Project Report: MERN Stack To-Do App

Table of Contents

1. PROJECT OVERVIEW

The **MERN Stack To-Do App** is a **full-stack web application** designed to help users manage their daily tasks efficiently.

This project is built using the **MERN stack** (MongoDB, Express.js, React.js, and Node.js) and follows modern web development practices. The backend handles user authentication, task management, and API requests, while the frontend provides a seamless and interactive user experience. The app also supports CRUD operations, allowing users to create, view, update, and delete tasks effortlessly.

❖ KEY OBJECTIVES

- User-Friendly Task Management: Provide an intuitive UI where users can efficiently add, edit, and organize their tasks.
- Categorization for Better Productivity: Introduce 7 cognitive-based task categories to help users optimize their workflow.
- Seamless Authentication: Ensure secure user login and signup using JWT-based authentication.
- Real-Time Interactivity: Implement React.js features like state management (Redux), React Hooks, and API integration for a smooth user experience.
- Scalable Backend: Use Node.js & Express.js for a scalable REST API with MongoDB as the database for storing users and tasks.
- Responsive Design: Ensure accessibility across different devices with a mobile-friendly UI.
- Deployment & Hosting: Deploy the frontend on Vercel and backend on Render or Heroku for easy accessibility.

2. TECH STACK

- Frontend: React.js with React Router DOM for navigation and Redux for state management.
- Backend: Node.js with Express.js for API handling.
- Database: MongoDB using Mongoose as an Object Data Modeling (ODM) library.
- HTTP Client: Axios for making API requests.
- Authentication: User authentication via JSON Web Tokens (JWT).
- Styling: Basic CSS and React Toastify for notifications.
- Deployment: GitHub and Vercel for hosting the front-end and backend.

3. FEATURES & FUNCTIONALITIES

❖ USER AUTHENTICATION

- Users can sign up and log in using a secure authentication system.
- Authentication is managed using JWT.

❖ TO-DO LIST MANAGEMENT (CRUD OPERATIONS)

- Create a new to-do item.
- Read and display all tasks.
- Update an existing to-do item.
- Delete tasks when completed or unnecessary.

❖ UNIQUE FEATURE: TASK CATEGORIZATION

- Unlike conventional to-do apps, this application introduces a **7-category task division system** inspired by the book *Mind Management, Not Time Management*.
- Tasks are categorized based on mental energy, focus, and strategic impact rather than urgency or deadlines.
- This approach helps users optimize their productivity by working on tasks that align with their mental state and priorities.
- The seven categories from the book are:

    1. Effort – Tasks requiring intense focus and energy.
    2. Creative – Tasks that need innovative thinking.
    3. Productive – Routine and systematic work.
    4. Administrative – Organizational and planning tasks.
    5. Supportive – Helping others or responding to queries.
    6. Learning – Acquiring new knowledge or skills.
    7. Restorative – Activities that recharge energy and creativity.

❖ FRONTEND UI

- Built using **React.js**.
- Uses React Router DOM for navigation.
- Implements useState and useEffect hooks.
- User-friendly interface with Toast notifications

❖ BACKEND API

- Developed using **Node.js** and **Express.js**.
- API routes for user authentication and CRUD operations.
- Handles authentication and request validation.

❖ DATABASE (MONGODB) DESIGN

- Mongoose is used to define schemas and interact with the database.

1. **USERS COLLECTION (USERS)**

```
{
  "_id": "ObjectId",
  "name": "John Doe",
  "email": "johndoe@example.com",
  "password": "hashed_password",
  "createdAt": "timestamp",
  "updatedAt": "timestamp"
}
```

2. **TASKS COLLECTION (TASKS)**

```
{
  "_id": "ObjectId",
  "userId": "ObjectId",
  "title": "Complete project report",
  "description": "Write the final project report and review before submission",
  "category": "Productive",
  "status": "Pending",
  "priority": "High",
  "dueDate": "timestamp",
  "createdAt": "timestamp",
  "updatedAt": "timestamp"
}
```

3. **SESSION TOKENS COLLECTION (SESSIONS)**

```
{
  "_id": "ObjectId",
  "userId": "ObjectId",
  "token": "JWT_token_string",
  "expiresAt": "timestamp"
}
```

4. **RELATIONSHIPS & INDEXING:**

- One-to-Many Relationship (User -> Tasks).
- Indexed email field in users for fast lookup.
- Indexed userId field in tasks for efficient queries.
- Indexed category and status fields in tasks for filtering.

❖ STATE MANAGEMENT (REDUX)

- Centralized state management using Redux.
- Ensures seamless data flow across the application.

❖ CONNECTING FRONTEND & BACKEND

- CORS middleware is used to enable cross-origin requests.
- Axios is used to fetch data from the backend API.

❖ DEPLOYMENT

- Hosted on GitHub and deployed via **Vercel**.
- Backend can be deployed on cloud services like **Render or Heroku**.

4. PROJECT STRUCTURE

❖ FRONTEND:

- React components for UI.
- Redux store for managing state.
- React Router DOM for navigation.

❖ BACKEND:

- Express.js for API routes.
- Mongoose models for database operations.
- JWT-based authentication.

❖ DATABASE:

- MongoDB collections for users and tasks.

5. CONCLUSION

The MERN To-Do App is a full-stack web application that demonstrates the implementation of user authentication, CRUD operations, and API integration. The project is structured to provide hands-on experience in modern web development and is designed to be scalable and maintainable. The unique 7-category task system differentiates this app by offering a new way to manage tasks based on mental workload and cognitive efficiency, making it more effective for productivity optimization.