# Full Stack Project Report Format

## Title Page

- Project Title : Task Flow – A Team Based Task Management and Productivity Tracker
- Institution / Organization: IBM Summer Internship Program
- Mentor/Supervisor: Mr. Jaivik Panchal
- Submission Date: 21 July 2025

# Certificate

This is to certify that the project title **“Task Flow – A Team Based Task Management and Productivity Tracker”** is an original work completed by the **Vishakha Shrimali** as part of the Full Stack Devlopment Internship under the guidance of **Mr.Jaivik Panchal.**

## Acknowledgment

I would like to express my sincere gratitude to my mentor, peers, and the IBM Internship Program for their support and guidance throughout the development of the **"Task Flow – A Team Based Task Management and Productivity Tracker"** project.

# Table of Contents

## Abstract

TaskFlow is a full-stack web application designed to streamline project and task management for teams. It provides an intuitive Kanban-style interface to organize work into boards, which helps teams visualize tasks, limit work-in-progress, and maximize workflow efficiency. The system implements secure user authentication (using JWT), email verification, and real-time collaboration features. Users can create and manage projects (boards) and tasks, assign them to team members, and track progress. Key features include user registration/login, project and task CRUD operations, team invitations, notifications, and a responsive drag-and-drop Kanban board interface. Overall, TaskFlow's objective is to streamline project workflows and improve productivity by providing a centralized platform where team members can collaborate on tasks and always see the latest status.

## Objective

TaskFlow's primary goal is to streamline project and task management for collaborative teams. Its objectives include:

- **Centralized Task Management:** Enable users to create and organize tasks into projects or boards in one place.
- **Member Assignment:** Allow team leads to assign tasks to specific members or groups, clarifying responsibility and ownership.
- **Progress Tracking:** Provide dashboards and Kanban boards so users can track task status (e.g. To Do, In Progress, Done) and project milestones in real time.
- **Collaboration and Communication:** Support team collaboration through user profiles and notifications, ensuring members receive timely updates about task changes.
- **Role-based Control:** Implement different user roles (Admin, Manager, Member) so that permissions are tailored to responsibilities.

These objectives ensure TaskFlow helps teams plan their work, stay on schedule, and maintain visibility into ongoing tasks.

## System Architecture

- **Diagram**: Frontend(HTML/CSS/JS) ↔ Backend(Node.js/Express.js) ↔ Database(MongoDB)

The Task Flow project follows a **client-server architecture** using the **three-tier model**:

- **Frontend**: Built with HTML, CSS, and JavaScript.
- **Backend**: Developed in Node.js using the Express framework.
- **Database**: MongoDB with Mongoose ORM for schema modeling.

•The system follows a **client-server architecture** where the **frontend communicates** with the backend using **HTTP requests**.

•The **backend validates** these requests, applies **authorization using middleware**, accesses MongoDB for data, and **sends JSON responses** to the frontend.

• **Authentication** is implemented using **JWT tokens**. Users receive a token upon login which is verified on every secure route.

• **User roles** like admin, manager and member are validated in protected backend routes to control access.

• The **MongoDB** database stores user data, project assignments, task details, and system-generated notifications.

## Technology Stack

- **Frontend:** HTML, CSS, JavaScript
- **Backend:** Node.js, Express.js
- **Database:** MongoDB (with Mongoose)
- **Tools:** GitHub, VS Code, Postman, MongoDB Compass

## Modules/Features

| **Module** | **Description** | Technologies Used | Key Features |
|---|---|---|---|
| Authentication | Manages user login, registration, and role-based access using JWT. | Node.js, Express, MongoDB, JWT, bcrypt | - Secure login<br>- Role-based access<br>- Password encryption |
| Dashboard | Displays system summary stats like total users, tasks, and project progress. | HTML, CSS, JS | - Project progress bar<br>- Completion rate<br>- User/project/task summary |
| Task Management (Kanban) | Allows users to drag and drop tasks across statuses (todo, in-progress, review, done). | HTML, CSS, JS, MongoDB | - Drag-and-drop<br>- Task CRUD<br>- Priority & status handling |
| Project Management | Enables admin/manager to create projects, assign users, and track progress. | Node.js, MongoDB, Express, JS | - Add/delete project<br>- Assign members<br>- Calculate and show progress |
| Team Management | Admins and managers can add members and assign them to projects. | JS, Express, MongoDB | - Add new users<br>- Assign to projects<br>- Auto-inject into project dashboard |
| Notification System | Shows real-time activity updates like task creation and status changes. | Node.js, Express, MongoDB | - Bell icon with unread dot<br>- Notification dropdown<br>- Mark as read |
| Profile Management | Users can view and edit their profile including name, email, and avatar. | JS, Express, MongoDB | - Avatar preview<br>- Update personal details<br>- Display completion rating |

## Frontend Development

- The frontend is developed using **HTML**, **CSS**, and **Vanilla JavaScript** .

- It delivers a clean, responsive UI with support for modern browser features.

- **Key UI Components & Pages:**

    - **Login & Signup Pages** – For authentication
    - **Dashboard Page** – Displays:
        - Project cards with progress bars and assigned users
        - Kanban-style Task Board (with drag-and-drop support)
        - Recent tasks, Project progress, and team information
    - **Modals** – For:
        - Adding/Delete Tasks
        - Adding /Delete Projects
        - Assigning Team Members
        - Profile Update
    - **Profile Panel** – Allows users to update their name, avatar, and email

- **UI/UX Strategy**:

    - Uses **Font Awesome** icons
    - UI is clean and developer-friendly
    - Fully responsive layout using **Flexbox** and **Grid**

## Backend Development

- Built using **Node.js** and **Express.js**

- Implements a clean **RESTful API** structure

- **Core API Routes**:

  - `/api/auth` – User registration and login (with JWT)
  - `/api/profile` – Get/update user profile and stats
  - `/api/dashboard` – Load all data (users, tasks, projects) in one call
  - `/api/tasks` – Create, read, update, delete tasks
  - `/api/projects` – Manage projects and assign members
  - `/api/notification` – Generate and fetch notifications

- **Authentication & Access Control**:

  - JWT token is generated on login and verified via middleware (`auth.js`)
  - Middleware ensures only authenticated users can access protected routes
  - **Role-based access:** (admin, manager, member) restricts certain actions:
    - Only Admin/Manager can create/delete projects or assign team members

## Database Design

- The project uses **MongoDB** as the database.
- Interaction is done via **Mongoose**, an ODM (Object Data Modeling) library for MongoDB in Node.js.

- **MongoDB** is used for storing:
    - **Users**: name, email, password (hashed), role (admin/manager/member), avatar
    - **Projects**: name, description, budget, deadline, assigned team members
    - **Tasks**: title, description, priority, status, project reference, assignedTo, createdBy
    - **Notifications**: message, read status, user reference, link
- **Data relationships** are maintained using MongoDB references (ObjectId).

## Database Schema:

- **User Collection:**
  `name`, `email`, `password (hashed)`, `role`, `avatar`
- **Project Collection:**
  `name`, `description`, `budget`, `deadline`, `assignedTo[]`, `createdAt`
- **Task Collection:**
  `title`, `description`, `priority`, `status`, `project`, `assignedTo`, `createdBy`, `createdAt`
- **Notification Collection:**
  `message`, `user`, `read`, `link`, `createdAt`

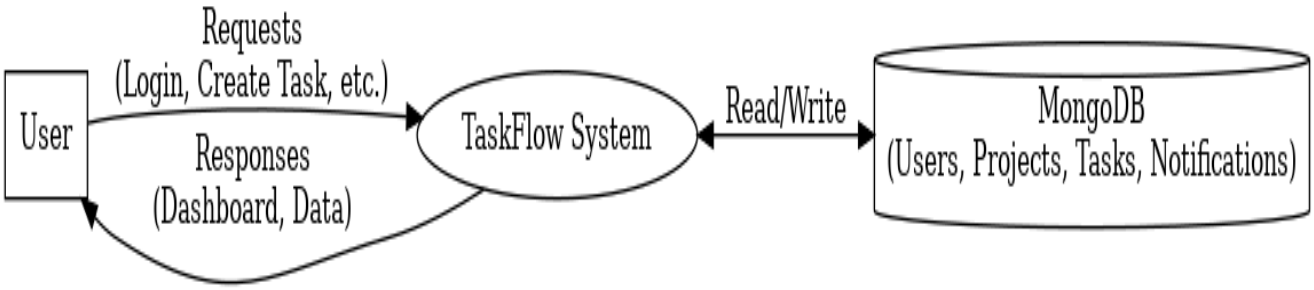
## Data Flow Diagrams

**DFD Level 0 – Context Diagram**

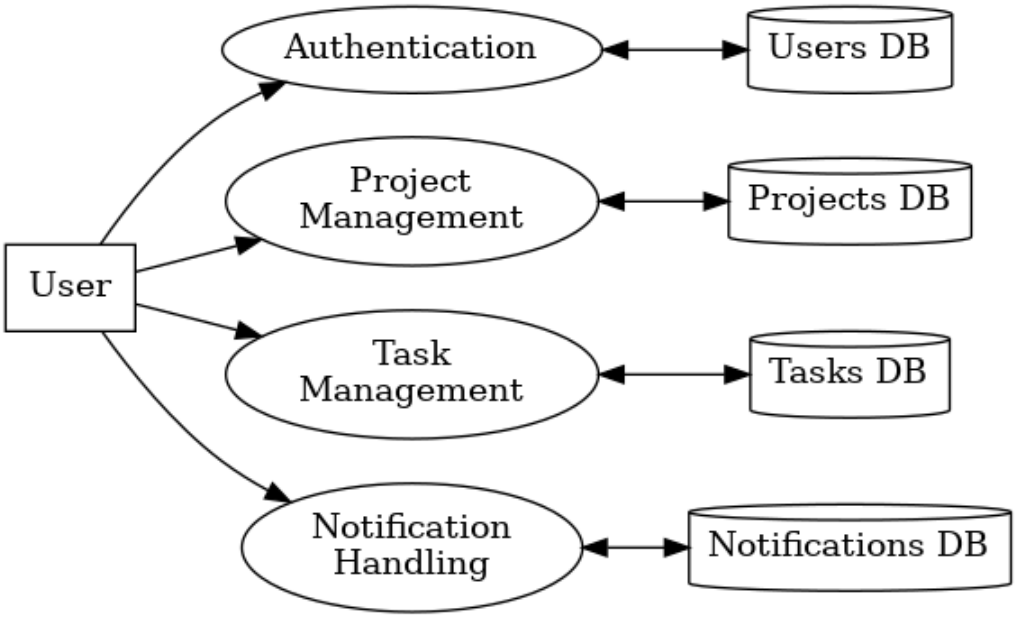
- **User** (Admin/Manager/Member): Performs login, manages tasks/projects, views dashboard.

- **TaskFlow System**: Handles all frontend, backend, authentication, and data management logic.

- **MongoDB Database**: Stores user, project, task, and notification data.




**DFD Level 1 – Decomposition Diagram**



## Testing (optional)

• **API Testing**: All REST APIs tested using Postman (login, project, task, notification).

• **Manual Testing**: Verified project/task creation, assignment, drag-drop, and notifications.

• **Unit Testing**: Basic logic tested manually in the frontend and backend using console/debugging.

## Security Measures

- **Password Hashing**: All passwords are securely encrypted using bcrypt.
- **JWT Authentication**: User sessions are validated using JWT stored in session storage.
- **Role-Based Access**: Admin and Manager routes are protected via middleware.
- **Input Validation**: Fields are validated both on frontend and backend to prevent invalid data.

## Limitations

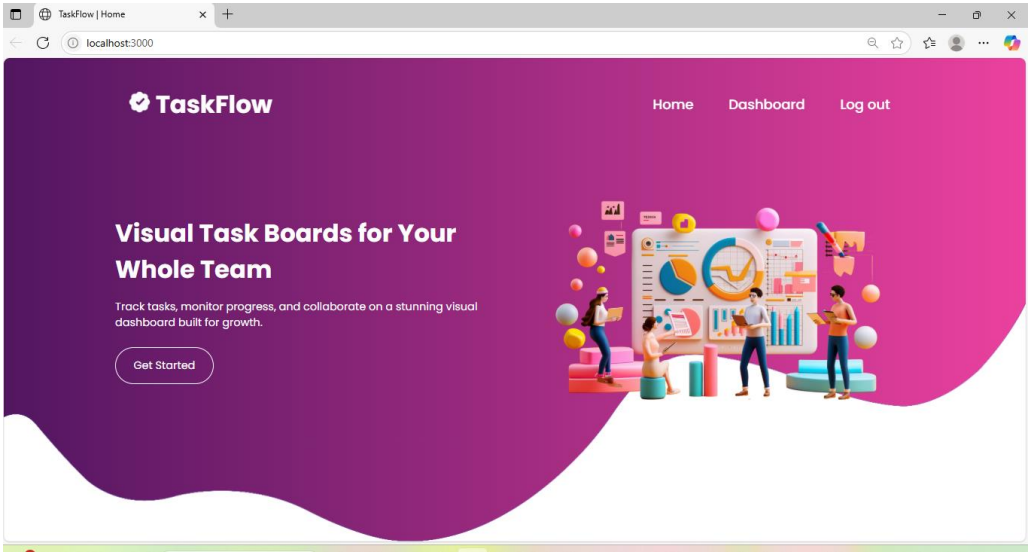
- No password reset functionality.
- Real-time updates are not implemented (requires refresh).
- No file upload support.
- Limited mobile responsiveness.
- Role permissions are basic (admin, manager, member).

## Future Enhancements

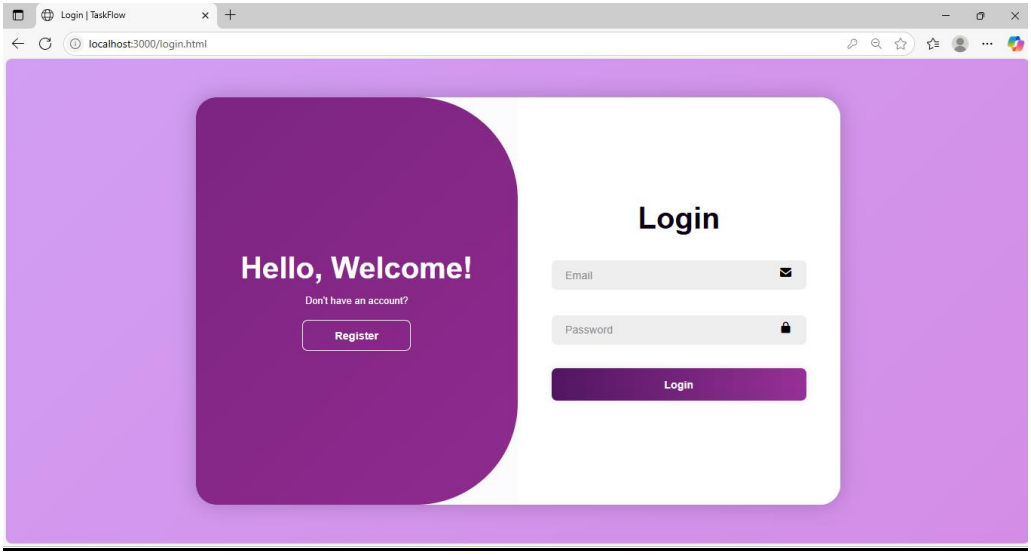
- Add real-time updates using WebSockets.
- Implement file uploads for tasks and projects.
- Add password reset feature with email OTP.
- Improve mobile UI and responsiveness.
- Introduce detailed role permissions and audit logs.

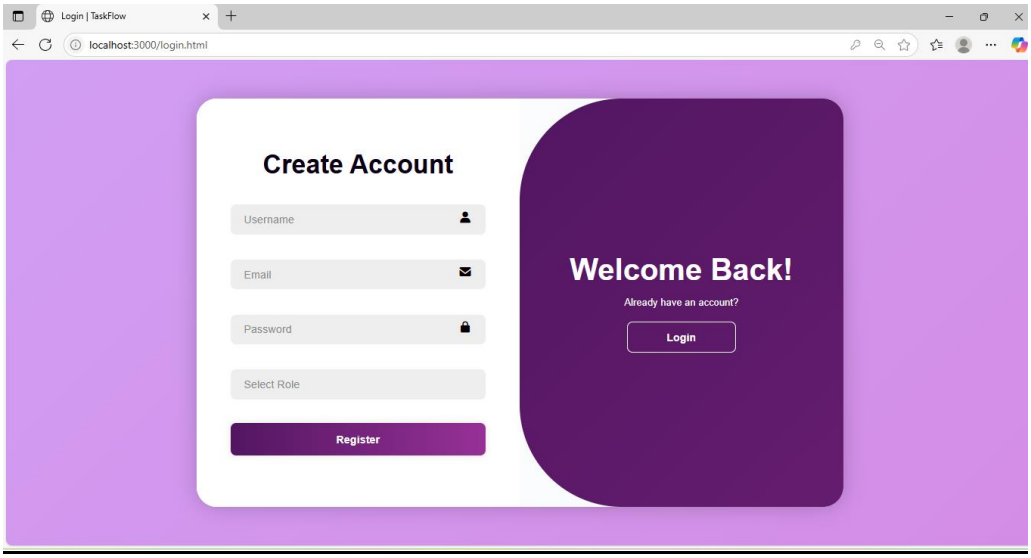
# Screenshots

- **Home page**

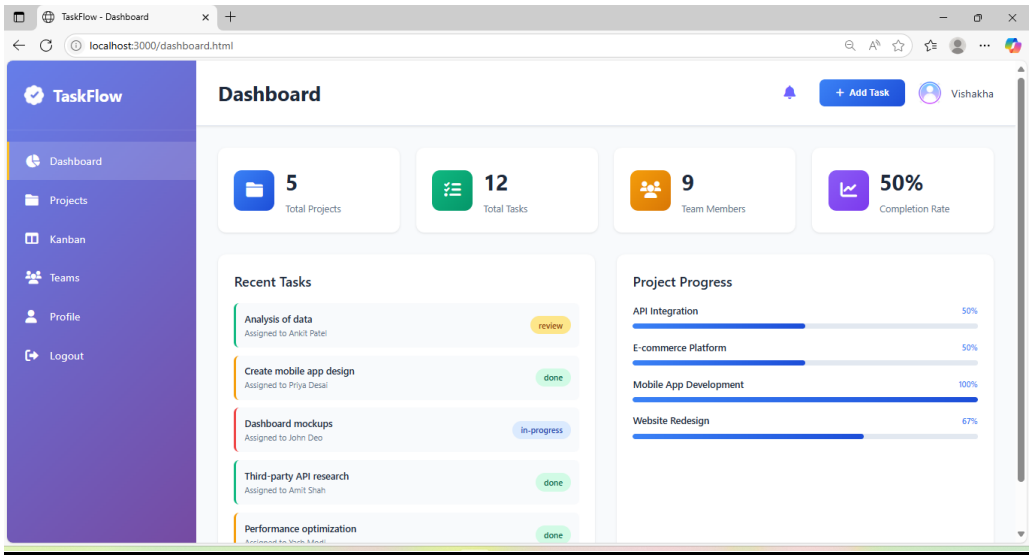


## Login page and Register page

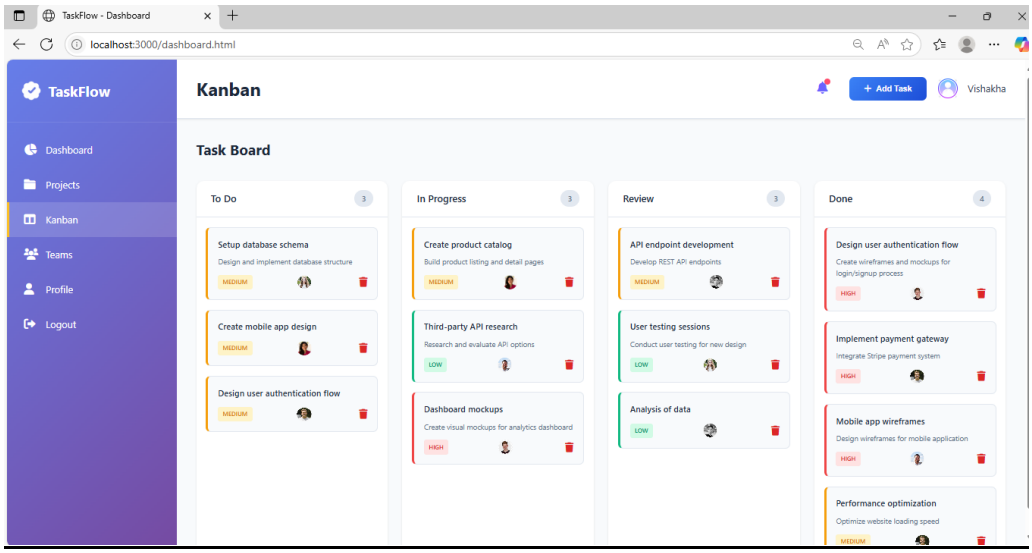


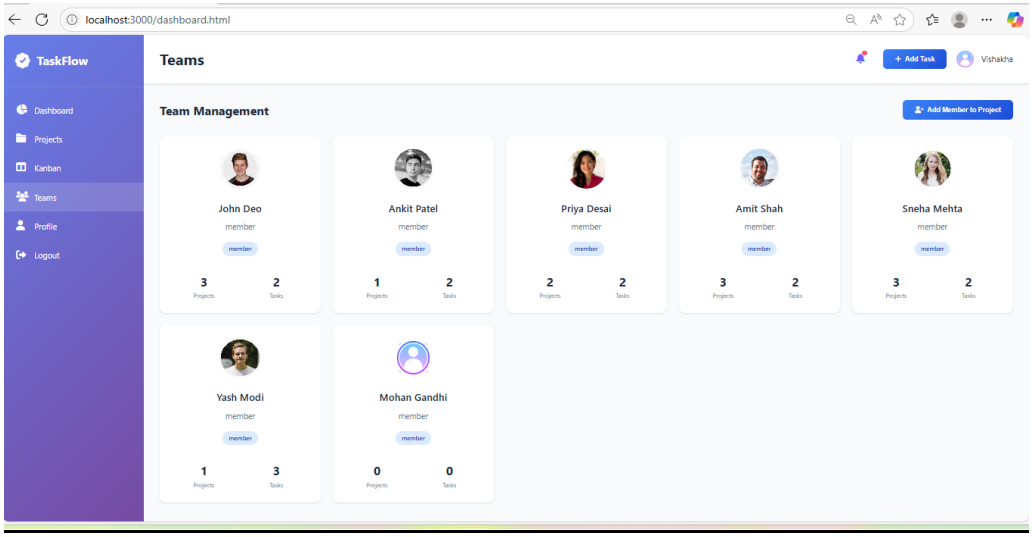



# Dashboard page

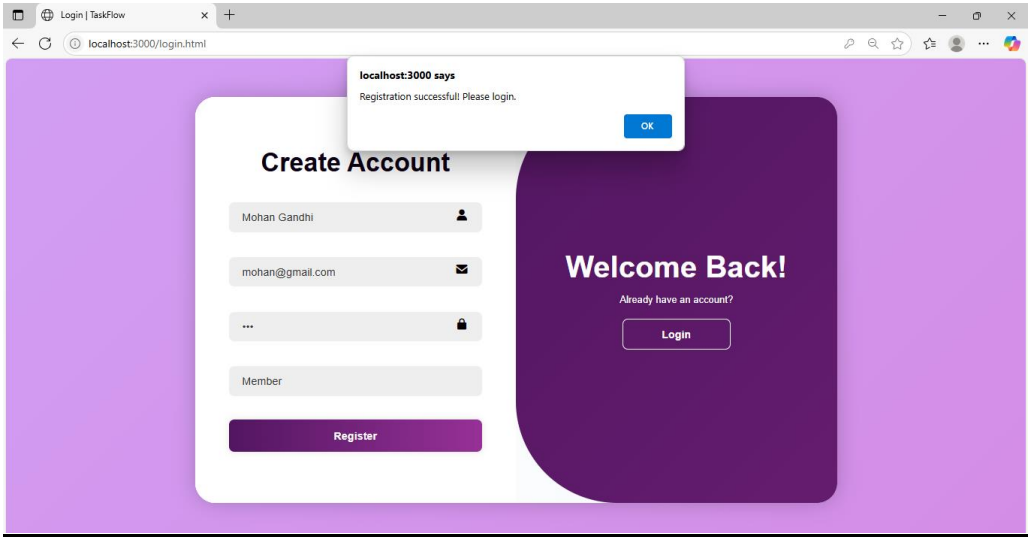


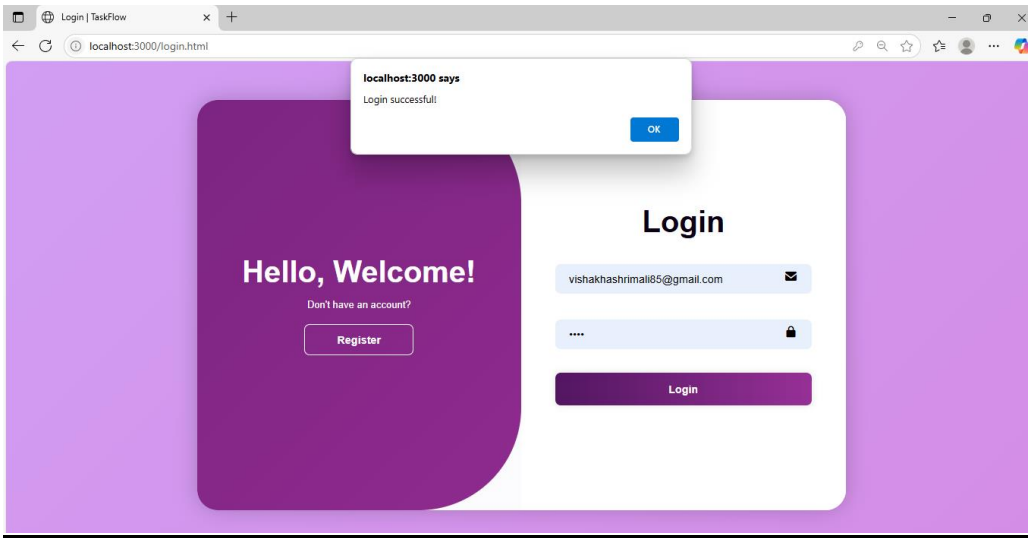
# Admin view


TaskFlow - Dashboard
localhost:3000/dashboard.html
TaskFlow
Kanban
+ Add Task
Vishakha
Dashboard
Projects
Kanban
Teams
Profile
Logout
Task Board
To Do
3
Setup database schema
Design and implement database structure
MEDIUM
Create mobile app design
MEDIUM
Design user authentication flow
MEDIUM
In Progress
3
Create product catalog
Build product listing and detail pages
MEDIUM
Third-party API research
Research and evaluate API options
LOW
Dashboard mockups
Create visual mockups for analytics dashboard
HIGH
Review
3
API endpoint development
Develop REST API endpoints
MEDIUM
User testing sessions
Conduct user testing for new design
LOW
Analysis of data
LOW
Done
4
Design user authentication flow
Create wireframes and mockups for login/signup process
HIGH
Implement payment gateway
Integrate Stripe payment system
HIGH
Mobile app wireframes
Design wireframes for mobile application
HIGH
Performance optimization
Optimize website loading speed
MEDIUM

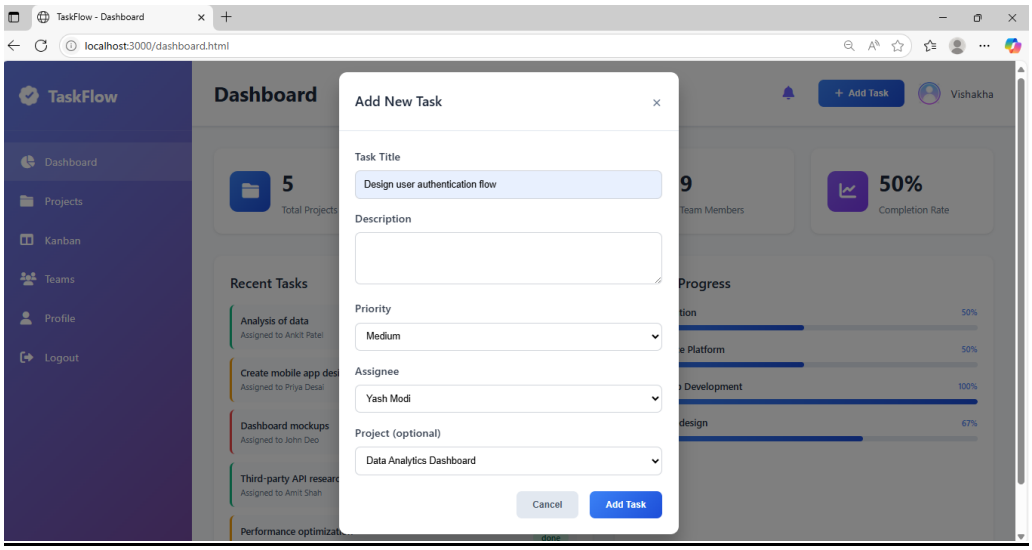

localhost:3000/dashboard.html
TaskFlow
Teams
+ Add Task
Vishakha
Dashboard
Projects
Kanban
Teams
Profile
Logout
Team Management
Add Member to Project
John Deo
member
member
3
Projects
2
Tasks
Ankit Patel
member
member
1
Projects
2
Tasks
Priya Desai
member
member
2
Projects
2
Tasks
Amit Shah
member
member
3
Projects
2
Tasks
Sneha Mehta
member
member
3
Projects
2
Tasks
Yash Modi
member
member
1
Projects
3
Tasks
Mohan Gandhi
member
member
0
Projects
0
Tasks

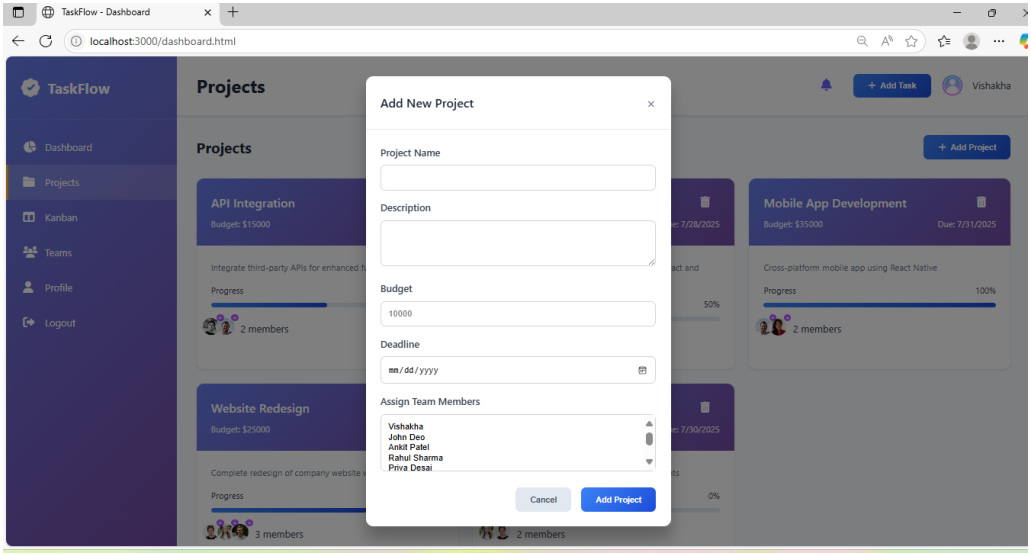
## CRUD oprations:

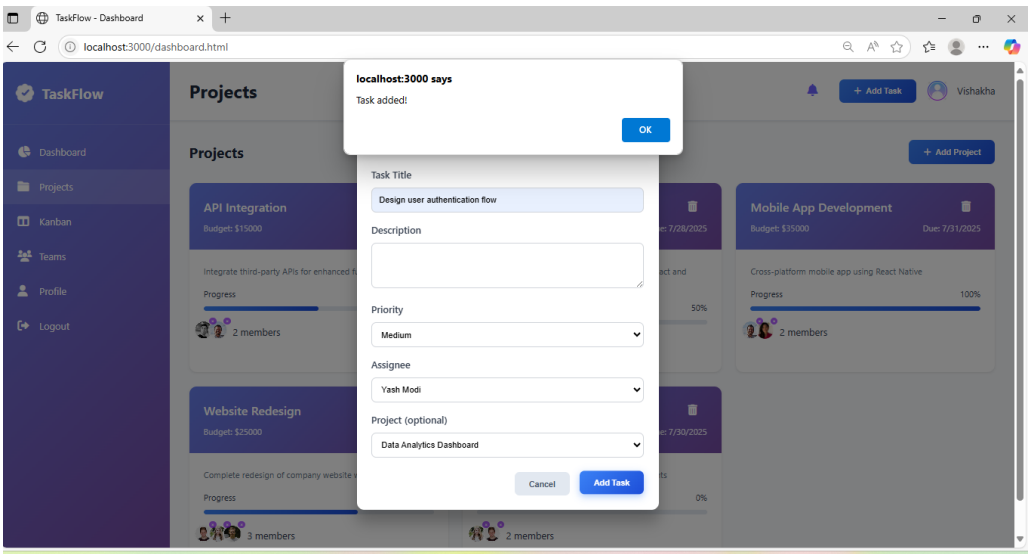


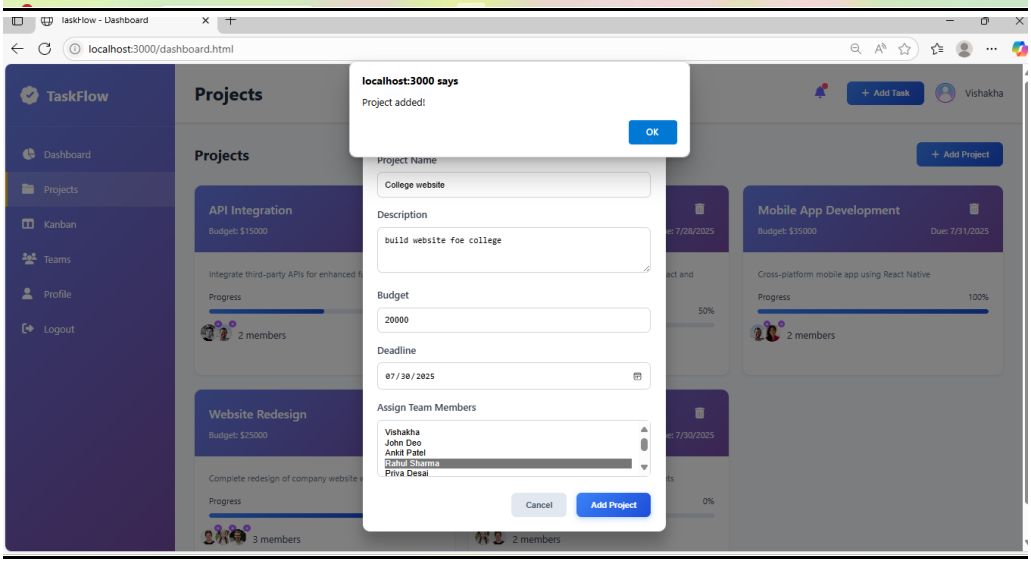



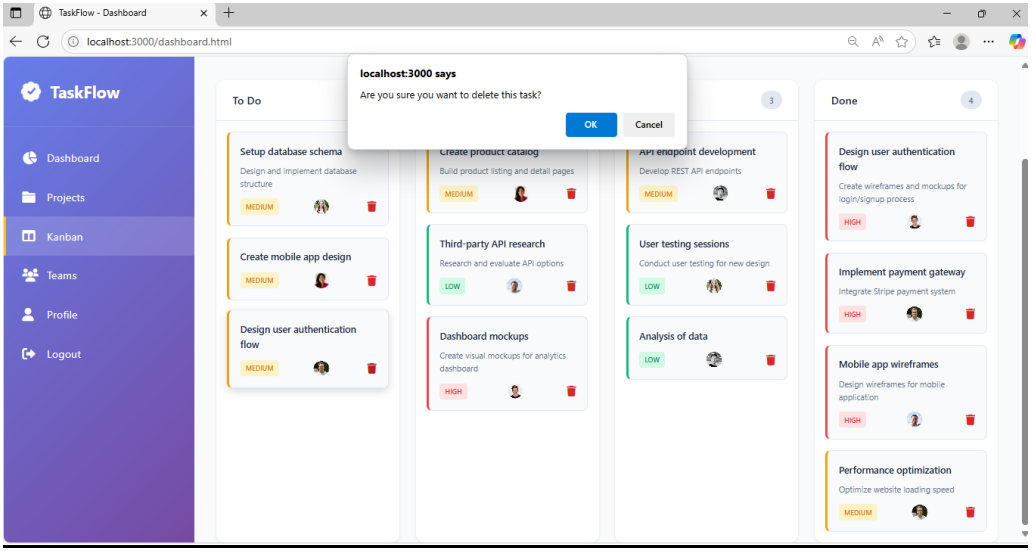
# Admin  and Manager assign and create project and task

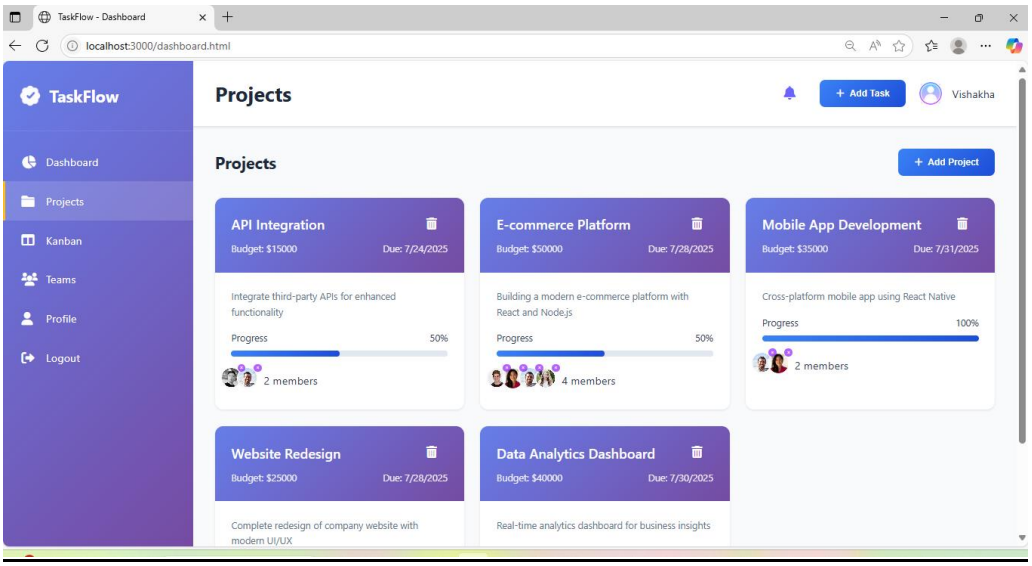



TaskFlow - Dashboard
localhost:3000/dashboard.html
TaskFlow
Projects
Add New Project
Project Name
Description
Budget
10000
Deadline
mm/dd/yyyy
Assign Team Members
Vishakha
John Deo
Ankit Patel
Rahul Sharma
Priva Desai
Cancel
Add Project
+ Add Task
Vishakha
+ Add Project
API Integration
Budget: $15000
Mobile App Development
Budget: $35000
Due: 7/31/2025
Website Redesign
Budget: $25000
2 members
3 members

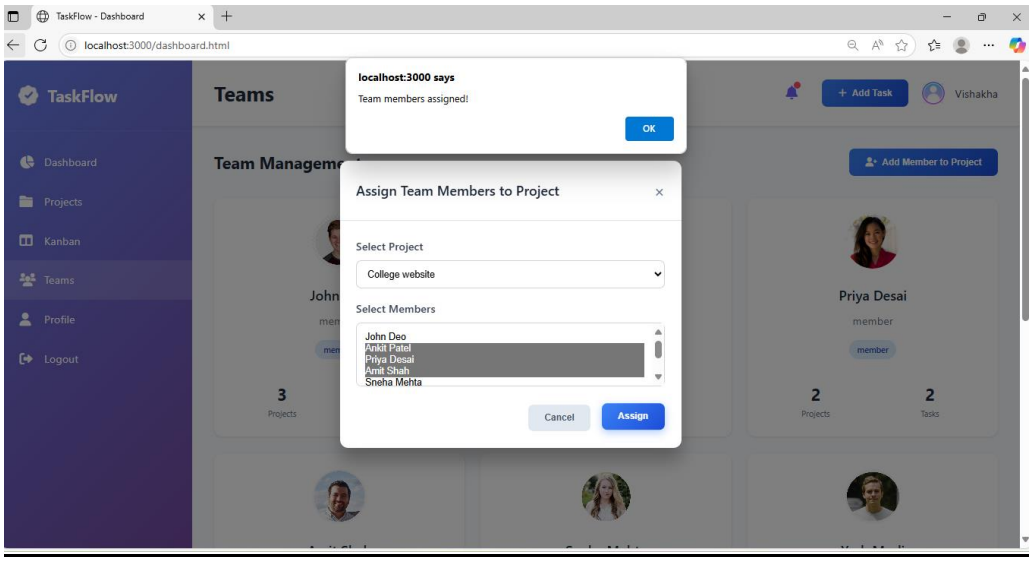

TaskFlow - Dashboard
localhost:3000/dashboard.html
localhost:3000 says
Task added!
OK
TaskFlow
Projects
Task Title
Design user authentication flow
Description
Priority
Medium
Assignee
Yash Modi
Project (optional)
Data Analytics Dashboard
Cancel
Add Task
+ Add Task
Vishakha
+ Add Project
API Integration
Budget: $15000
Mobile App Development
Budget: $35000
Due: 7/31/2025
Website Redesign
Budget: $25000
2 members
3 members


localhost:3000/dashboard.html
localhost:3000 says
Project added!
OK
TaskFlow
Projects
Project Name
College website
Description
build website foe college
Budget
20000
Deadline
07/30/2025
Assign Team Members
Vishakha
John Deo
Ankit Patel
Rahul Sharma
Priva Desai
Cancel
Add Project
+ Add Task
Vishakha
+ Add Project
API Integration
Budget: $15000
Mobile App Development
Budget: $35000
Due: 7/31/2025
Website Redesign
Budget: $25000
2 members
3 members

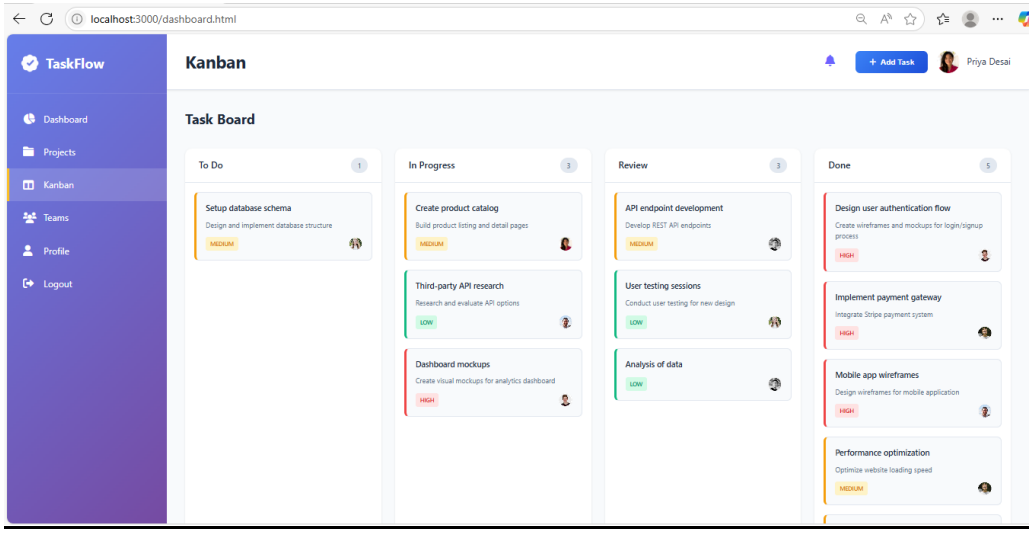

TaskFlow - Dashboard
localhost:3000/dashboard.html
localhost:3000 says
Are you sure you want to delete this task?
OK
Cancel
TaskFlow
Dashboard
Projects
Kanban
Teams
Profile
Logout
To Do
Done
Setup database schema
Design and implement database structure
MEDIUM
Create mobile app design
MEDIUM
Design user authentication flow
MEDIUM
Third-party API research
Research and evaluate API options
LOW
Dashboard mockups
Create visual mockups for analytics dashboard
HIGH
User testing sessions
Conduct user testing for new design
LOW
Analysis of data
LOW
Design user authentication flow
Create wireframes and mockups for login/signup process
HIGH
Implement payment gateway
Integrate Stripe payment system
HIGH
Mobile app wireframes
Design wireframes for mobile application
HIGH
Performance optimization
Optimize website loading speed
MEDIUM

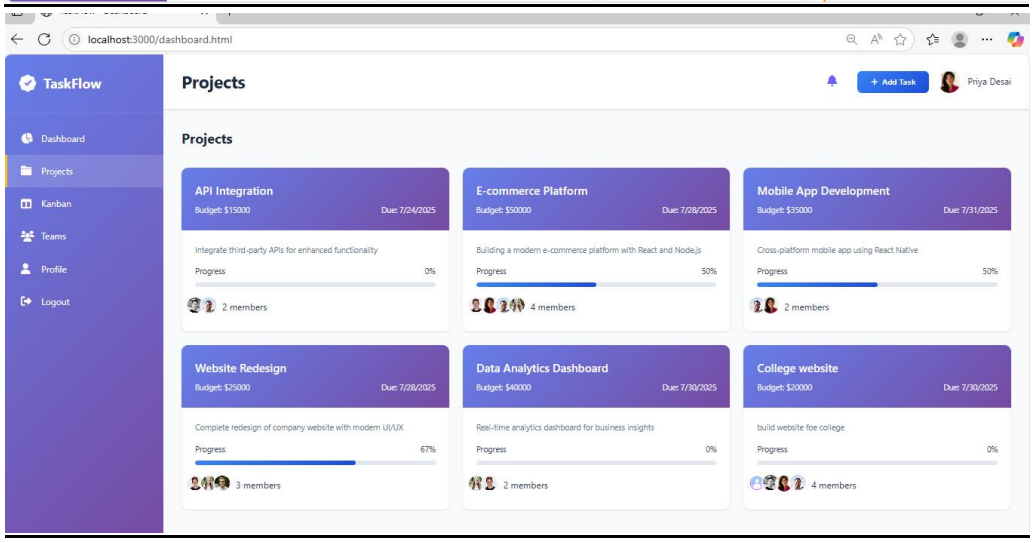

TaskFlow - Dashboard
localhost:3000/dashboard.html
TaskFlow
Dashboard
Projects
Kanban
Teams
Profile
Logout
Projects
+ Add Task
Vishakha
Projects
+ Add Project
API Integration
Budget: $15000
Due: 7/24/2025
Integrate third-party APIs for enhanced functionality
Progress
50%
2 members
E-commerce Platform
Budget: $50000
Due: 7/28/2025
Building a modern e-commerce platform with React and Node.js
Progress
50%
4 members
Mobile App Development
Budget: $35000
Due: 7/31/2025
Cross-platform mobile app using React Native
Progress
100%
2 members
Website Redesign
Budget: $25000
Due: 7/28/2025
Complete redesign of company website with modern UI/UX
Data Analytics Dashboard
Budget: $40000
Due: 7/30/2025
Real-time analytics dashboard for business insights

# Admin and manager can add or assign member in project

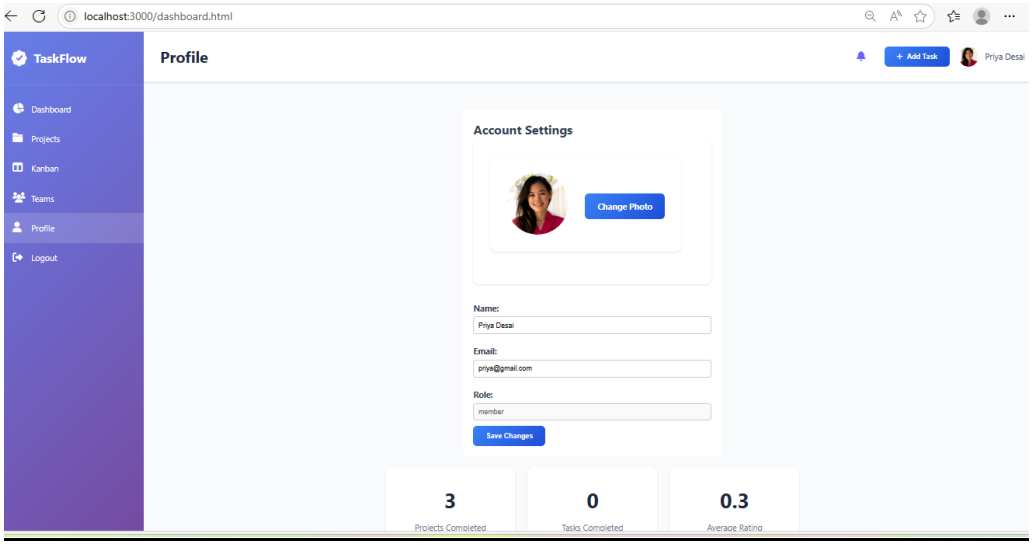


# Team member view

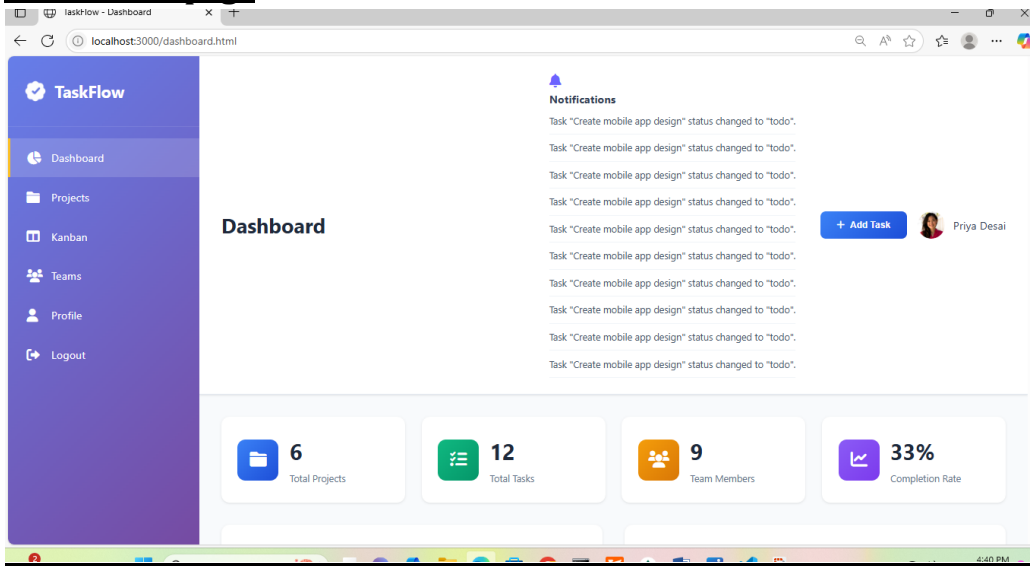


## Profile page




## Notification page



## Annexures

- GitHub repo link https://github.com/Vishakha1625/Taskflow-project

## References

https://expressjs.com/
https://mongoosejs.com/

https://jwt.io/introduction

- **Frontend Libraries**: Font Awesome, Google Fonts

- **Tools Used**: Postman, VS Code, GitHub