Software Requirements Specification (SRS)

Task Management Application (MERN Stack)

Overview

This document provides a summary of requirements for developing a simplified task management application using the MERN stack (MongoDB, Express.js, React.js, Node.js). The application organizes tasks into lists, allowing users to track and manage their work efficiently. This application is designed to be completed within a 45-day timeframe.

Core Features

1. User Authentication

- User registration with email and password
- User login with JWT-based authentication
- Logout functionality

2. List Management

- Create new lists with titles (e.g., “To Do”, “In Progress”, “Done”)
- Edit list titles
- Delete lists

3. Task Management

- Create tasks with title and optional description
- Move tasks between lists
- Edit task details
- Delete tasks

Data Models

User Model

{ _id: ObjectId, email: String (required, unique), password: String (hashed, required), name: String (required), createdAt: Date, updatedAt: Date }

List Model

{ _id: ObjectId, title: String (required), position: Number, createdAt: Date, updatedAt: Date }

Task Model

{ _id: ObjectId, title: String (required), description: String, list: ObjectId (ref: List), position: Number, createdBy: ObjectId (ref: User), createdAt: Date, updatedAt: Date }

Key API Endpoints

Authentication API

- POST /api/auth/register
- POST /api/auth/login
- POST /api/auth/logout

List API

- GET /api/lists
- POST /api/lists
- PUT /api/lists/:id
- DELETE /api/lists/:id

Task API

- GET /api/lists/:id/tasks
- POST /api/lists/:id/tasks
- PUT /api/tasks/:id
- DELETE /api/tasks/:id
- PUT /api/tasks/:id/move

Technology Stack

Frontend

- React.js for UI components
- React Router for navigation
- Axios for API requests
- React Beautiful DND for drag-and-drop functionality

Backend

- Node.js runtime environment
- Express.js for API framework
- MongoDB for database
- Mongoose for object modeling
- JSON Web Tokens for authentication
- Bcrypt for password hashing

Development Flow

1. Git Setup (Day 1)

- Initialize Git repository
- Create .gitignore file
- Setup branch structure (main, development)
- Initial commit

2. MongoDB Setup (Day 2-3)

- Install MongoDB locally or setup MongoDB Atlas
- Configure database connection
- Setup database testing environment
- Create database access credentials

3. Schema Design (Day 4-6)

- Define User schema
- Define List schema
- Define Task schema
- Implement schema validation
- Create relationships between schemas

4. API Codebase Setup (Day 7-20)

- Initialize Node.js project
- Install dependencies
- Setup Express server
- Implement authentication middleware
- Create user routes and controllers
- Create list routes and controllers
- Create task routes and controllers
- Implement API testing

5. UI Codebase Setup (Day 21-45)

- Initialize React application
- Setup folder structure
- Implement authentication views (login/register)
- Create dashboard component
- Implement list components with CRUD operations
- Implement task components with CRUD operations
- Add drag-and-drop functionality for tasks between lists
- Connect UI to API endpoints
- Implement responsive design
- Testing and bug fixes

Minimal Viable Product Deliverables

- Functional authentication system
- Ability to create, read, update, and delete lists
- Ability to create, read, update, and delete tasks
- Drag-and-drop functionality for tasks between lists
- Responsive UI
- Basic documentation for setup and usage

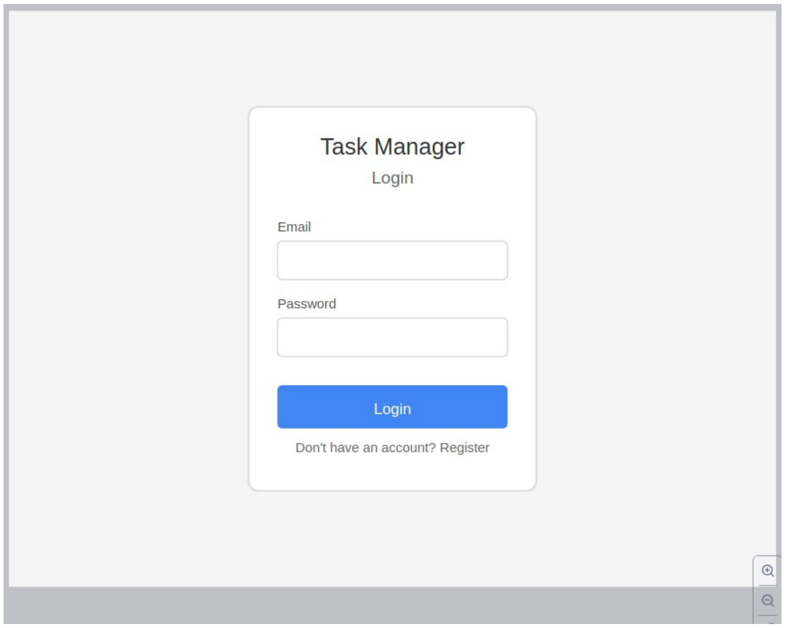
Further improvements(Optional)

If time permits, you can optionally work on the following items:

1. Add users and allocate tasks to them.
2. Add and list task comments.
3. Create projects and organize tasks under each project.

Few wire-frames for reference:

Login:



Task Manager

Login

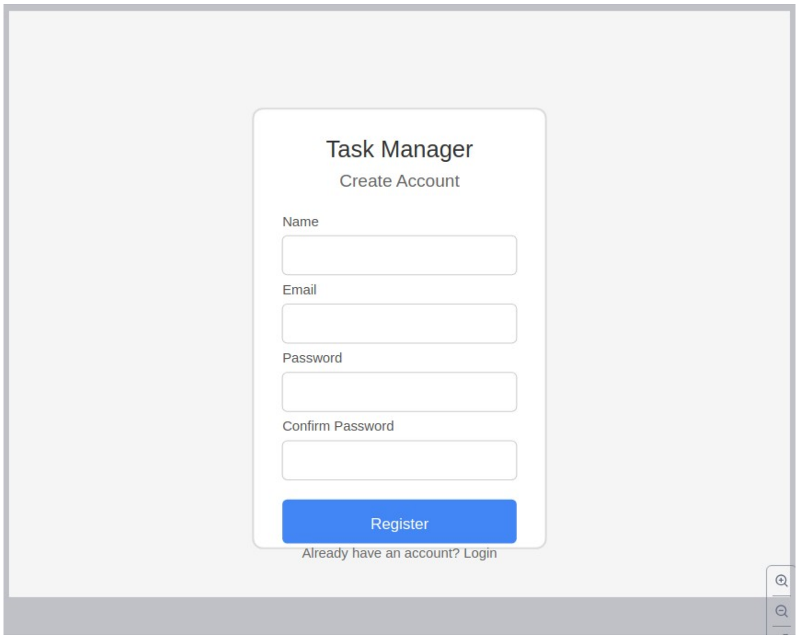
Email

Password

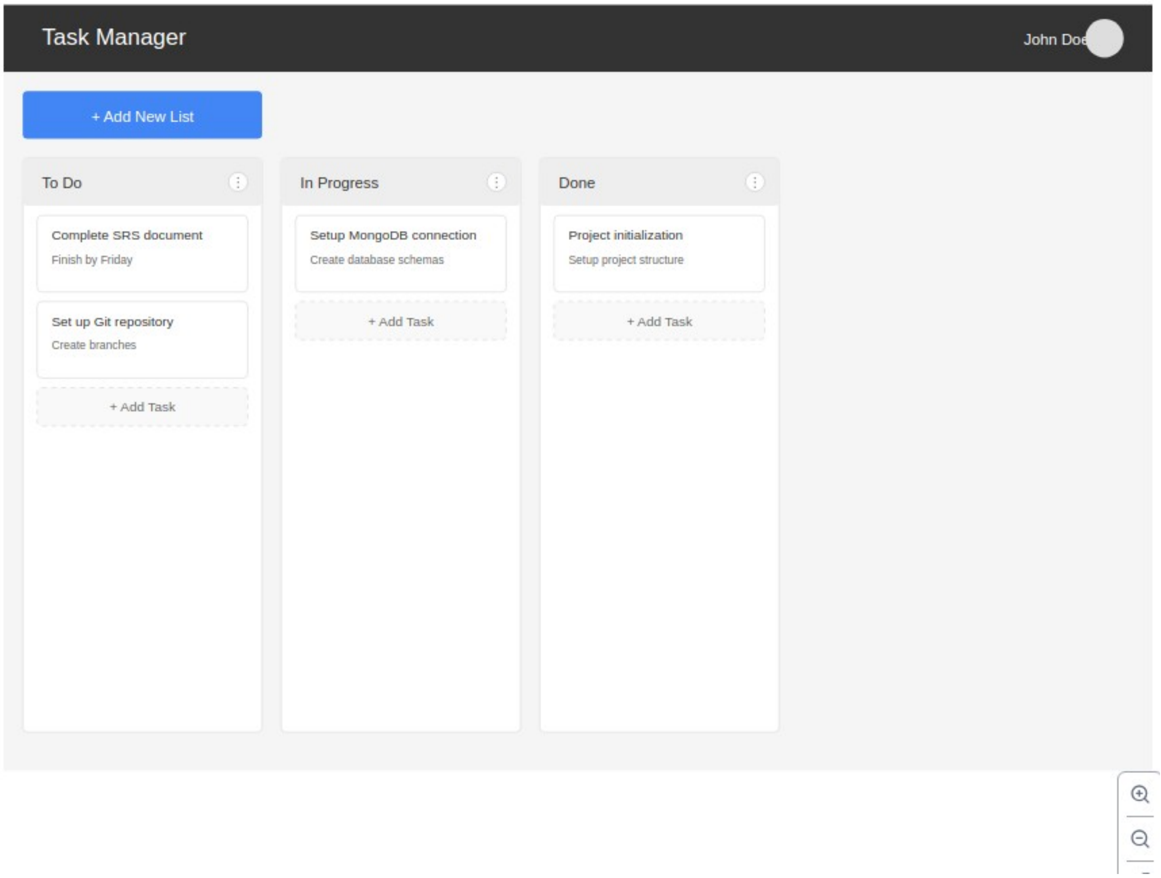
Login

Don't have an account? Register

Registration:



Dashboard

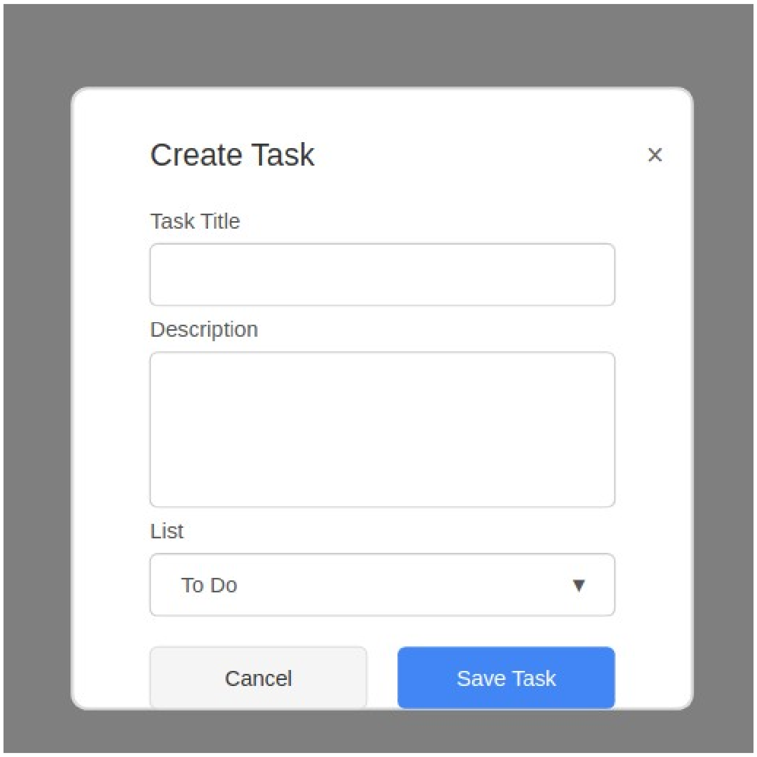


To Do: Complete SRS document, Finish by Friday; Set up Git repository, Create branches

In Progress: Setup MongoDB connection, Create database schemas

Done: Project initialization, Setup project structure

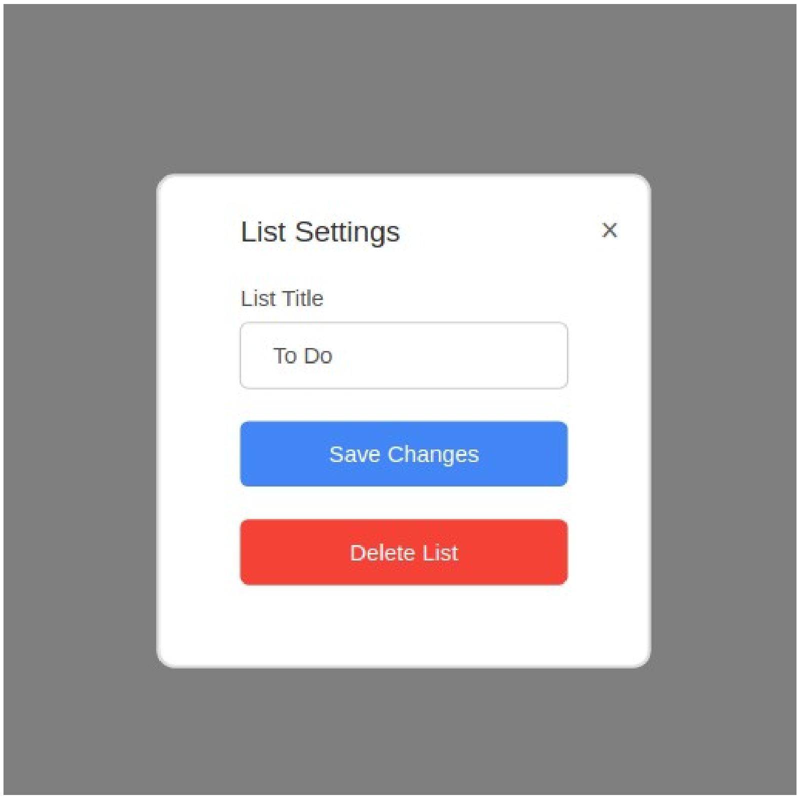
Create/edit task:



List

To Do ▼

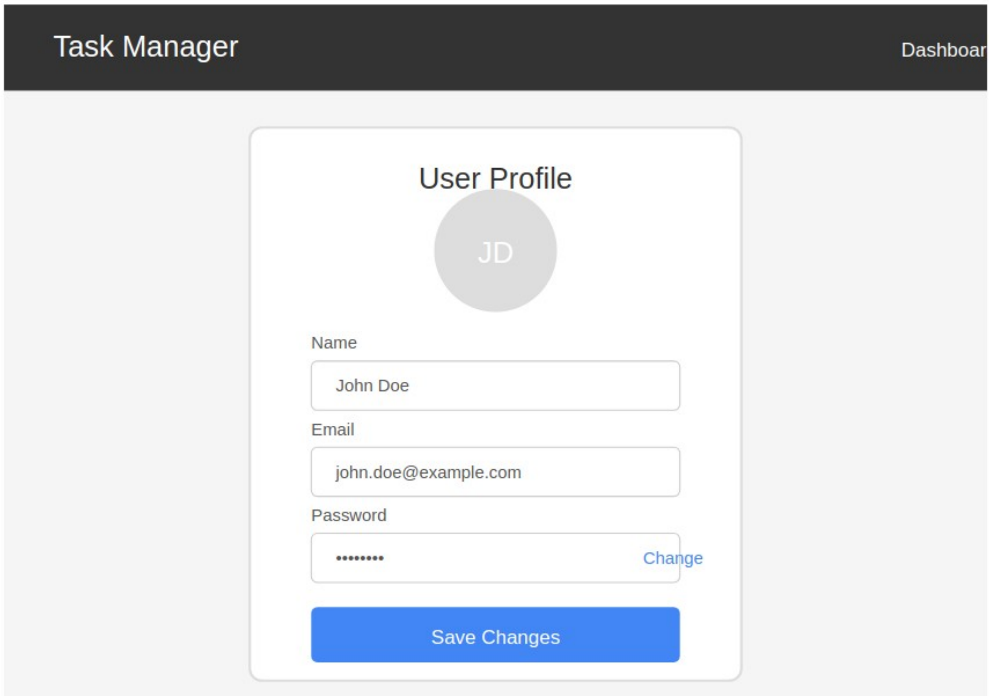
Add/ edit list:



List Title

To Do

Edit user profile:



User Profile

Name: John Doe

Email: john.doe@example.com

Password: ••••••••