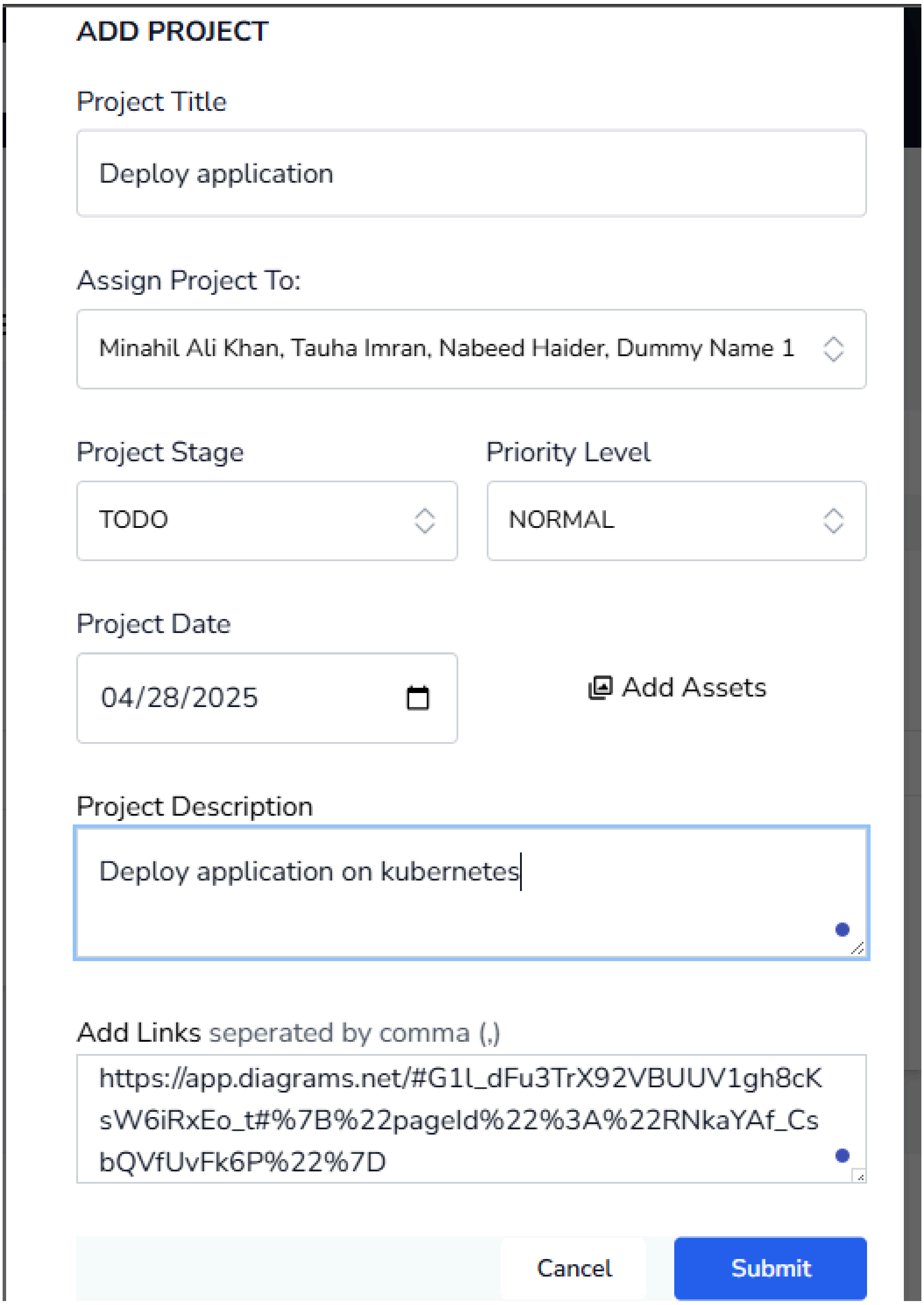


Change Passowrd

New Passowrd

Confirm New Passowrd

Fig: change password



ADD PROJECT

Project Title

Deploy application

Assign Project To:

Minahil Ali Khan, Tauha Imran, Nabeed Haider, Dummy Name 1

Project Stage

TODO

Priority Level

NORMAL

Project Date

04/28/2025

Add Assets

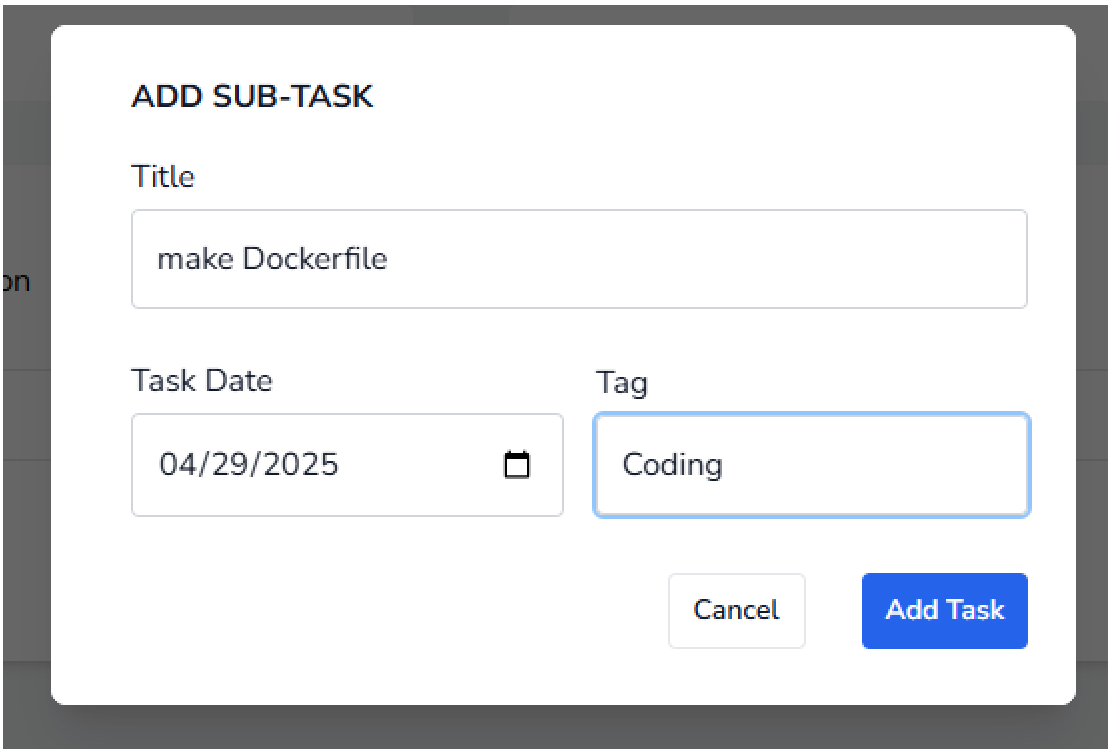
Project Description

Deploy application on kubernetes

Add Links seperated by comma (,)

https://app.diagrams.net/#G1L_dFu3TrX92VBUUV1gh8cKsW6iRxEo_t#%7B%22pageId%22%3A%22RNkaYAf_CsbQVfUvFk6P%22%7D

Fig: add project



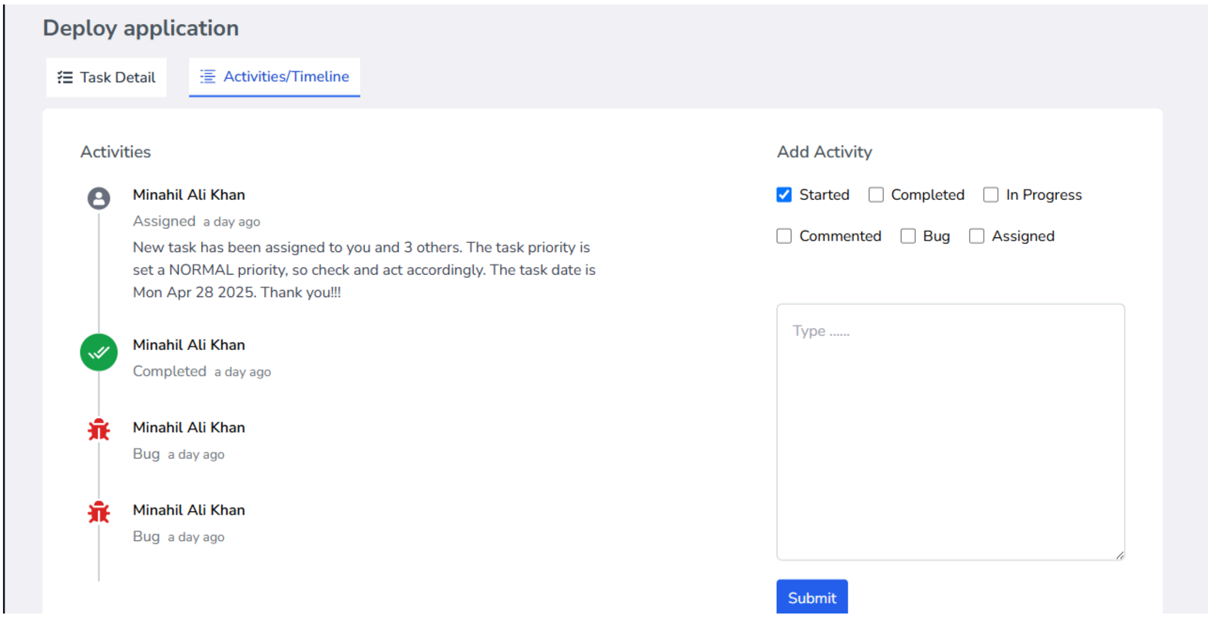
ADD SUB-TASK

Title: make Dockerfile

Task Date: 04/29/2025

Tag: Coding

Fig: add task to project



Deploy application

Activities

Minahil Ali Khan
Assigned a day ago
New task has been assigned to you and 3 others. The task priority is set a NORMAL priority, so check and act accordingly. The task date is Mon Apr 28 2025. Thank you!!!

Minahil Ali Khan
Completed a day ago

Minahil Ali Khan
Bug a day ago

Minahil Ali Khan
Bug a day ago

Add Activity

☑ Started ☐ Completed ☐ In Progress
☐ Commented ☐ Bug ☐ Assigned

Type ......

Fig: project timeline

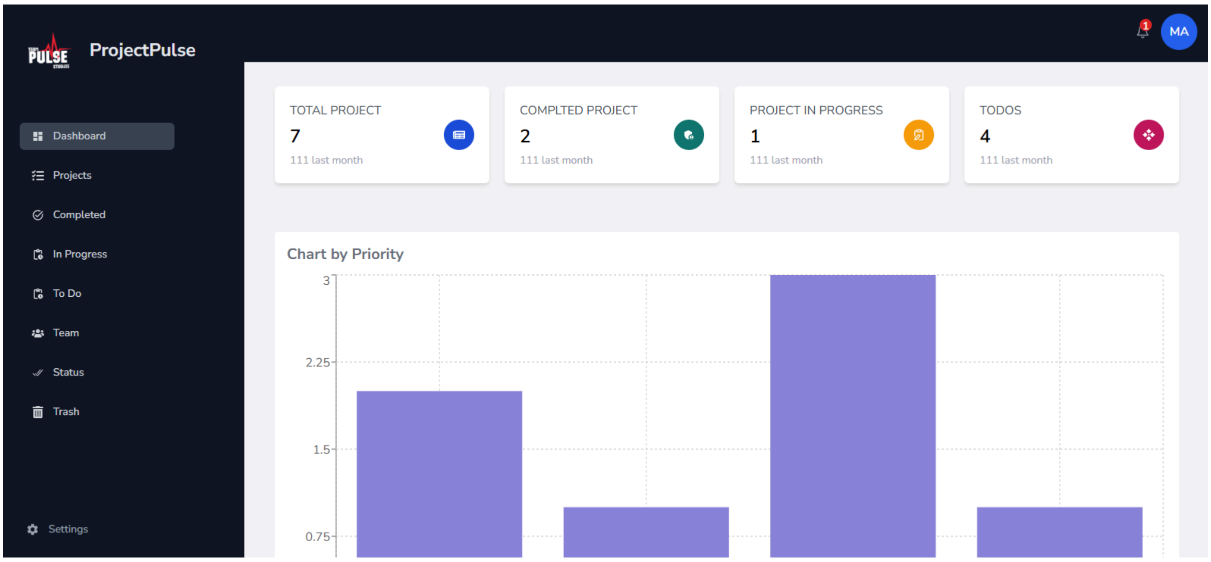


Fig: admin view

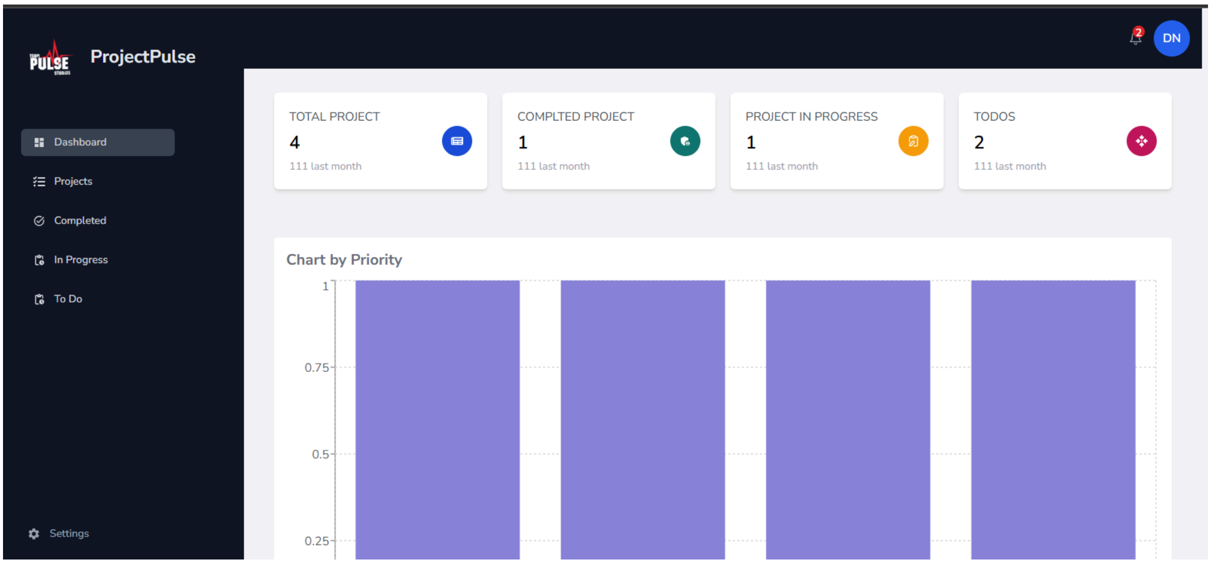


Fig: team member view

10. Product Burndown Chart

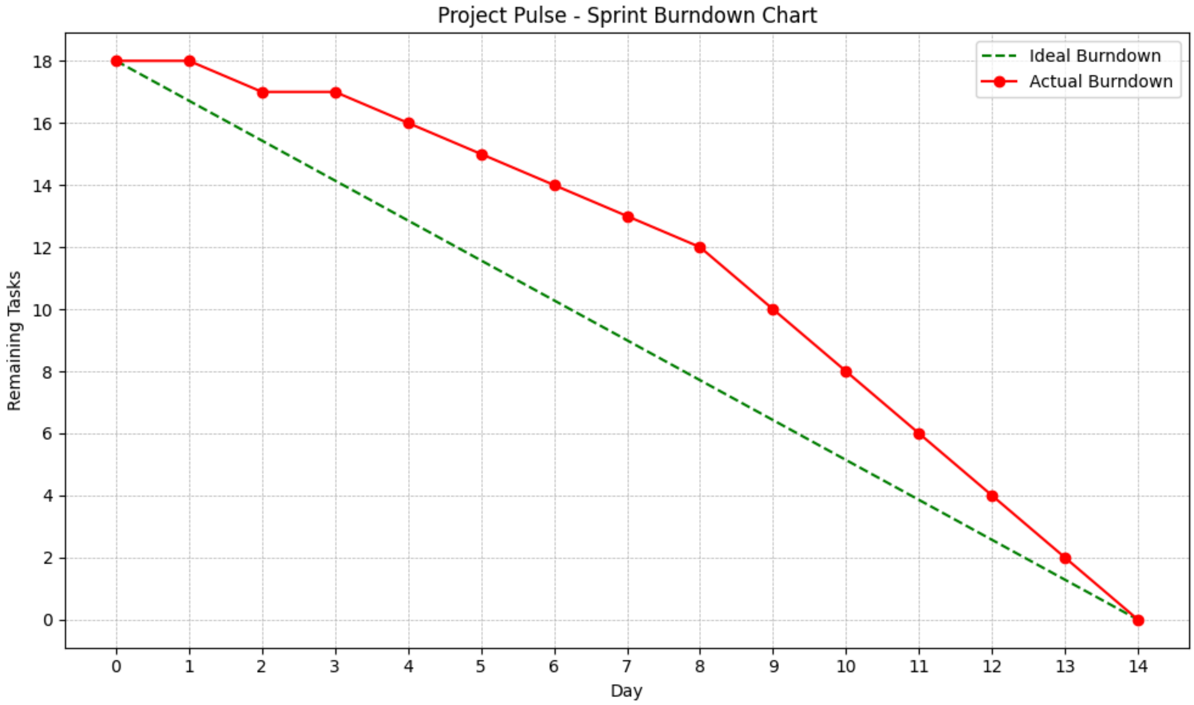
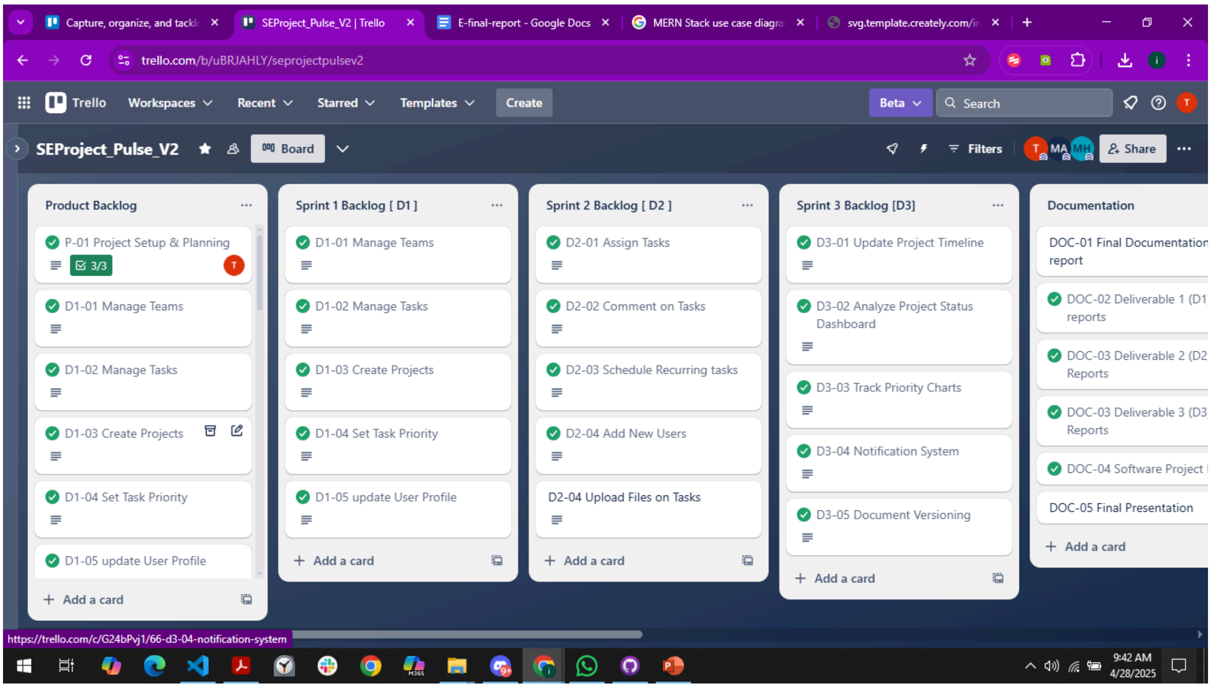


fig: burndown chart

11. Trello Board Screenshots

The following are screenshots of the current status of our trello board


Capture, organize, and tackl
SEProject_Pulse_V2 | Trello
E-final-report - Google Docs
MERN Stack use case diagra
svg.template.creately.com/in
trello.com/b/uBRJAHLY/seprojectpulsev2
Trello
Workspaces
Recent
Starred
Templates
Create
Beta
Search
SEProject_Pulse_V2
Board
Filters
Share
Add a card
https://trello.com/c/G24bPvj1/66-d3-04-notification-system
9:42 AM
4/28/2025

Product Backlog

- P-01 Project Setup & Planning 3/3
- D1-01 Manage Teams
- D1-02 Manage Tasks
- D1-03 Create Projects
- D1-04 Set Task Priority
- D1-05 update User Profile

Sprint 1 Backlog [D1]

- D1-01 Manage Teams
- D1-02 Manage Tasks
- D1-03 Create Projects
- D1-04 Set Task Priority
- D1-05 update User Profile

Sprint 2 Backlog [D2]

- D2-01 Assign Tasks
- D2-02 Comment on Tasks
- D2-03 Schedule Recurring tasks
- D2-04 Add New Users
- D2-04 Upload Files on Tasks

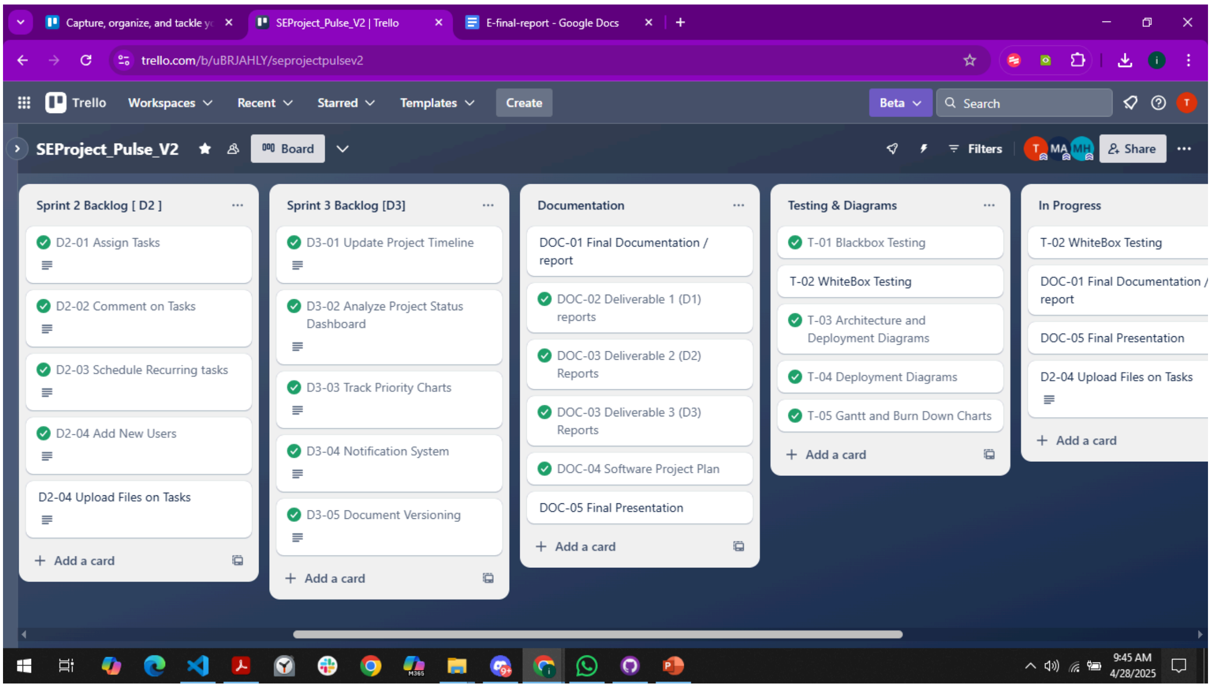
Sprint 3 Backlog [D3]

- D3-01 Update Project Timeline
- D3-02 Analyze Project Status Dashboard
- D3-03 Track Priority Charts
- D3-04 Notification System
- D3-05 Document Versioning

Documentation

- DOC-01 Final Documentation report
- DOC-02 Deliverable 1 (D1 reports
- DOC-03 Deliverable 2 (D2 Reports
- DOC-03 Deliverable 3 (D3 Reports
- DOC-04 Software Project
- DOC-05 Final Presentation

fig: Trello board pt1


Capture, organize, and tackle y
SEProject_Pulse_V2 | Trello
E-final-report - Google Docs
trello.com/b/uBRJAHLY/seprojectpulsev2
Trello
Workspaces
Recent
Starred
Templates
Create
Beta
Search
SEProject_Pulse_V2
Board
Filters
Share
9:45 AM
4/28/2025

Sprint 2 Backlog [D2]
- D2-01 Assign Tasks
- D2-02 Comment on Tasks
- D2-03 Schedule Recurring tasks
- D2-04 Add New Users
- D2-04 Upload Files on Tasks
- Add a card

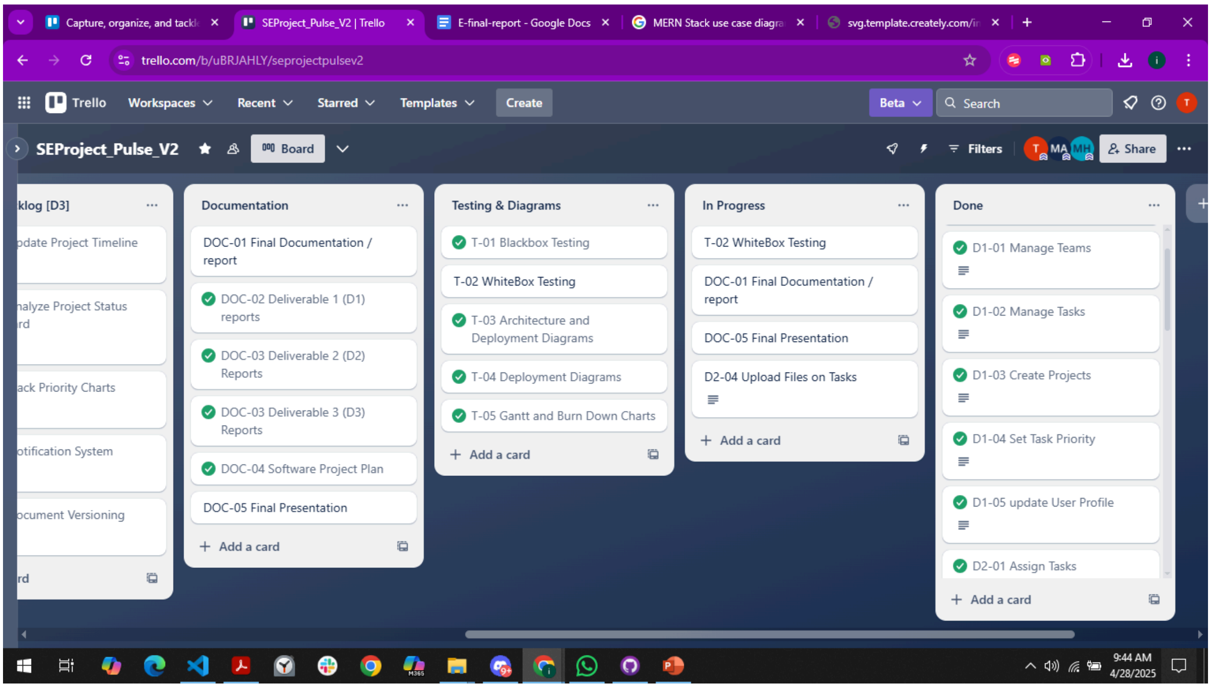
Sprint 3 Backlog [D3]
- D3-01 Update Project Timeline
- D3-02 Analyze Project Status Dashboard
- D3-03 Track Priority Charts
- D3-04 Notification System
- D3-05 Document Versioning
- Add a card

Documentation
- DOC-01 Final Documentation / report
- DOC-02 Deliverable 1 (D1) reports
- DOC-03 Deliverable 2 (D2) Reports
- DOC-03 Deliverable 3 (D3) Reports
- DOC-04 Software Project Plan
- DOC-05 Final Presentation
- Add a card

Testing & Diagrams
- T-01 Blackbox Testing
- T-02 WhiteBox Testing
- T-03 Architecture and Deployment Diagrams
- T-04 Deployment Diagrams
- T-05 Gantt and Burn Down Charts
- Add a card

In Progress
- T-02 WhiteBox Testing
- DOC-01 Final Documentation / report
- DOC-05 Final Presentation
- D2-04 Upload Files on Tasks
- Add a card

fig: Trello board pt2


Capture, organize, and tackl
SEProject_Pulse_V2 | Trello
E-final-report - Google Docs
MERN Stack use case diagra
svg.template.creately.com/
trello.com/b/uBRJAHLY/seprojectpulsev2
Trello
Workspaces
Recent
Starred
Templates
Create
Beta
Search
SEProject_Pulse_V2
Board
Filters
Share
9:44 AM
4/28/2025

klog [D3]
- pdate Project Timeline
- alyze Project Status rd
- ack Priority Charts
- otification System
- ocument Versioning
- rd

Documentation
- DOC-01 Final Documentation / report
- DOC-02 Deliverable 1 (D1) reports
- DOC-03 Deliverable 2 (D2) Reports
- DOC-03 Deliverable 3 (D3) Reports
- DOC-04 Software Project Plan
- DOC-05 Final Presentation
- Add a card

Testing & Diagrams
- T-01 Blackbox Testing
- T-02 WhiteBox Testing
- T-03 Architecture and Deployment Diagrams
- T-04 Deployment Diagrams
- T-05 Gantt and Burn Down Charts
- Add a card

In Progress
- T-02 WhiteBox Testing
- DOC-01 Final Documentation / report
- DOC-05 Final Presentation
- D2-04 Upload Files on Tasks
- Add a card

Done
- D1-01 Manage Teams
- D1-02 Manage Tasks
- D1-03 Create Projects
- D1-04 Set Task Priority
- D1-05 update User Profile
- D2-01 Assign Tasks
- Add a card

fig: Trello board pt3

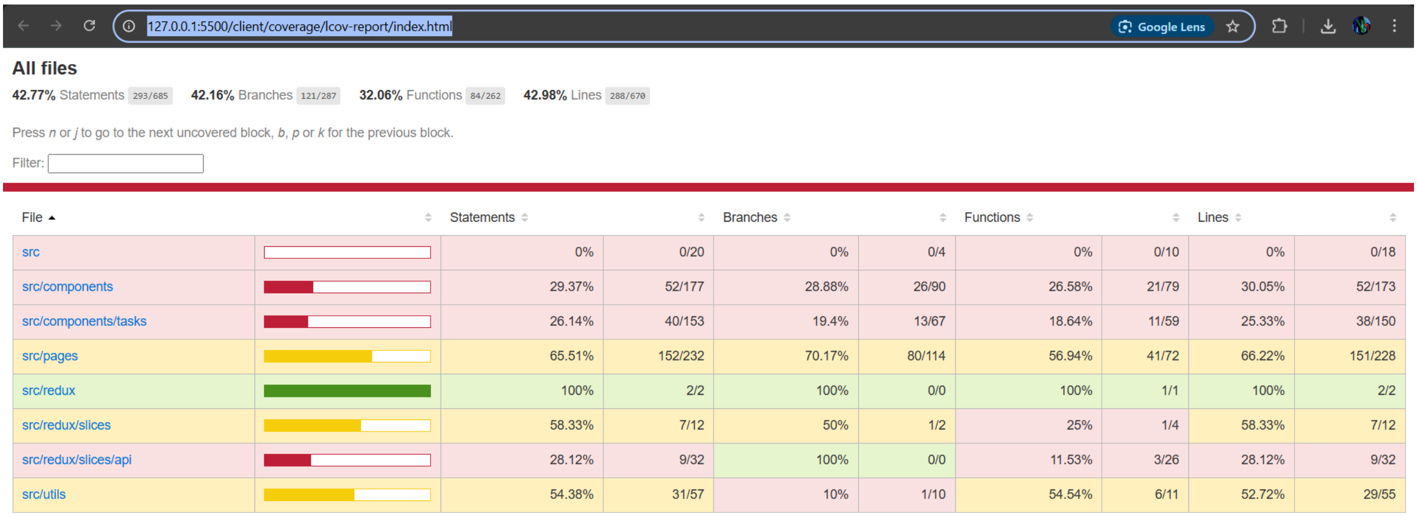
Visit the Trello Board for yourself using the link : https://trello.com/b/uBRJAHLY/seprojectpulsev2

12. Black Box Test Cases

Test Case ID	Test Area	Description	Status
TC-FF-01	Functional	Create and assign a task	Passed
TC-FF-02	Functional	Manage teams – add/remove member	Passed
TC-FF-03	Functional	Create project with milestones	Passed
TC-FF-04	Functional	Track priority chart	Passed
TC-FF-05	Functional	Comment on tasks and notify	Passed
TC-FF-06	Functional	Notification on status update	Passed
TC-FF-07	Functional	Schedule recurring task weekly	Passed
TC-FF-08	Functional	Update profile avatar & bio	Passed
TC-FF-09	Functional	Assign tags to a task	Passed
TC-FF-10	Functional	Update project timeline via Gantt chart	Passed
TC-FF-11	Functional	Filter project status dashboard	Passed
TC-FF-12	Functional	Upload files with version control	Passed
TC-ECP-01	ECP *(Equivalence Class Partitioning)*	Create task with valid inputs	Passed
TC-ECP-02	ECP	Title too short validation	Passed
TC-ECP-03	ECP	Title too long validation	Passed
TC-ECP-04	ECP	Due date in the past validation	Passed
TC-ECP-05	ECP	Invalid date format validation	Passed
TC-BVA-01	BVA *(Boundary Value Analysis)*	Title at minimum length	Passed
TC-BVA-02	BVA	Title just below minimum length	Passed
TC-BVA-03	BVA	Title at maximum length	Passed
TC-BVA-04	BVA	Title above maximum length	Passed
TC-BVA-05	BVA	Recurrence interval = 1 day (minimum)	Passed
TC-BVA-06	BVA	Recurrence interval = 0 days (invalid)	Passed
TC-BVA-07	BVA	Recurrence interval = 30 days (maximum)	Passed
TC-BVA-08	BVA	Recurrence interval = 31 days (invalid)	Passed

13. White Box Test Cases

13.1 FrontEnd White Box Testing results



127.0.0.1:5500/client/coverage/lcov-report/index.html

All files

42.77% Statements 293/685 **42.16%** Branches 121/287 **32.06%** Functions 84/262 **42.98%** Lines 288/670

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

File ▲		Statements		Branches		Functions		Lines	
src		0%	0/20	0%	0/4	0%	0/10	0%	0/18
src/components		29.37%	52/177	28.88%	26/90	26.58%	21/79	30.05%	52/173
src/components/tasks		26.14%	40/153	19.4%	13/67	18.64%	11/59	25.33%	38/150
src/pages		65.51%	152/232	70.17%	80/114	56.94%	41/72	66.22%	151/228
src/redux		100%	2/2	100%	0/0	100%	1/1	100%	2/2
src/redux/slices		58.33%	7/12	50%	1/2	25%	1/4	58.33%	7/12
src/redux/slices/api		28.12%	9/32	100%	0/0	11.53%	3/26	28.12%	9/32
src/utils		54.38%	31/57	10%	1/10	54.54%	6/11	52.72%	29/55

fig: white box Testing results pt1

Created .test.jsx files to test for most of the functionalities (.js files)

Run using command : npm run test:coverage

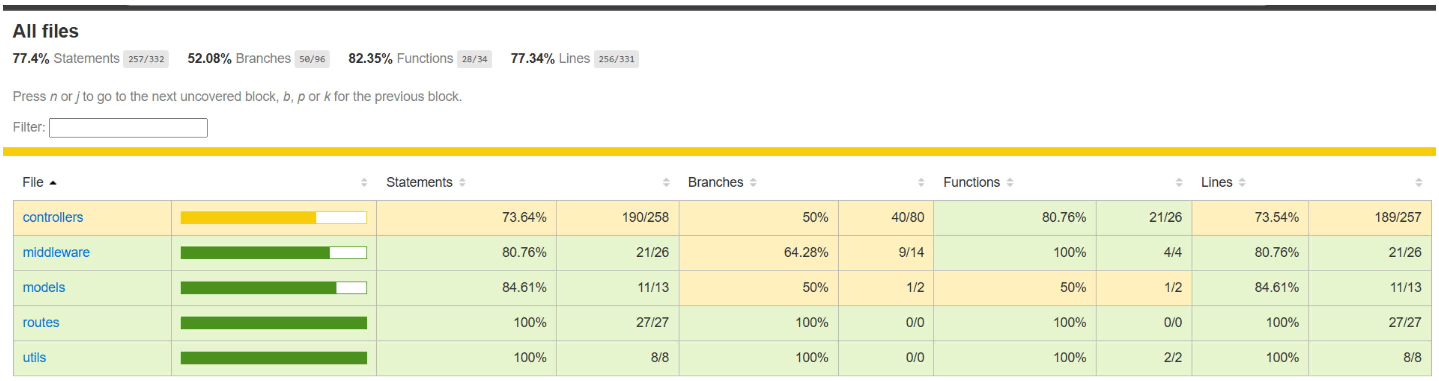
43 % statements coverage

42 % branches coverage

32 % functions coverage

43 % lines coverage

13.2 Backend White Box Testing Results



All files

77.4% Statements 257/332   52.08% Branches 50/96   82.35% Functions 28/34   77.34% Lines 256/331

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

File ▲		Statements		Branches		Functions		Lines	
controllers		73.64%	190/258	50%	40/80	80.76%	21/26	73.54%	189/257
middleware		80.76%	21/26	64.28%	9/14	100%	4/4	80.76%	21/26
models		84.61%	11/13	50%	1/2	50%	1/2	84.61%	11/13
routes		100%	27/27	100%	0/0	100%	0/0	100%	27/27
utils		100%	8/8	100%	0/0	100%	2/2	100%	8/8

fig: white box Testing results pt2

Created .test.js files to test for most of files (.js files) present at the backend(server) directory.

Run using command : npm run test

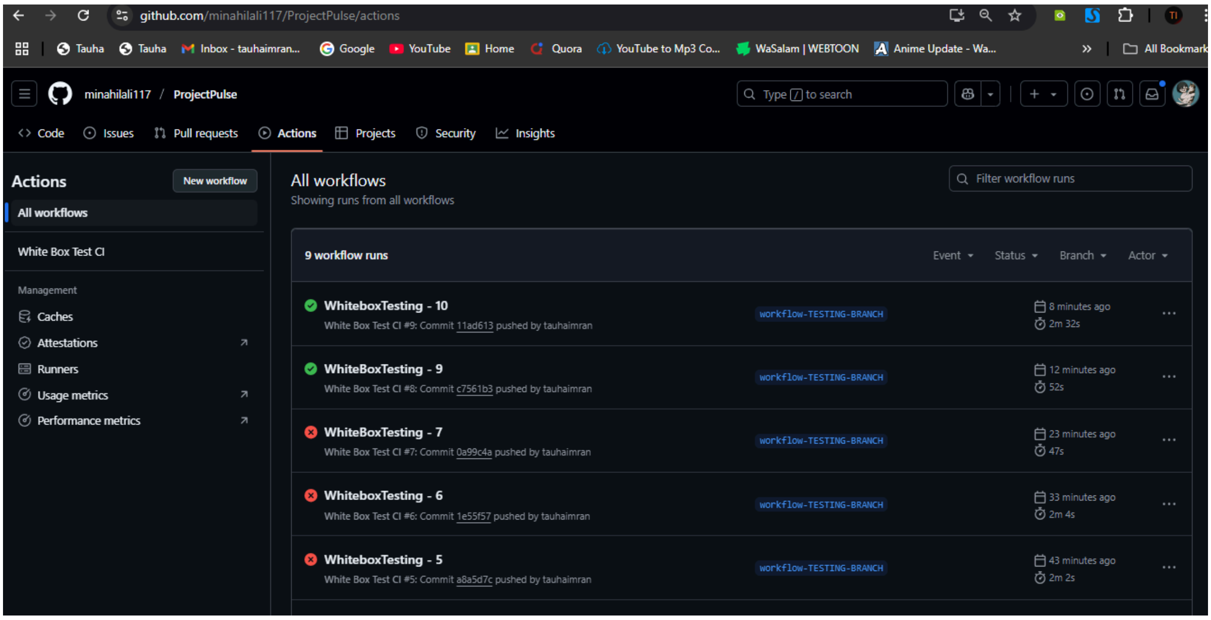
77.4 % statements coverage

52.1 % branches coverage

82.4 % functions coverage

77.3 % lines coverage

13.3 White Box Testing with Github Workflows results



All workflows
Showing runs from all workflows

9 workflow runs

WhiteboxTesting - 10
White Box Test CI #9: Commit 11ad613 pushed by tauhaimran — workflow-TESTING-BRANCH — 8 minutes ago — 2m 32s

WhiteBoxTesting - 9
White Box Test CI #8: Commit c7561b3 pushed by tauhaimran — workflow-TESTING-BRANCH — 12 minutes ago — 52s

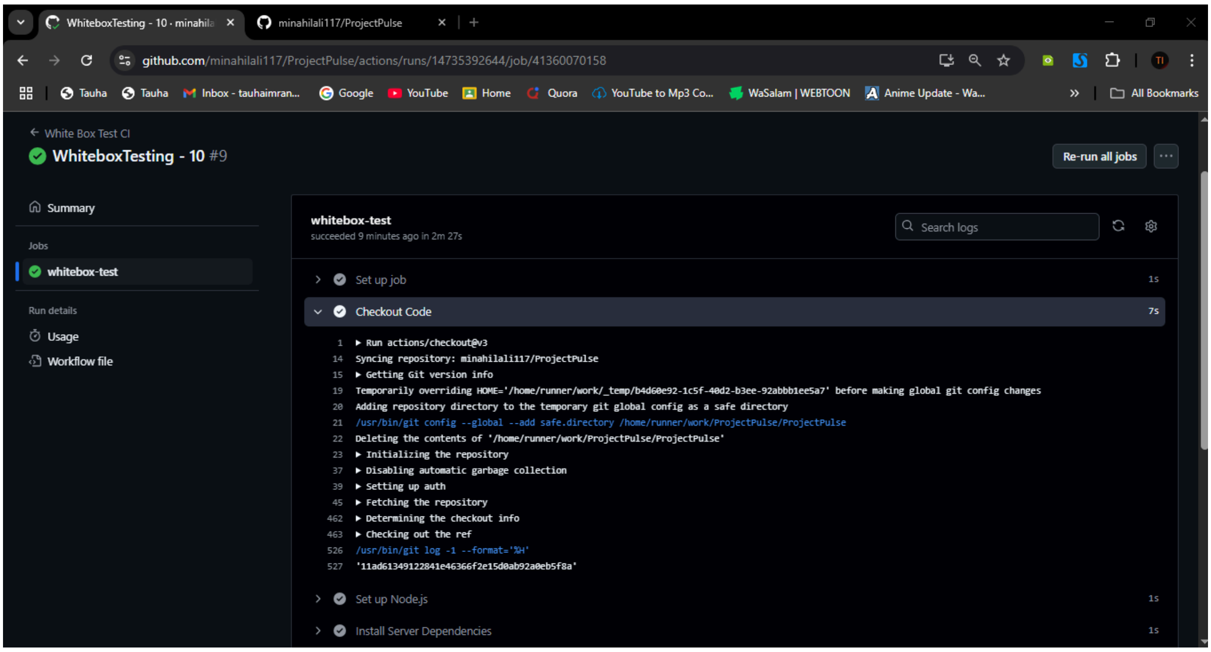
WhiteBoxTesting - 7
White Box Test CI #7: Commit 0a99c4a pushed by tauhaimran — workflow-TESTING-BRANCH — 23 minutes ago — 47s

WhiteboxTesting - 6
White Box Test CI #6: Commit 1e55f57 pushed by tauhaimran — workflow-TESTING-BRANCH — 33 minutes ago — 2m 4s

WhiteboxTesting - 5
White Box Test CI #5: Commit a8a5d7c pushed by tauhaimran — workflow-TESTING-BRANCH — 43 minutes ago — 2m 2s

fig: white box Testing github workflow pt 1

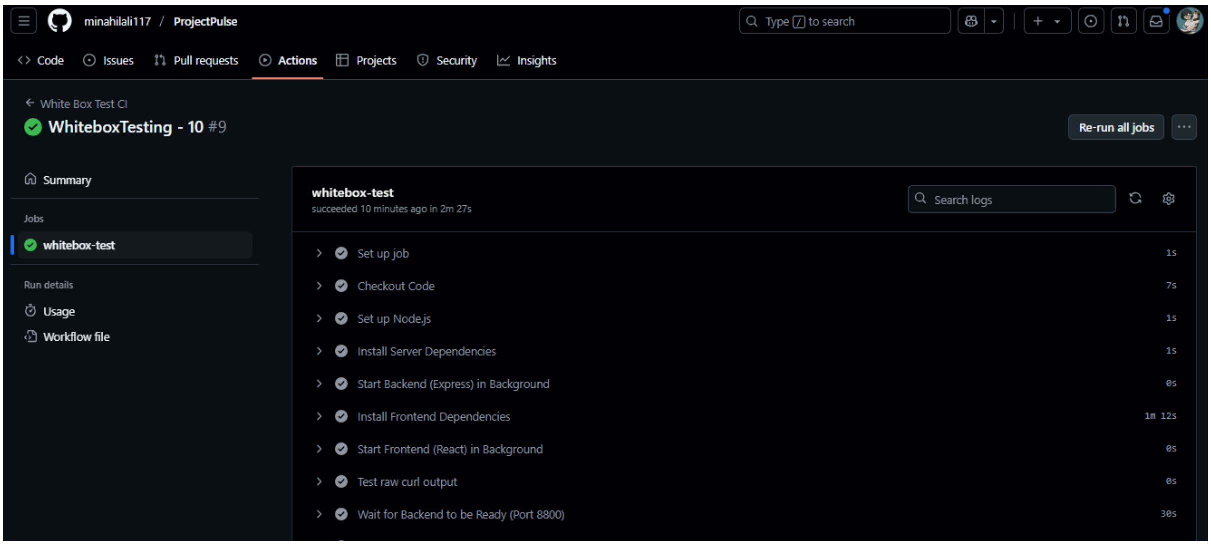
Attempts to run White Box testing with github Workflows were also made. The figure above shows the attempts made and how many workflows were successful. Some detailed pictures of the workflow are shown below



whitebox-test
succeeded 9 minutes ago in 2m 27s

```
> Set up job                                                         1s
v Checkout Code                                                      7s
  ▶ Run actions/checkout@v3
  Syncing repository: minahilali117/ProjectPulse
  ▶ Getting Git version info
  Temporarily overriding HOME='/home/runner/work/_temp/b4d60e92-1c5f-40d2-b3ee-92abbb1ee5a7' before making global git config changes
  Adding repository directory to the temporary git global config as a safe directory
  /usr/bin/git config --global --add safe.directory /home/runner/work/ProjectPulse/ProjectPulse
  Deleting the contents of '/home/runner/work/ProjectPulse/ProjectPulse'
  ▶ Initializing the repository
  ▶ Disabling automatic garbage collection
  ▶ Setting up auth
  ▶ Fetching the repository
  ▶ Determining the checkout info
  ▶ Checking out the ref
  /usr/bin/git log -1 --format='%H'
  '11ad61349122841e46366f2e15d0ab92a0eb5f8a'
> Set up Node.js                                                     1s
> Install Server Dependencies                                        1s
```

fig: white box Testing github workflow pt 2



whitebox-test
succeeded 10 minutes ago in 2m 27s

```
> Set up job                                                         1s
> Checkout Code                                                      7s
> Set up Node.js                                                     1s
> Install Server Dependencies                                        1s
> Start Backend (Express) in Background                              0s
> Install Frontend Dependencies                                  1m 12s
> Start Frontend (React) in Background                               0s
> Test raw curl output                                               0s
> Wait for Backend to be Ready (Port 8800)                          30s
```

fig: white box Testing github workflow pt 3

14. Work Division Summary

Team Member	**Contribution**
Tauha Imran	WBS/Gantt preparation, testing, sprint management, team management, database modeling, documentation, and version control.
Minahil Ali	Backend APIs (tasks, comments), Frontend development (project dashboard, notifications), database modeling, diagrams
Nabeed Haider	Backend APIs (tasks, comments), testing, Authentication system, profile management, testing, documentation

15. Lessons Learned

- Importance of clear sprint planning and deliverable tracking.
- Early architecture diagrams prevent major redesigns.
- A proper GitHub branching strategy prevents merge conflicts.
- Black box testing early saves debugging time later.
- Need better time management for overlapping sprints.