

PM-Flow: Real-time Collaborative Project Management System

PM-Flow is a feature-rich, real-time collaborative project management system developed to streamline the way teams handle complex workflows and project tracking. It is designed to support the full lifecycle of project execution — from initial planning and team assignment to task delegation, progress monitoring, communication, and reporting.

In modern development environments, where remote collaboration and agile processes are the norm, PM-Flow serves as a centralized platform that offers seamless integration of project and task management with real-time updates and secure user access. It supports multiple user roles including Admins, Project Managers, and Team Members, each with tailored access and responsibilities.

This system enables users to:

- Create and manage projects and tasks with relevant data such as priority, status, due dates, and assignees.
- Collaborate in real-time through comment threads and project-level chat interfaces.
- Monitor progress visually, filter tasks, and receive notifications or updates based on team activity.
- Access a responsive, user-friendly interface accessible across devices.

PM-Flow is a full-stack application designed using Spring Boot for the backend and React.js for the frontend. It emphasizes best practices in security through JWT-based authentication, and scalable architecture using RESTful APIs and MySQL for data persistence.

The development of this platform exposes students to real-world engineering practices such as:

- Version control and team collaboration with Git & GitHub
- API design and documentation
- Authentication and role-based access control
- Component-based frontend development and routing
- Real-time functionality using polling technique

Prerequisites

- **Frontend**

- React JS
- HTML
- CSS
- Bootstrap
- Visual Studio Code
- Node.js
- npm

- **Backend**

- Java
- Maven
- Spring Boot
- MySQL
- Eclipse IDE
- Postman for API testing

- **Versioning**

- Git
- GitHub

Project Repository - <https://github.com/EbeyJoeRegi/PM-Flow>

Steps to run the Application

1. **Clone the Repository**

- `git clone https://github.com/EbeyJoeRegi/PM-Flow`
- `cd PM-Flow`

2. **Set Up the Frontend**

- `cd Frontend`
- Install Dependencies
 - `npm install`
- Start the Development Server
 - `npm run dev`

- The frontend will be available at: `http://localhost:5173`

3. Configure Environment Variables (.env)

- Backend API Connection
 - `VITE_API_BASE_URL = http://localhost:8080`

4. Open the Backend in Eclipse

- File → Import → Existing Maven Project → Select the “Backend” folder

5. Create MySQL Database

- Open MySQL terminal or GUI (e.g., MySQL Workbench) and execute:
- `CREATE DATABASE pmflow;`
- `USE pmflow;`
- `INSERT INTO user (email, first_name, last_name, password_hash, role, username)`
`VALUES ('admin@incture.com', 'Admin', 'PM FLOW',`
`'$2a$10$2e55q3vpUrxSzTjZIXdwp.c9UaTlut6lZiSctLbhDsyCia.NwAiBi', 'ADMIN',`
`'admin');`

6. Configure application.properties

- Located at: `src/main/resources/application.properties`
- Update the file with your local MySQL credentials

7. Run the Application in Eclipse:

- Right-click on `PmflowApplication.java`
- Choose Run As → Spring Boot Application
- Console should display: Started `PmflowApplication` in ... seconds

8. Access & test the APIs

- Open Postman and follow the steps
- First, hit the register/login endpoint to obtain a JWT token.
- Add the token in the headers of secured endpoints: Authorization: Bearer
`<your_jwt_token>`
- All other endpoints should use this authorization and method is executed

NOTE: The Manager role is not available by default. An Admin must promote a member to the Manager role.

Default Admin Credentials:

- Email: `admin@incture.com`
- Password: `PASS@word123`

Project Structure

PM-FLOW/

```
|— FrontEnd/
|  |— .env
|  |— index.html
|  |— package.json
|  |— package-lock.json
|  |— vite.config.js
|  |— node_modules/
|  |
|  |— public/
|  |  |— logo/
|  |    |— Logo.png
|  |    |— PM Flow.png
|  |
|  |— src/
|  |  |— App.jsx
|  |  |— config.jsx
|  |  |— main.jsx
|  |  |
|  |  |— api/
|  |    |— adminApi.jsx
|  |    |— commonApi.jsx
|  |    |— managerApi.jsx
|  |    |— teamMemberApi.jsx
|  |  |
|  |  |— components/
|  |    |— Pagination.jsx
|  |  |
|  |  |— pages/
|  |    |— CollaborationProjects.jsx
|  |    |— Home.jsx
|  |    |— Login.jsx
```

```
| | | └─ NotFound.jsx
| | | └─ ProjectChatPage.jsx
| | | └─ Register.jsx
| | |
| | | └─ admin/
| | |   └─ AdminDashboard.jsx
| | |   └─ ProjectDetails.jsx
| | |   └─ Projects.jsx
| | |     └─ Users.jsx
| | |
| | | └─ manager/
| | |   └─ ManagerDashboard.jsx
| | |   └─ managerProjectDetail.jsx
| | |   └─ managerProjects.jsx
| | |     └─ managerTaskDetail.jsx
| | |
| | | └─ member/
| | |   └─ AssignedTasks.jsx
| | |   └─ MemberDashboard.jsx
| | |     └─ ProjectCollaboration.jsx
| | |
| | └─ redux/
| |   └─ store.jsx
| |   └─ userSlice.jsx
| |
| └─ routes/
|   └─ AppRoutes.jsx
|   └─ ProtectedRoute.jsx
|
| └─ styles/
|   └─ Admin.css
|   └─ collaborationProjects.css
|   └─ Home.css
|   └─ login.css
```

```
| | | └─ managerDashboard.css
| | | └─ managerProjectDetail.css
| | | └─ managerProjects.css
| | | └─ managerTaskDetail.css
| | | └─ Member.css
| | | └─ notfound.css
| | | └─ projectChatPage.css
| | | └─ ProjectCollab.css
| | | └─ Register.css
| | | └─ style.css
| | |
| | └─ utils/
| |   └─ Helper.js
| |
| └─ Backend/
|   └─ pom.xml
|   └─ mvnw
|   └─ mvnw.cmd
|   └─ HELP.md
|   └─ target/
|   └─ src/
|     └─ main/
|       └─ java/
|         └─ com/example/pmflow/
|           └─ PmflowApplication.java
|           |
|           └─ controller/
|             └─ AuthController.java
|             └─ ChatController.java
|             └─ ProjectController.java
|             └─ TaskController.java
|             └─ UserController.java
|             |
|             └─ dto/
```

```
| | | | | | |— AdminUpdateTaskRequest.java
| | | | | | |— AdminUpdateUserRequest.java
| | | | | | |— AuthRequest.java
| | | | | | |— AuthResponse.java
| | | | | | |— ChatRequestDTO.java
| | | | | | |— ChatResponseDTO.java
| | | | | | |— ChatSummaryDTO.java
| | | | | | |— MemberProjectDTO.java
| | | | | | |— ProjectCreateRequestDTO.java
| | | | | | |— ProjectDetailDTO.java
| | | | | | |— ProjectSummaryDTO.java
| | | | | | |— ProjectUpdateRequestDTO.java
| | | | | | |— RegisterRequest.java
| | | | | | |— StatusEndDateUpdateDTO.java
| | | | | | |— TaskRequest.java
| | | | | | |— TaskResponse.java
| | | | | | |— TeamMemberDTO.java
| | | | | | |— UpdateTaskRequest.java
| | | | | | |— UserDTO.java
| | | | | |
| | | | | |— entity/
| | | | | | |— ChatMessage.java
| | | | | | |— Project.java
| | | | | | |— ProjectStatus.java
| | | | | | |— Role.java
| | | | | | |— Task.java
| | | | | | |— User.java
| | | | | |
| | | | | |— enums/
| | | | | | |— TaskPriority.java
| | | | | | |— TaskStatus.java
| | | | | |
| | | | | |— repository/
| | | | | | |— ChatMessageRepository.java
```

```
| | | | | | |— ChatProjectRepository.java
| | | | | | |— ProjectRepository.java
| | | | | | |— RoleRepository.java
| | | | | | |— TaskRepository.java
| | | | | | |— UserRepository.java
| | | | | |
| | | | | |— security/
| | | | | | |— JwtAuthFilter.java
| | | | | | |— JwtService.java
| | | | | | |— SecurityConfig.java
| | | | | | |— TokenBlacklistService.java
| | | | | |
| | | | | |— service/
| | | | | | |— AuthService.java
| | | | | | |— ChatService.java
| | | | | | |— ProjectService.java
| | | | | | |— TaskService.java
| | | | | | |— UserDetailsServiceImpl.java
| | | | | | |— UserService.java
| | | | | |
| | | | | |— resources/
| | | | | | |— static/
| | | | | | |— templates/
| | | | | | |— application.properties
| | | | | |
| | | | | |— test/
| | | | | | |— java/
| | | | | | |— com/example/pmflow/
| | | | | | |— PmflowApplicationTests.java
| | | | | | |— controller/
| | | | | | |— AuthControllerTest.java
| | | | | | |— ChatControllerTest.java
| | | | | | |— ProjectControllerTest.java
| | | | | | |— TaskControllerTest.java
| | | | | | |— UserControllerTest.java
```


PM-FLOW Frontend

Project Requirements

1. Functional Requirements

❖ Authentication & Authorization UI

1. Implement Login and Registration forms with validation.
2. Display appropriate success and error messages on login/register.
3. Use role-based protected routes for Admin, Manager, and Team Member.
Automatically redirect users to their role-specific dashboards after login.

❖ Admin Panel

- Dashboard
 - Display total number of Users, Active Projects, List of the all the Project
- User Management
 - View all registered users (Admins, Managers, Team Members).
 - Edit user profiles and update their roles.
 - Remove users from the system.
- Project Management
 - Create new projects with required fields (name, manager, Due date, Team members).
 - View list of all projects
 - Edit or delete existing project.
 - View project details including associated team members.

❖ Manager Panel

1. Dashboard
 - Overview of all projects assigned to the manager
 - Display project status meter.
 - List of top 5 projects with upcoming due dates.
2. Project Management
 - View all projects where the user is the assigned manager.
 - Filter projects by status.
 - Search projects by name.
 - Sort projects by start and end dates.

3. Project Detail

- Show complete project details including Description and Team members
- Update the project status and due date
- Create new tasks under the selected project.
- Filter tasks by assignee, search tasks by task name.

4. Task Detail

- View and edit task details: name, description, due date, priority, status.
- Access the 1:1 comment/chat section with the task assignee.
- Real-time updates via polling (every 5 seconds) for messages.

5. Collaboration

- Engage in group chat with all team members assigned to a project.

❖ Member Panel

1. Dashboard

- 5 most recent tasks assigned to the user.
- Task status overview (Pie chart).
- Top 5 tasks with upcoming due dates.

2. Tasks Assigned

- View all tasks assigned to the users.
- Search tasks by name.
- Sort tasks by Priority and Status

3. Task Detail

- View complete details of an individual task
- Update the task status.
- Access the 1:1 chat section with the manager.
- Polling-based real-time chat updates

4. Collaboration

- Chat with the project manager and team members assigned to the same project.

2. Application Structure

1. Redux – State Management

- managing global application state particularly for storing authenticated user details, role and tokens.
- Actions are used to Handle logout to clear the redux store and local storage.

2. Pagination Component

- Custom reusable component for Ensuring better UI handling for long lists such as projects or tasks.
- Includes Previous, Next, and Page Numbers

3. Util.js – Utility Functions

- Contains helper functions reused across components.
- Date Formatting, Status & Priority Color Mapping.

4. API Integration (Axios)

- Configured axios with BASE_URL from .env
- API files associated with each user roles.

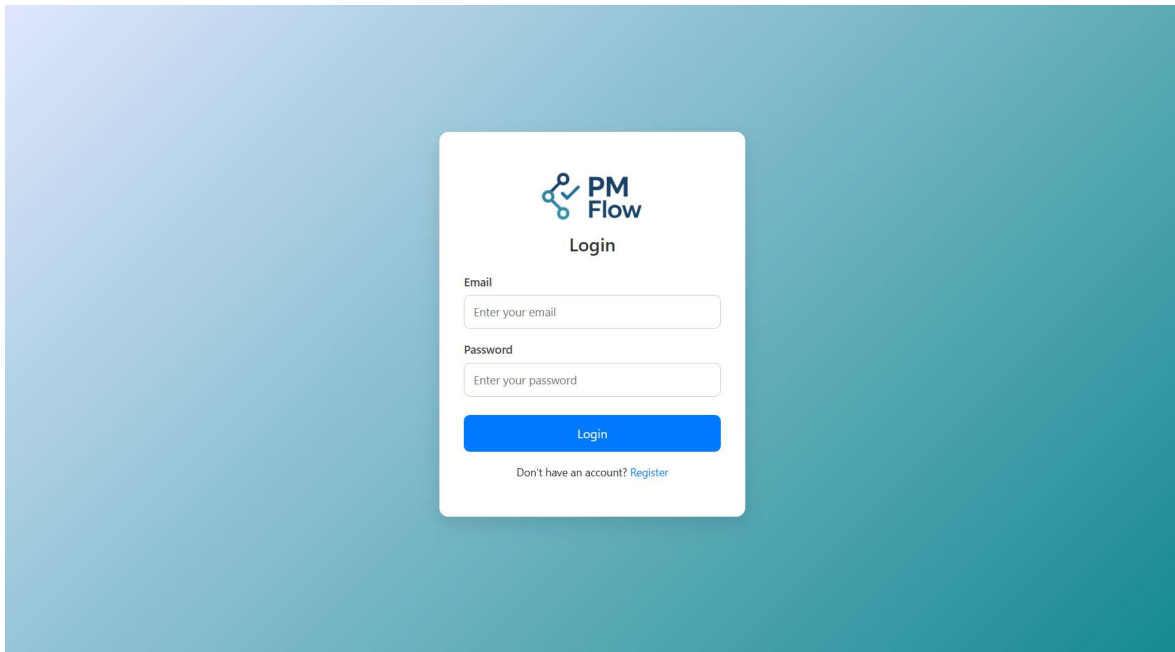
5. Routes for Navigation & Access Control

- Defines user-specific routes and protects them based on roles.
- Unauthorized users redirected to login page

3. Tools & Libraries Used

- **Node.js** - JavaScript runtime needed to run/build React apps.
- **Redux Toolkit** - Simplifies global state management in complex applications.
- **Axios** - For making HTTP requests to backend APIs.
- **React Toastify** - To display toast notifications (success, error, etc.).
- **React Icons** - Includes icon packs like Font Awesome and Material Icons.
- **Recharts** - For rendering charts (e.g., pie charts in dashboards).
- **Bootstrap** - pre-built responsive UI components and grid system.

1. LOGIN



The Login Page is the entry point for all users of the PM FLOW application, including Admins, Managers, and Team Members. It facilitates secure user authentication and role-based redirection to their respective dashboards upon successful login.

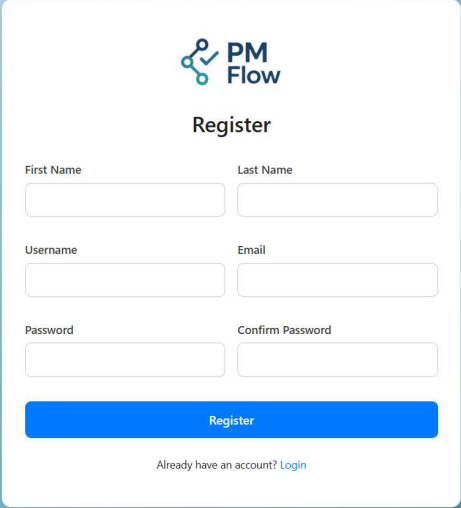
Key Components

- **Email Input Field:** Accepts the user's registered email.
- **Password Input Field:** Accepts the user's password (masked input).
- **Login Button:** Initiates the authentication process.
- **Error Message Display:** Shows validation errors or incorrect credentials.
- **Role-Based Redirection:** Redirects the user to the appropriate dashboard based on their role:
 - Admin → /admin
 - Manager → /manager
 - Member → /member

Error Handling

- Incorrect credentials
- Server unreachable
- Empty input fields

2. REGISTRATION

The image shows a registration form for the PM Flow system. The form is centered on a blue gradient background. At the top of the form is the PM Flow logo, which consists of a stylized blue icon of three people connected by lines, followed by the text "PM Flow". Below the logo is the heading "Register". The form contains six input fields arranged in three rows: "First Name" and "Last Name" in the first row, "Username" and "Email" in the second row, and "Password" and "Confirm Password" in the third row. Each field is a simple white rectangle with a thin border. Below the input fields is a prominent blue button with the word "Register" in white text. At the very bottom of the form, there is a small link that says "Already have an account? Login".

The Registration Page allows new users to create an account in the PM FLOW system. All registrations through this page are automatically assigned the "Member" role.

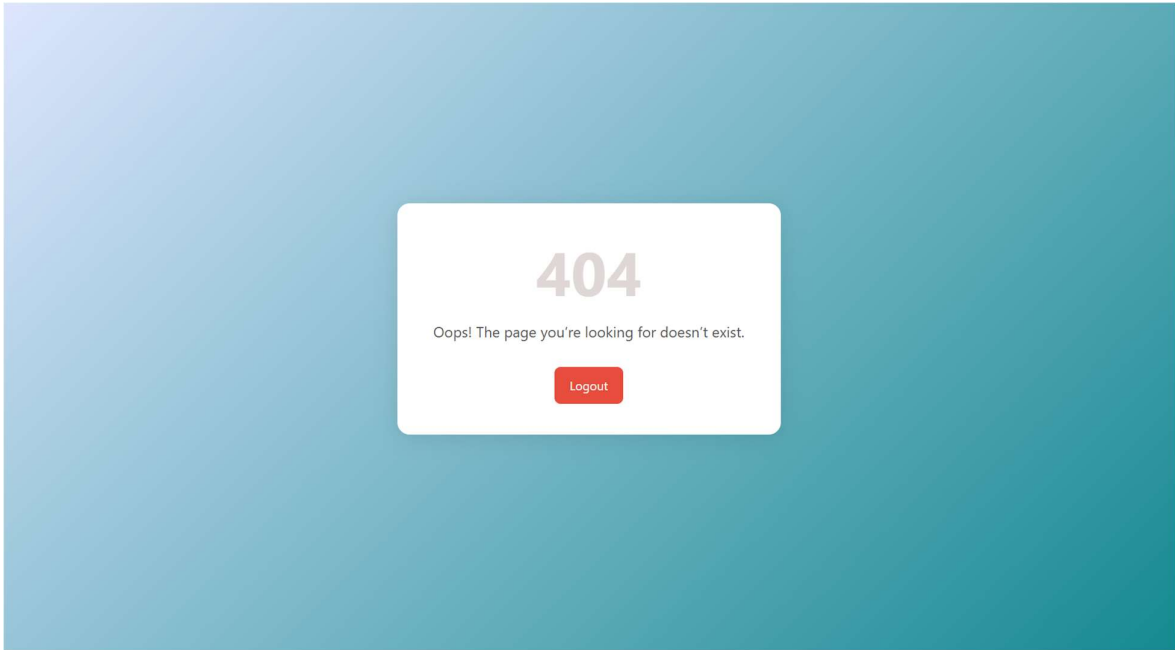
Key Components

- **First Name Field:** Text input for the user's first name.
- **Last Name Field:** Text input for the user's last name.
- **Username Field:** Unique identifier for the user (used for login and display).
- **Email Field:** Must be a valid email format.
- **Password Field:** Password input (masked).
- **Confirm Password Field:** Verifies the entered password matches.

Error Handling

- Missing or empty fields.
- Email or Username already in use.
- Email should contain **@incture.com**.
- Password mismatch.
- Server or network issues

3. PAGE NOT FOUND



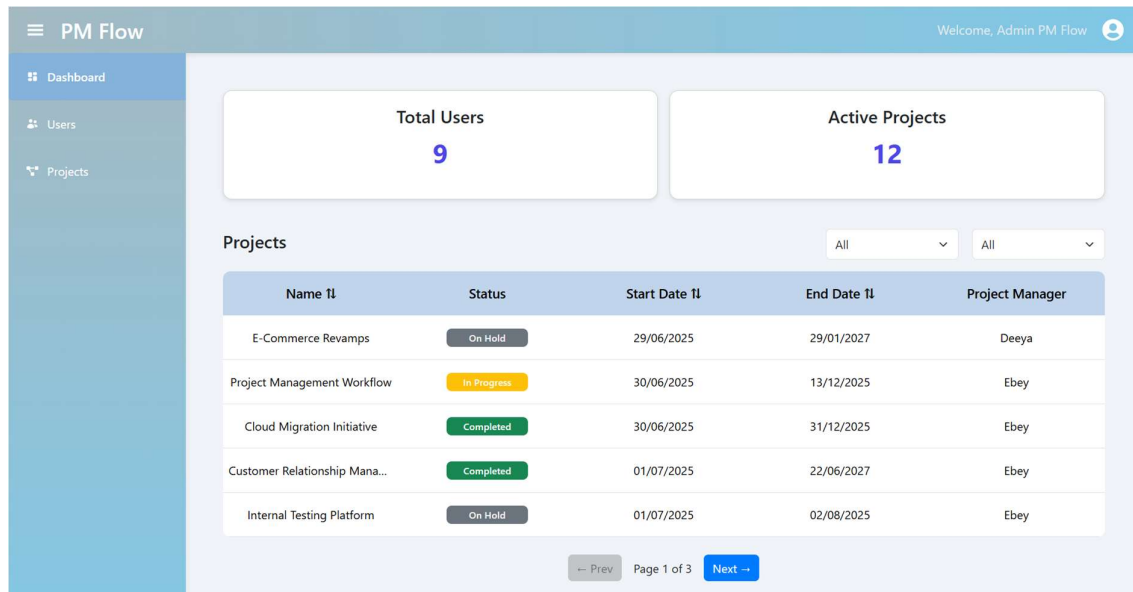
The Page Not Found page is a fallback screen that appears when a user navigates to an undefined or invalid route within the PM FLOW application. It improves user experience by clearly informing users that the requested page does not exist and provides navigation options to recover gracefully.

Key Components

- **404 Error Message:** A large, clear message like “404 - Page Not Found”.
- **Description Text:** A short explanation - “The page you are looking for does not exist.”
- **Logout Redirect Button:** A button to navigate the user back to the Login page.

4. ADMIN

DASHBOARD OVERVIEW

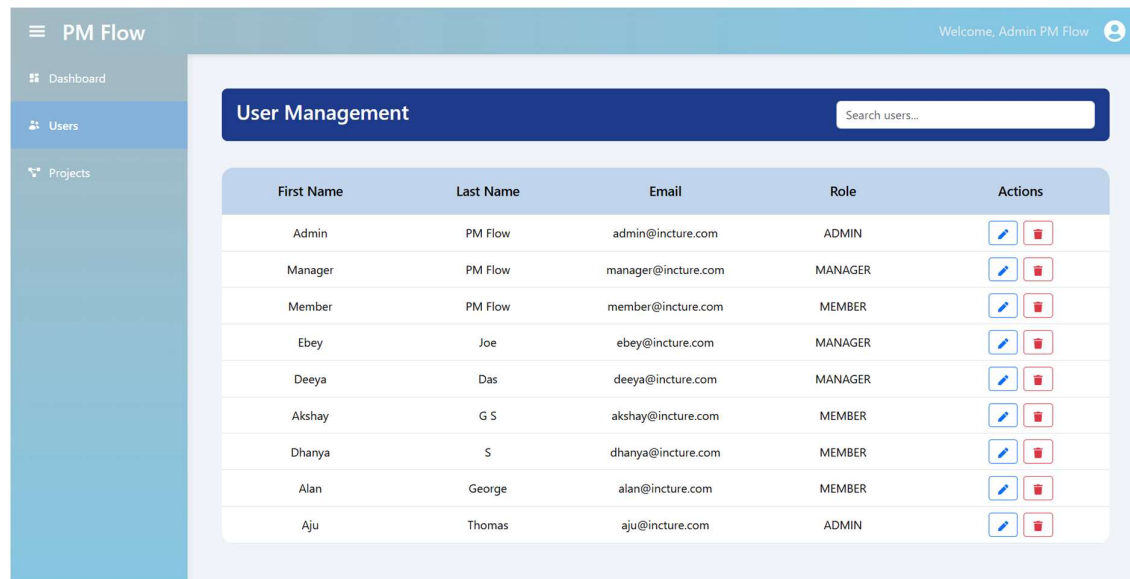




















The Admin Dashboard serves as the central control panel for administrators. It provides a quick overview of key metrics and a dynamic table for managing and monitoring all projects in the system.

Key Components

- **Total Users:** Displays the count of all registered users.
- **Active Projects:** Shows the number of projects currently marked as "active".
- **Projects List:** A paginated list of all projects in the system.
- **Sorting:** Allows sorting the project list by:
 - Project Name
 - Start Date
 - Due Date
- **Filtering:**
 - By Project Manager using a dropdown selector.
 - By Project Status

USER MANAGEMENT



First Name	Last Name	Email	Role	Actions
Admin	PM Flow	admin@incture.com	ADMIN	 
Manager	PM Flow	manager@incture.com	MANAGER	 
Member	PM Flow	member@incture.com	MEMBER	 
Ebey	Joe	ebey@incture.com	MANAGER	 
Deeya	Das	deeya@incture.com	MANAGER	 
Akshay	G S	akshay@incture.com	MEMBER	 
Dhanya	S	dhanya@incture.com	MEMBER	 
Alan	George	alan@incture.com	MEMBER	 
Aju	Thomas	aju@incture.com	ADMIN	 

The **User Management** section allows administrators to view, edit, and delete all registered users in the system through an intuitive and interactive table. This feature ensures effective role control and user lookup.

Key Components

- **User Table Overview** - Displays all users in a tabular layout.
- **Search Functionality** – Supports real-time filtering by User's name and email.
- **Edit Functionality** - Inline editing of First Name, Last Name Role
- **Delete Functionality** - Clicking the delete icon (trash) removes the selected user.
- **Toast Notifications** - After a successful edit or delete, a toast appears with relevant feedback.

PROJECT MANAGEMENT

PM Flow

Dashboard

Users

Projects

Welcome, Admin PM Flow

Projects

Search

+ New Project

Name	Manager	Status	End Date	Actions
E-Commerce Revamps	Deeya	ON HOLD	29-01-2027	<div><div></div><div></div></div>
Project Management Workflow	Ebey	IN PROGRESS	13-12-2025	<div><div></div><div></div></div>
Cloud Migration Initiative	Deeya	COMPLETED	31-12-2025	<div><div></div><div></div></div>
Customer Relationship Manage...	Manager	COMPLETED	22-06-2027	<div><div></div><div></div></div>
Internal Testing Platform	Ebey	ON HOLD	02-08-2025	<div><div></div><div></div></div>
HR Onboarding Tool	Ebey	IN PROGRESS	02-07-2025	<div><div></div><div></div></div>
Support Ticket Dashboard	Deeya	ON HOLD	21-12-2025	<div><div></div><div></div></div>
Sales Forecasting Module	Manager	COMPLETED	09-08-2025	<div><div></div><div></div></div>
SmartTask – Project & Task Manag...	Ebey	COMPLETED	23-07-2025	<div><div></div><div></div></div>
MedicarePro – Patient Appointme...	Deeya	IN PROGRESS	19-07-2025	<div><div></div><div></div></div>

Prev

Page 1 of 2

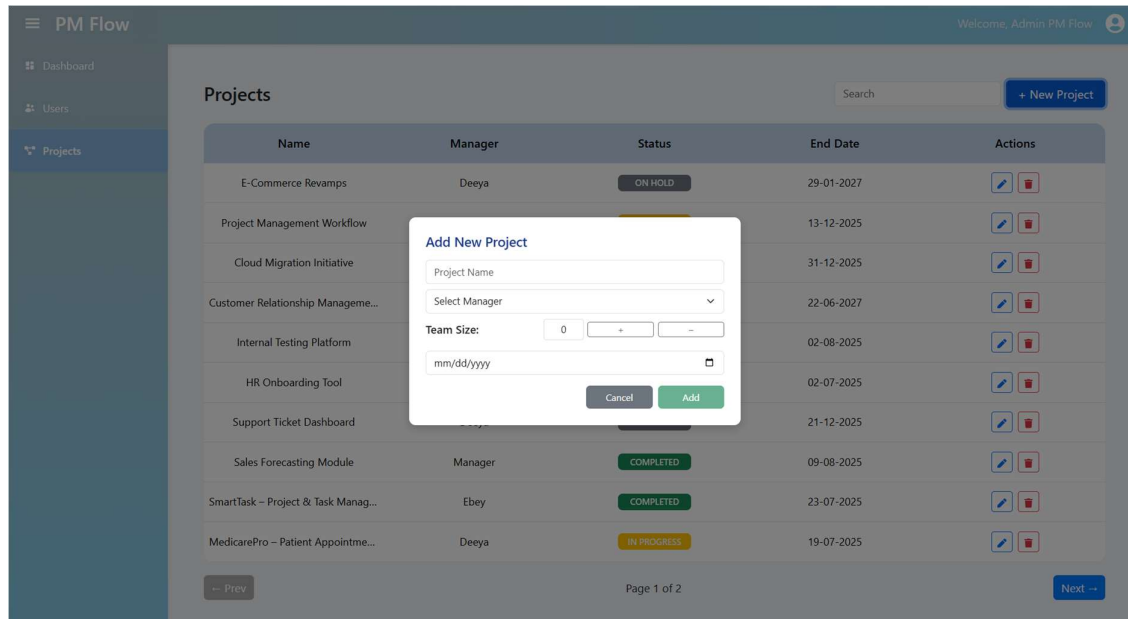
Next

The Project Management section allows administrators to manage all projects across the organization. It allows viewing, editing, deleting, and adding projects through an interactive, user-friendly interface. Admins can monitor project assignments, track timelines and the allocation of managers.

Key Components

- **Project Table Overview** - Showing Project Name, Assigned Manager, Status, End Date, and Action buttons.
- **Search Capability** - Real-time search bar to filter projects by name.
- **Edit Functionality** - Edit icon triggers inline form for editing project details. Editable fields include Project Name, Assigned Manager and End Date.
- **Delete Functionality** - Clicking the Delete icon removes the selected project from the system.
- **Add New Project** - “+ New Project” button opens a modal to create and submit a new project.
- **Toast Notifications** - Feedback is provided for successful edit or delete operations via toast messages.

ADD NEW PROJECT

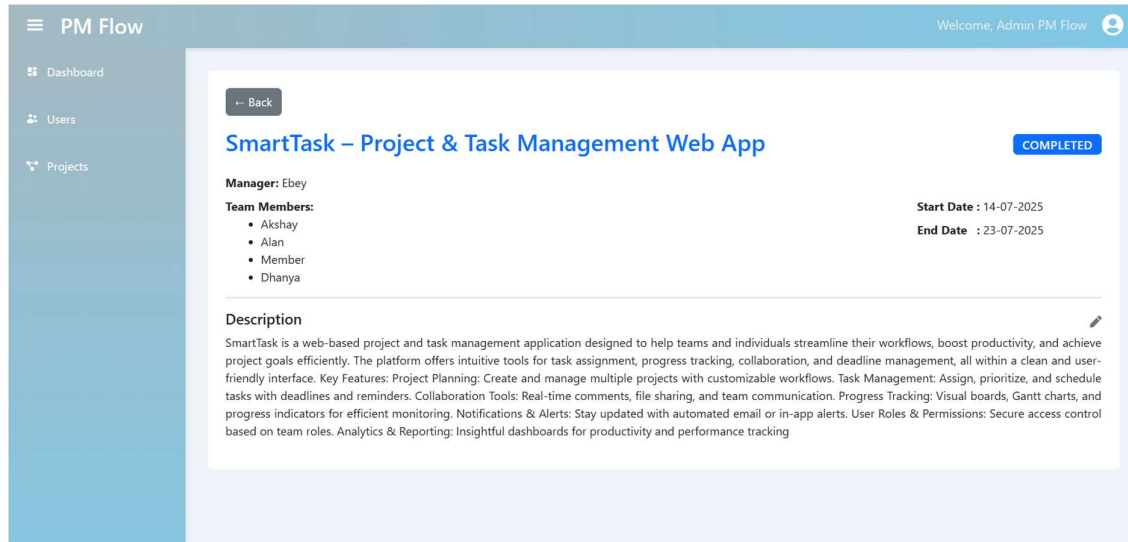


The “+ New Project” feature enables administrators to add and register new projects into the system. This functionality ensures that all necessary data is collected in a consistent and user-friendly manner.

Key Components

- Accepts the following necessary details from the admin such as Project Name, Manager, Team Members and the Due Date
- The project description is initialized as null by default.
- The project status is automatically set to “NOT STARTED” on creation.
- Toast Notifications provide instant feedback.

PROJECT DETAILS



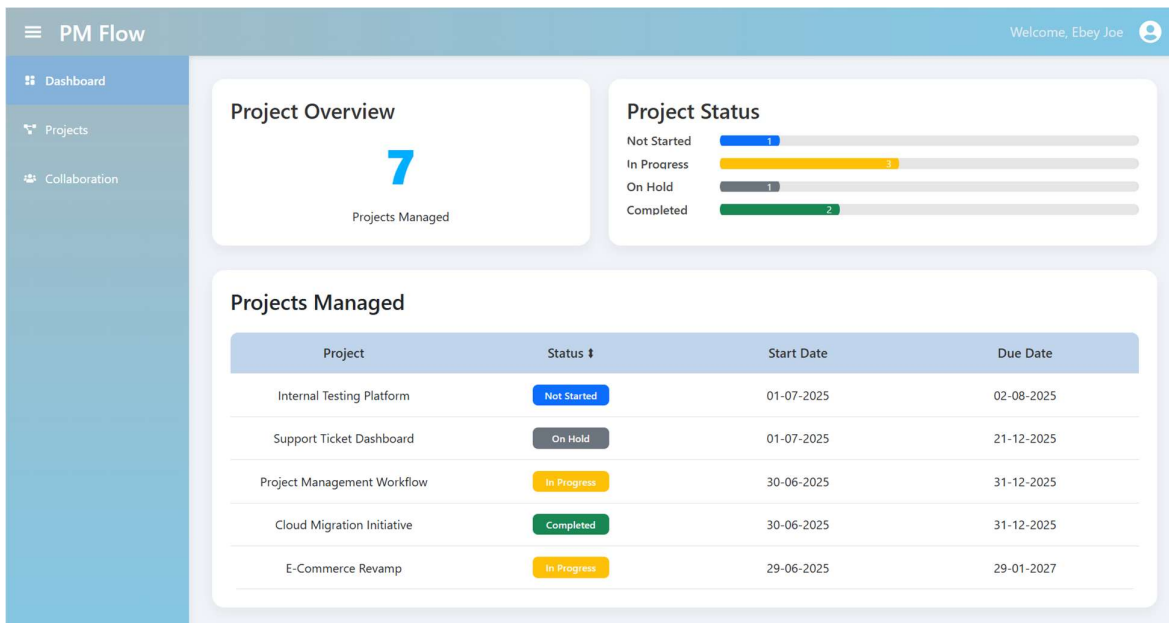
The Project Details View presents a comprehensive overview of an individual project's key attributes, enabling administrators to monitor progress and manage content such as the project description.

Key Components

- Displays detailed information about the selected project, including name, manager, status, and dates.
- Shows a list of team members assigned to the project.
- Allows to add or update the project description.
- Toast notifications provide real-time feedback when the description is successfully updated or fails due to an error.

5. MANAGER

DASHBOARD



The Manager Dashboard is the central hub for project managers in PM FLOW. It provides a high-level overview of all projects managed by the logged-in manager, helping them monitor project progress, track closest deadlines.

Key Components

- **Projects Managed** - Displays the total number of projects currently managed by the user. Dynamically fetched from the backend using the manager's ID.
- **Project Status (Progress Meter)** - Visual representation showing distribution of projects by status:
 - Not Started
 - In Progress
 - Completed
 - On Hold
- **Projects Table** - A sortable table of managed projects whose end dates are approaching soon.
- **Sorting Options** - Sort by status

PROJECT MANAGEMENT

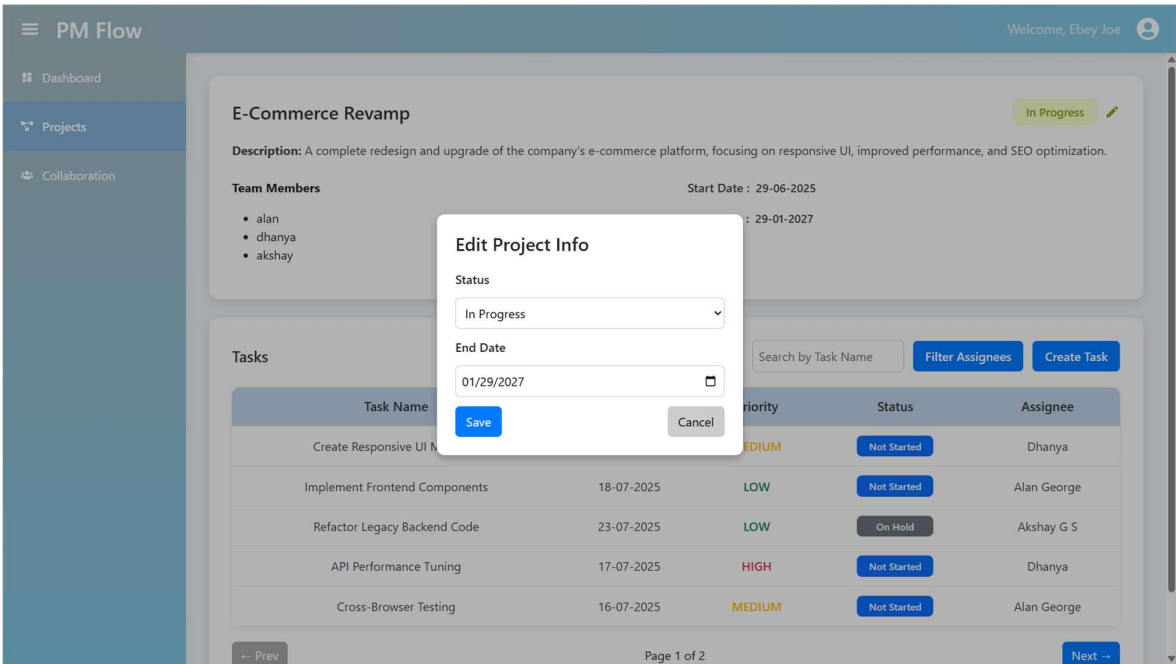
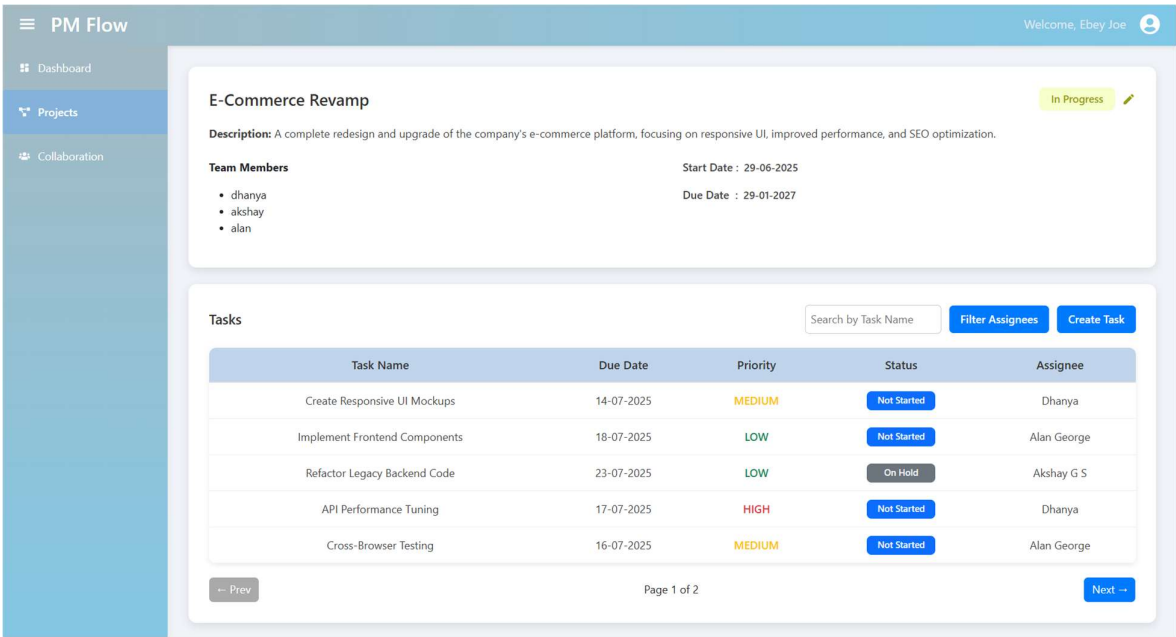
Name	Status	Start Date ↑	Due Date ↑
E-Commerce Revamp	In Progress	29-06-2025	29-01-2027
Project Management Workflow	In Progress	30-06-2025	31-12-2025
Cloud Migration Initiative	Completed	30-06-2025	31-12-2025
Customer Relationship Management...	Completed	01-07-2025	22-06-2027
Internal Testing Platform	Not Started	01-07-2025	02-08-2025
HR Onboarding Tool	In Progress	01-07-2025	02-07-2025
Support Ticket Dashboard	On Hold	01-07-2025	21-12-2025
SmartTask – Project & Task Manage...	Not Started	14-07-2025	23-07-2025
EduConnect – Online Learning Platfo...	Not Started	14-07-2025	24-07-2025
MedicarePro – Patient Appointment ...	Not Started	14-07-2025	19-07-2025

The Project Management Page provides managers with a structured view of all the projects they oversee. It includes search, filter, sorting, and pagination features, allowing for easy navigation and focused project tracking.

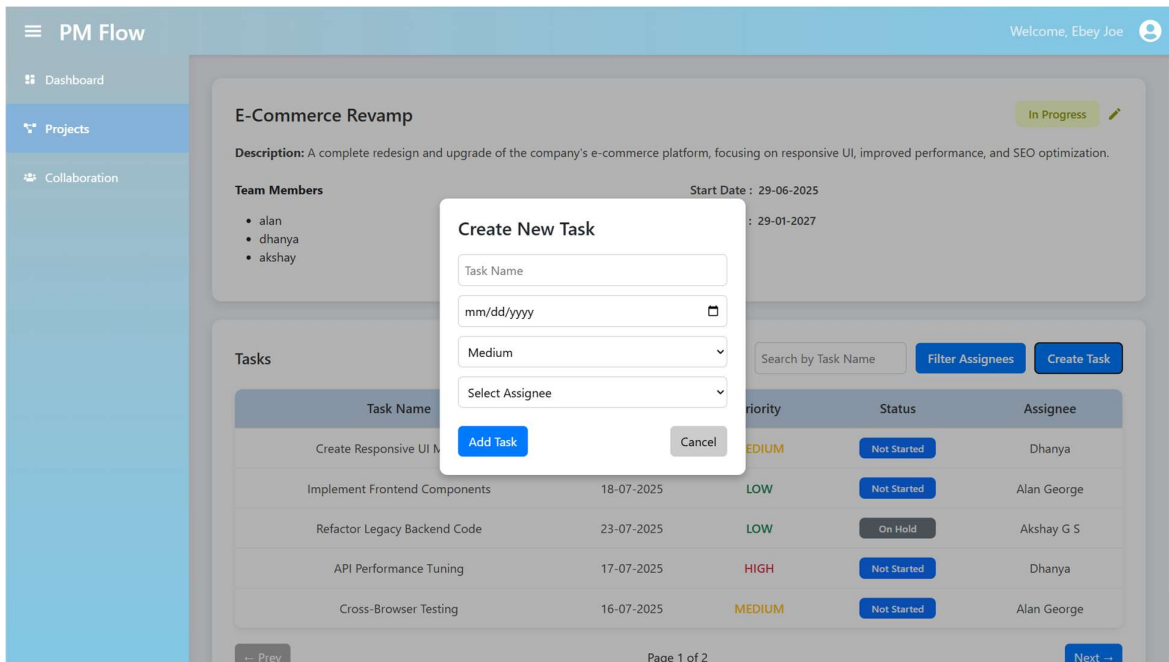
Key Components

- **Search by Project Name**
 - Input field for real-time search.
 - Filters the table to show projects whose names match the entered text.
- **Filter by Status** - Dropdown to filter projects based on their current status
- **Projects Table** - Displays all projects assigned to the currently logged-in manager.
- **Sorting** - by Start Date and End Date.
- **Navigation** - Clicking a row or project name navigates to the Project Detail Page.
- If no projects assigned to the manager or no results match search/filter criteria it shows a “No projects found” message.
- **Pagination** - Displays 10 projects per page.

PROJECT DETAIL PAGE



Edit Project Info



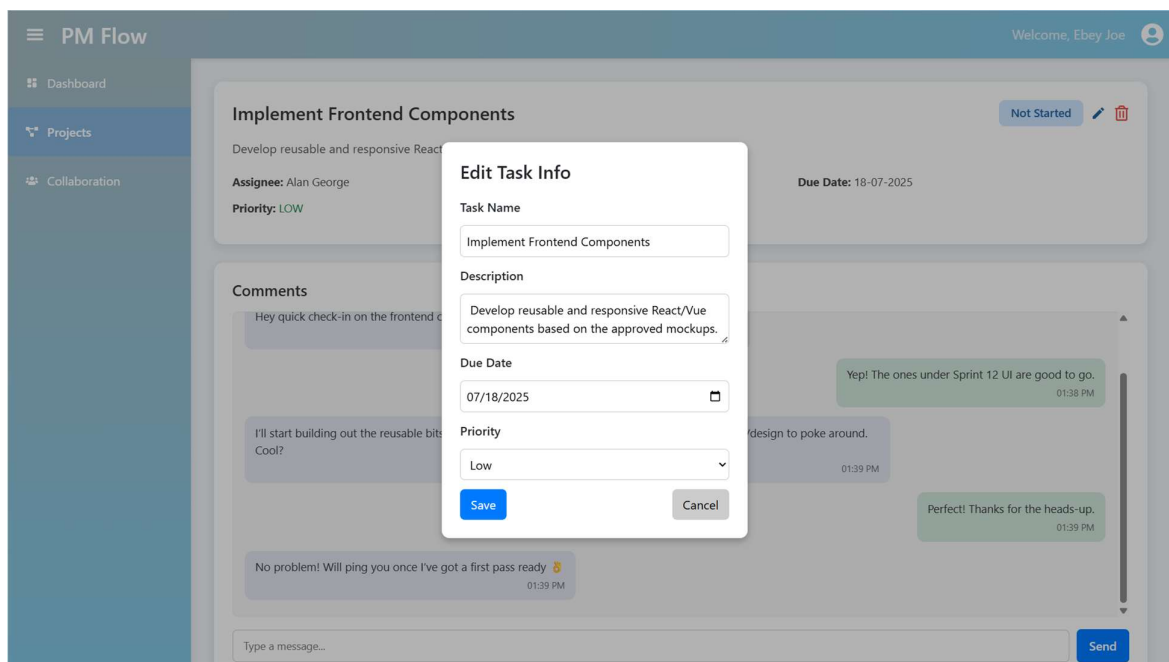
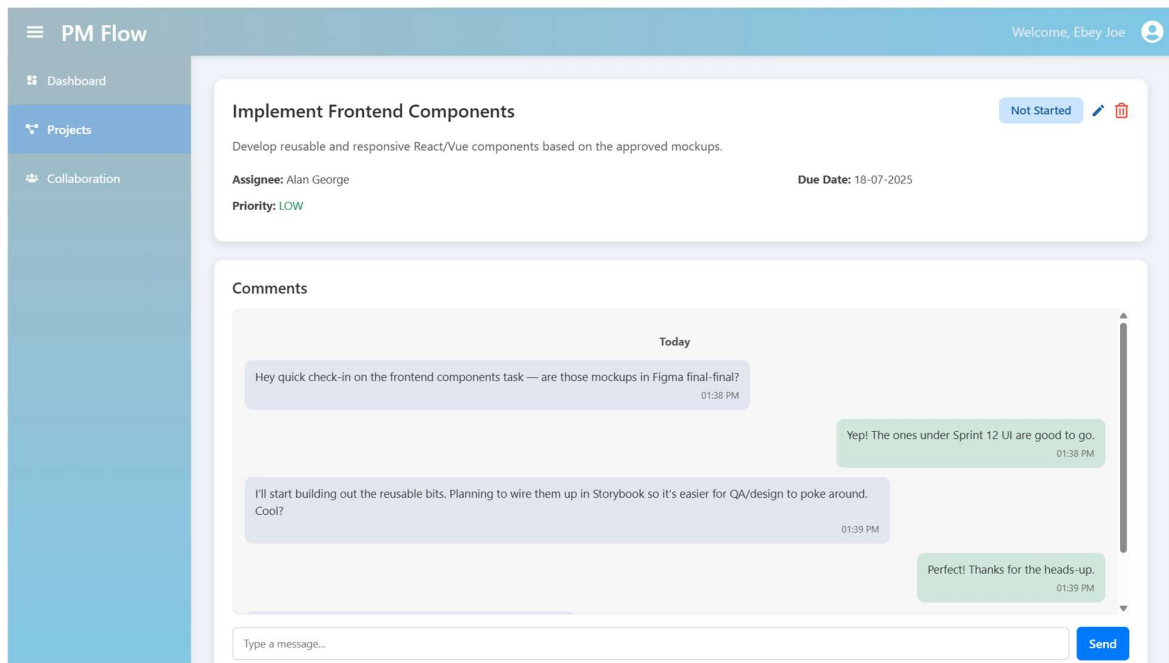
Create a new Task

The Project Detail Page provides an in-depth view of a selected project, including general project information, assigned team members, and associated tasks. Managers can update project details, create new tasks, filter tasks, and monitor task progress effectively from this page.

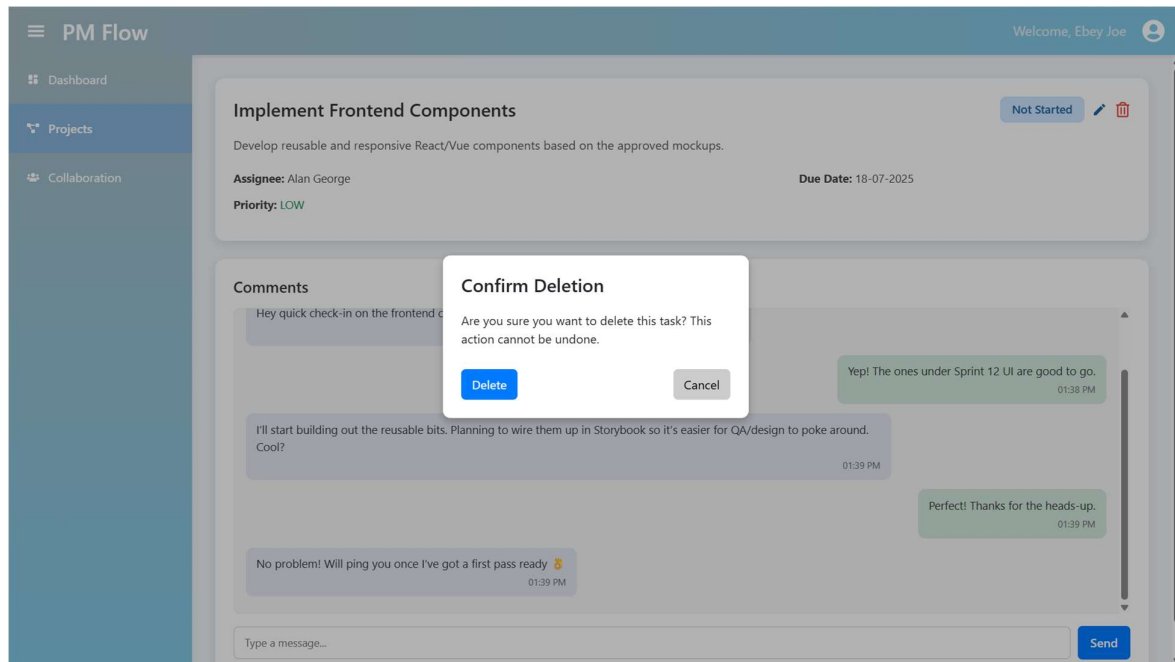
- **Project Information** - To view and update project-level information such as name, description, status, timeline, and team members.
- **Edit Project Info**
 - An Edit button next to the project status.
 - Opens a modal popup to edit - Project Status (dropdown), Due Date (date picker).
 - Validation - At least one field must change to enable the Save button.
- **Task Management** - To view, filter, search, create, and manage tasks under the current project.
 - Sorting: Columns sortable by Due Date, Priority, and Status.
 - Search Functionality - Search by Task Name (input box). Filters task table in real-time.
 - Filter by Assignee
 - Checkbox list of project team members. Allows multi-selection of assignees to filter the task list.
- **Create Task Modal**
 - A new task is created by providing the task name, due date, priority, and assignee (team members).

- The default values are: description is null, and status is 'Not Started'.
- Validation - All fields are required before enabling the Create button.

TASK DETAIL PAGE



Edit Task Info



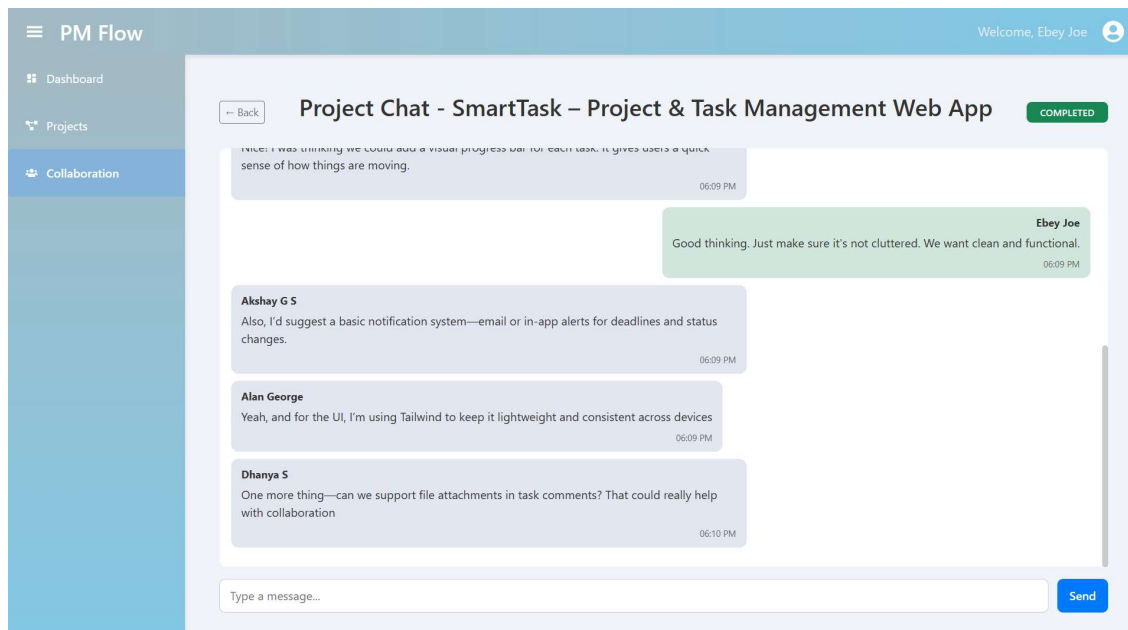
Delete Task

The Task Detail Page provides an in-depth view of a specific task, allowing managers to view and edit task details, delete the task if needed, and engage in real-time 1:1 chat (under the "Comments" section) with the assigned team member.

Key Components

- Displays task details including Task Name, Description, Status, Assignee, Priority, and Due Date.
- Edit icon allows updating Task Name, Description, Due Date, and Priority.
- Save button is enabled only if at least one field is modified.
- Provides option to delete the task if needed.
- Includes a Comments section for real-time 1:1 chat with the assignee.
- Chat uses polling to fetch new messages every 5 seconds.

COLLABORATION



The Reusable Collaboration Page used between Manager & Members enables seamless team communication by providing a centralized space for group discussions per project. Users can view a list of all projects they are associated with and initiate real-time group chats with other team members and the project manager. This feature fosters collaboration, quick updates, and enhanced project coordination without needing external communication tools.

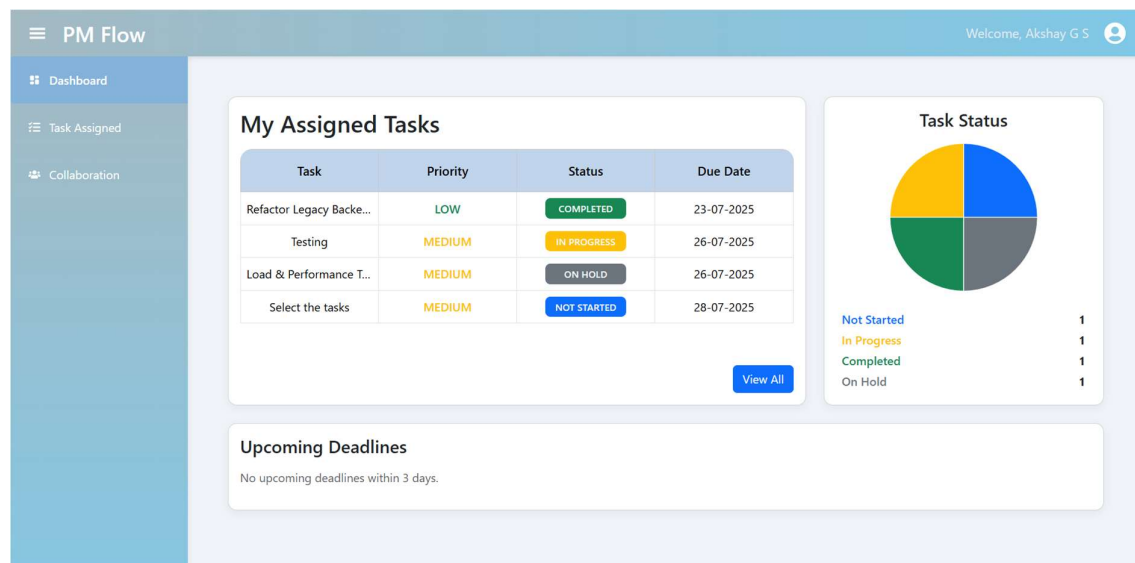
Key Components

- Displays a list of all projects associated with the logged-in user.

- Each project in the list includes the project name and on clicking a project, users are redirected to a dedicated chat page for that specific project.
- The Chat Page shows a real-time group chat interface. Messages are displayed with sender information.
- Polling-based updates fetch new messages every 5 seconds to simulate real-time communication.

6. MEMBER

DASHBOARD



The Member Dashboard is designed to provide individual team members with a concise and interactive overview of their task responsibilities. It ensures quick access to critical information like task priority, deadlines, and task progression through intuitive UI elements.

Key Components:

- **My Assigned Tasks Card**
 - Displays a preview of five tasks assigned to the current member.
 - A "View All" button navigates the user to a comprehensive list of all assigned tasks.
- **Task Status Visualization:**
 - Implemented via a Pie Chart to reflect task distribution by status.
- **Upcoming Deadlines Tracker:**

- Tasks with due dates falling within the next three days are automatically flagged and displayed. This ensures timely awareness and promotes better planning.

TASK ASSIGNED

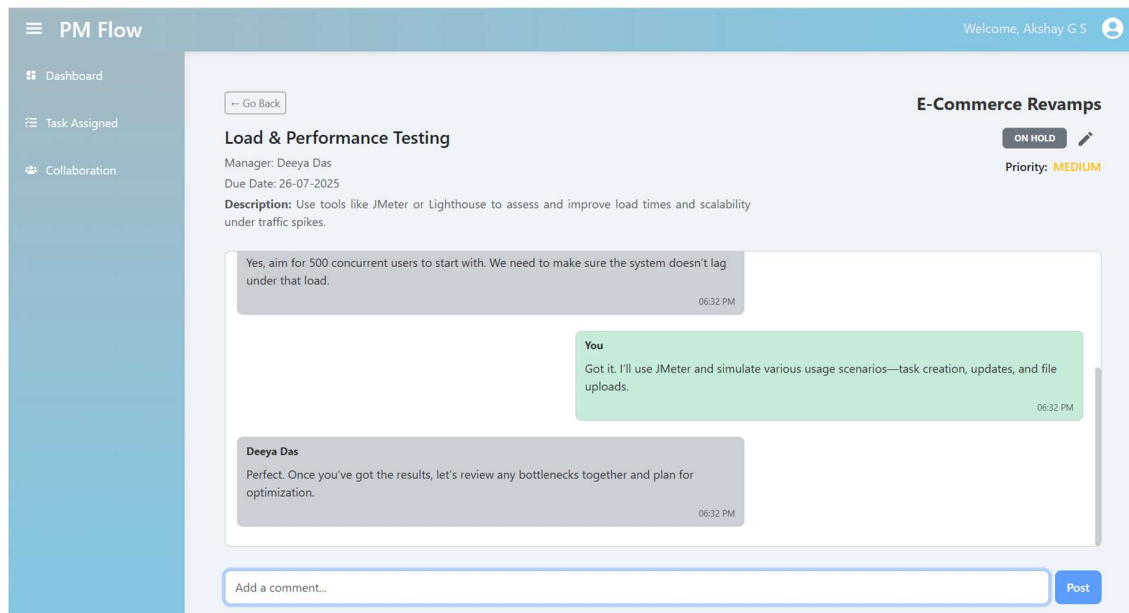
PM Flow					
Welcome, Akshay G S					
Dashboard					
Task Assigned					
Collaboration					
MY TASKS					
Search tasks					
Task	Priority TL	Status TL	Project	Manager	Due Date TL
Refactor Legacy Backen...	LOW	COMPLETED	E-Commerce Revamps	Deeya Das	23-07-2025
Testing	MEDIUM	IN PROGRESS	Sales Forecasting Module	Manager PM Flow	26-07-2025
Load & Performance Tes...	MEDIUM	ON HOLD	E-Commerce Revamps	Deeya Das	26-07-2025
Select the tasks	MEDIUM	NOT STARTED	HR Onboarding Tool	Ebey Joe	28-07-2025
All Tasks Displayed					

The Task Assigned section provides users with a comprehensive view of all tasks allocated to them. This serves as a centralized space for team members to track their responsibilities, monitor progress, and stay informed about project timelines.

Key Features:

- **Structured Task Table** - Displays task information
- **Interactive Sorting Controls** - Users can sort tasks based on priority, status, and due date.
- **Search Functionality** - A real-time search bar enables users to filter tasks by keywords making it easy to locate specific assignments.
- **Clickable Task Rows** that navigate to the Task Detail View, showing full description, manager, and comment/chat interface.

TASK DETAILS

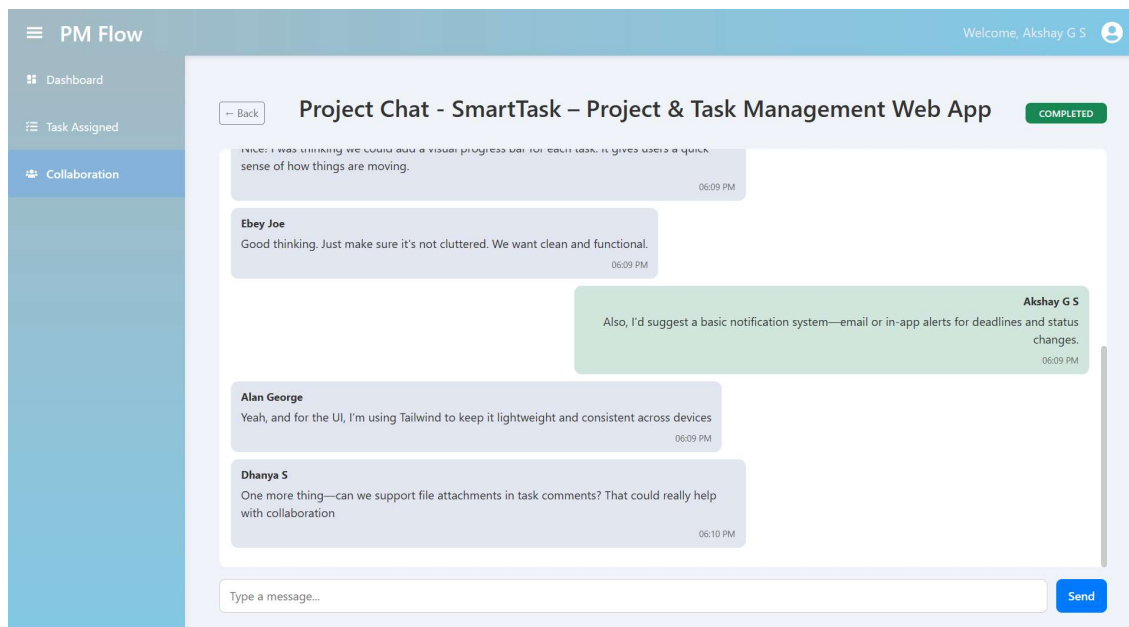
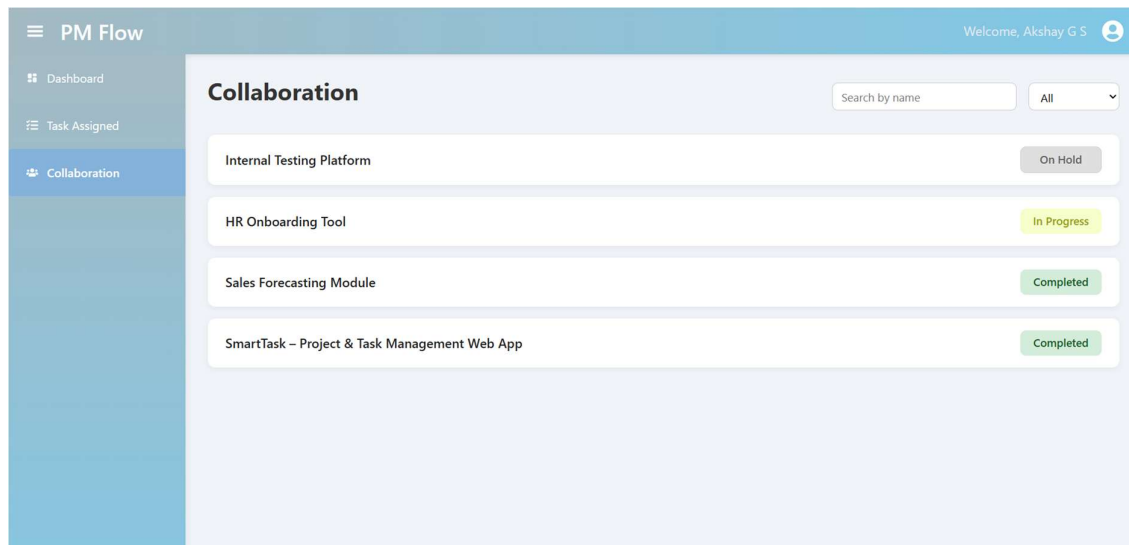


The Task Details View provides an in-depth, task-specific interface where team members can interact with individual assignments, update progress, and communicate directly with their project manager.

Key Components

- **Key task attributes** - Upon opening a specific task, the member can view a detailed Information about the tasks.
- **Status Update Functionality** – status of the task can be updated.
- **One-on-One Chat with Manager**
 - Private message with manager
 - Includes sender name and timestamp
 - Only the assigned team member and project manager can participate in the chat.

COLLABORATION



The Reusable Collaboration Page used between Manager & Members enables seamless team communication by providing a centralized space for group discussions per project. Users can view a list of all projects they are associated with and initiate real-time group chats with other team members and the project manager. This feature fosters collaboration, quick updates, and enhanced project coordination without needing external communication tools.

Key Components

- Displays a list of all projects associated with the logged-in user.
- Each project in the list includes the project name and on clicking a project, users are redirected to a dedicated chat page for that specific project.
- The Chat Page shows a real-time group chat interface. Messages are displayed with sender information.
- Polling-based updates fetch new messages every 5 seconds to simulate real-time communication.

BACKEND

1. Functional Requirements

i. User Management & Authentication

- JWT-based login and registration
- Role-based access control for ADMIN, PROJECT_MANAGER, and MEMBER
- Encrypted password storage using BCrypt

ii. Project Management

- Admins can create, assign, update, and delete projects
- Project Managers can view and manage their own projects
- Supports filtering, summary view, and project statistics

iii. Task Management

- Project Managers assign tasks to team members
- Members can view and update task status
- Task features include title, description, due date, and status tracking

iv. Chat Module

- Private chats: Between project manager and team member (within the same project)
- Group chats: Among all members and manager of a project
- Access control ensures only project members can view chats
- Admin has unrestricted access

v. Logging and JUnit Testing

- Logging is implemented using SLF4J in both controller and service layers
- Every key action, success, and exception is logged
- JUnit testing is implemented for the controller layer using `@WebMvcTest` and `MockMvc`
- Both positive and negative test scenarios are covered
- API endpoints were manually tested using Postman

DATABASE DESIGN

- USERS

Column	Type	Constraints
id	BIGINT	PK, Auto Increment
username	VARCHAR	UNIQUE
email	VARCHAR	UNIQUE
password_hash	VARCHAR	NOT NULL
first_name	VARCHAR	
last_name	VARCHAR	
created_at	DATETIME	
updated_at	DATETIME	
role	ENUM (ADMIN, PROJECT_MANAGER, MEMBER)	DEFAULT: MEMBER

- PROJECTS

Column	Type	Constraints
id	BIGINT	PK, Auto Increment
name	VARCHAR	NOT NULL
description	TEXT	
status	ENUM (NOT_STARTED, IN_PROGRESS, ...)	DEFAULT: NOT_STARTED
start_date	DATE	
end_date	DATE	
manager_id	BIGINT	FK → users(id), NOT NULL
created_at	DATETIME	
updated_at	DATETIME	

- PROJECT_TEAM_MEMBERS (JOIN TABLE)

Column	Type	Constraints
project_id	BIGINT	FK → projects(id), PK
user_id	BIGINT	FK → users(id), PK

- TASKS

Column	Type	Constraints
id	BIGINT	PK, Auto Increment
name	VARCHAR	NOT NULL
description	TEXT	
due_date	DATETIME	
priority	ENUM (HIGH, MEDIUM, LOW)	
status	ENUM (NOT_STARTED, IN_PROGRESS, ...)	
project_id	BIGINT	FK → projects(id), NOT NULL
assignee_id	BIGINT	FK → users(id), Nullable
created_at	DATETIME	
updated_at	DATETIME	

- CHAT_MESSAGES

Column	Type	Constraints
id	BIGINT	PK, Auto Increment
sender_id	BIGINT	FK → users(id), NOT NULL
receiver_id	BIGINT	FK → users(id), Nullable
project_id	BIGINT	FK → projects(id), NOT NULL
task_id	BIGINT	FK → tasks(id), ON DELETE CASCADE
content	TEXT	NOT NULL
is_group	BOOLEAN	
timestamp	DATETIME	

ENTITY RELATIONSHIPS

- User ↔ Project
 - One-to-Many: User (as manager) manages multiple projects
 - Many-to-Many: project_team_members join table for team membership

- **Project ↔ Task**
 - One-to-Many: Each project has multiple tasks
- **User ↔ Task**
 - One-to-Many: User can be an assignee to multiple tasks
- **Task ↔ ChatMessage**
 - One-to-Many: Each task can have multiple chat_messages
- **User ↔ ChatMessage**
 - Sender: Many chat_messages can be sent by a user
 - Receiver: (Optional) Many messages can be received by a user

PROJECT MANAGEMENT

Roles Supported

ROLE	ACCESS LEVEL
Admin	Full access to all projects
Project Manager	Access only to their assigned projects

CORE FEATURES

1. Admin Features

- Create Project - Create a new project with manager and team members
- View All Projects - View a list of all projects
- View Project by ID or Name - Fetch details of any project
- Update Project by ID/Name - Update name, description, status, manager, team members, and end date
- Delete Project - Delete a project using its ID or name
- Filter Projects - Filter all projects by name, manager name, status, or end date
- Count Projects by Status - Count all projects for a given status

2. Project Manager Features

- View My Projects - View all projects managed by the current manager
- Get Project by Name (Owned) - Fetch summary or detailed info of their project
- Update Status and End Date Only - Update only status and end date of owned project
- Get Team Members of Project - View team member list for a project they manage

- Filter My Projects - Filter only their projects
- Count My Projects by Status - Count number of projects they manage by status

ACCESS CONTROL

Access is enforced via Spring Security using `hasRole("ADMIN")` or `hasRole("PRODUCT_MANAGER")` in the Security Config.

LOGGING

SLF4J logging is implemented in:

- Controller Layer - Logs every endpoint call
- Service Layer - Logs all business operations like creation, updates, filtering, etc.

JUNIT TESTS

JUnit tests implemented for Controller Layer (`ProjectControllerTest.java`) - Tests all endpoints using `MockMvc`, Covers success cases and error conditions

API ENDPOINTS

PROJECTS

HTTP Method	Endpoint	Role	Description
POST	<code>/api/projects/create</code>	ADMIN	Create a new project
GET	<code>/api/projects/all</code>	ADMIN	Get list of all projects
GET	<code>/api/projects/{projectId}</code>	ADMIN	Get project by ID
GET	<code>/api/projects/by_name?name={projectName}&detailed={true false}</code>	ADMIN	Get project by name (detailed or summary)
PUT	<code>/api/projects/{projectId}</code>	ADMIN	Update project fields (name, desc, status, manager, members, endDate)

PUT	/api/projects/by_name/{projectName}	ADMIN	Update project by name (same fields as above)
DELETE	/api/projects/{projectId}	ADMIN	Delete project by ID
DELETE	/api/projects/by_name/{projectName}	ADMIN	Delete project by name
GET	/api/projects/filter?projectName={...}&managerName={...}&status={...}&endDate={...}	ADMIN	Filter projects (any one or more fields optional)
GET	/api/projects/count?status={IN_PROGRESS NOT_STARTED COMPLETED}	ADMIN	Count projects by status
GET	/api/projects/manager/{managerId}	PRODUCT_MANAGER	Get list of projects for a specific manager
GET	/api/projects/manager/{managerId}/by_name?name={...}&detailed={true false}	PRODUCT_MANAGER	Get project by name for that manager
PUT	/api/projects/manager/{managerId}/update_status_enddate/{projectId}?status={...}&endDate={...}	PRODUCT_MANAGER	Update only status and/or end date
GET	/api/projects/manager/{managerId}/team_members/{projectId}	PRODUCT_MANAGER	Get team members (IDs and usernames) for the project
GET	/api/projects/manager/{managerId}/filter?projectName={...}&status={...}&endDate={...}	PRODUCT_MANAGER	Filter only their projects
GET	/api/projects/manager/{managerId}/count?status={...}	PRODUCT_MANAGER	Count their projects by status

TASKS

HTTP Method	Endpoint	Access Description	Description
POST	/api/tasks	Authenticated Users	Create a new task
GET	/api/tasks/project/{projectId}	Authenticated Users	Get all tasks under a project
GET	/api/tasks/user/{userId}	Authenticated Users	Get all tasks assigned to a user
GET	/api/tasks/{taskId}	Authenticated Users	Get the details of a particular task
PUT	/api/tasks/{taskId}/status?status={status}	Assignee, Project Manager, Admin	Update task status via query param
PUT	/api/tasks/{taskId}/assign?userId={userId}	Project Manager, Admin	Assign a task to a user
PUT	/api/tasks/{taskId}	Assignee, Project Manager, Admin	Update task name, description, due date, priority, assignee
PUT	/api/tasks/admin/{id}	Admin Only	Admin updates task name and status
DELETE	/api/tasks/{taskId}	Project Manager, Admin	Delete a task by ID

USERS

HTTP Method	Endpoint	Access Description	Description
POST	/api/auth/register	Public	Register a new user (default role: MEMBER)
POST	/api/auth/login	Public	Authenticate the user and return a JWT token
POST	/api/auth/logout	Authenticated Users	Logout user (JWT invalidation via header)
GET	/api/users/me	Authenticated Users	Get profile of currently logged-in user
PUT	/api/users/me	Authenticated Users	Update basic profile (role update not allowed)
GET	/api/users	Admin Only	Get a list of all users

PUT	/api/users/admin/{id}	Admin Only	Admin updates user info & role
-----	-----------------------	------------	--------------------------------

MODULE 4: CHAT SERVICE MODULE

ROLES SUPPORTED

ROLE	ACCESS LEVEL
Admin	Full access to all chat messages (private and group)
Project Manager	Access only to messages in projects they manage
Member	Access only to messages in projects they are assigned to

CORE FEATURES

1. Send Private Message

- Endpoint: POST /api/chat/private/sender/{senderId}/receiver/{receiverId}/project/{projectId}/task/{taskId}
- Sender and receiver must belong to the same project and task.
- Only visible to sender, receiver, and Admin.

2. Send Group Message

- Endpoint: POST /api/chat/group/sender/{senderId}/project/{projectId}
- Message is visible to all members of the given project (including the manager).
- Only project participants or Admin can send/view.

3. Get Private Messages

- Endpoint: GET /api/chat/private/sender/{senderId}/receiver/{receiverId}/project/{projectId}/task/{taskId}
- Returns full conversation between two users for a specific project and task.
- Only visible to the two users or Admin.

4. Get Group Messages

- Endpoint: GET /api/chat/group/project/{projectId}
- Returns all group messages sent in the context of the project.
- Only accessible by project participants or Admin.

5. Get Projects Assigned to Authenticated User

- Endpoint: GET /api/chat/assigned-projects
- Returns list of project names and statuses for projects the logged-in user (manager or team member) is assigned to.

ACCESS CONTROL

- Access is enforced using authenticated() in Spring Security configuration.
- Role-based restrictions are handled explicitly in the Service Layer, not through hasRole().
- Admins can view all chats; members/managers can view only their own chats related to their assigned projects.

LOGGING

SLF4J logging is implemented in:

- Controller Layer: Logs every API endpoint call
- Service Layer: Logs all business logic – sending, fetching messages, validations, and access checks

JUNIT TESTS

JUnit tests implemented for:

Controller Layer only (ChatControllerTest.java)

- Tests all endpoints using MockMvc
- Covers success and failure scenarios (including unauthorized access)

API ENDPOINTS

HTTP Method	Endpoint	Access Description	Description
POST	/api/chat/private/sender/{senderId}/receiver/{receiverId}/project/{projectId}/task/{taskId}	Sender, Receiver	Send a private message for a specific project and task
POST	/api/chat/group/sender/{senderId}/project/{projectId}	Authenticated Project Manager and Members	Send a group message for a specific project
GET	/api/chat/private/sender/{senderId}/receiver/{receiverId}/project/{projectId}/task/{taskId}	Sender, Receiver, Admin	Get full private chat history between two users
GET	/api/chat/group/project/{projectId}	Project manager, Members, Admin	Get group chat messages for a project
GET	/api/chat/assigned_projects	Authenticated	Get assigned project names and statuses for current user

NOTE: All endpoints require a valid JWT token (i.e.), Authorization: Bearer <token>

FUTURE SCOPE

- WebSocket upgrade for real-time delivery using STOMP
- Notification integration (email/alert on new message)
- Dark Mode Support - Dark/light theme toggle for better accessibility and user preference.
- Drag-and-Drop Task Board - Implement a drag-and-drop interface for task status changes.

References

- **React Icons** - <https://react-icons.github.io/react-icons>
- **React Toastify** - <https://fkhadra.github.io/react-toastify>
- **Axios** - <https://axios-http.com>
- **Bootstrap** - <https://getbootstrap.com>
- **GitHub** - <https://github.com>
- **React (Official Docs)** - <https://reactjs.org>
- **Spring Boot Documentation** - <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>
- **Spring Security + JWT Authentication** - <https://www.baeldung.com/spring-security-oauth-jwt>
- **MySQL Official Documentation** - <https://dev.mysql.com/doc/>
- **JPA & Hibernate Documentation** - https://docs.jboss.org/hibernate/orm/current/userguide/html_single/Hibernate_User_Guide.html
- **Postman** - <https://www.postman.com/>
- **Mockito** - <https://site.mockito.org/>
- **JUnit 5** - <https://junit.org/junit5/docs/current/user-guide/>