APSCHE SMART INTERNZ

Internship Title :- Backend development with node js and Mongo DB – APSCHE LTB3

Project Title :- NOTENEST – THE NOTE TAKING APP

Project Guide :- Sindhu Bhargavi Vurukuti

Project Team members :- P. Pavani

K.Sai devi sri

Ch.Mohan babu

B.Sai sagar

Project Description :-

The NoteNest App back-end is designed to offer users a reliable and secure environment for managing their personal notes. It features user authentication with JWT-based token security, ensuring that each user's data remains private and protected. The API enables users to perform CRUD (Create, Read, Update, Delete) operations on their notes, allowing them to organize information effectively.

The backend is developed using Node.js and Express.js, leveraging MongoDB as the database to store user information and notes efficiently. Mongoose ODM is utilized to interact with MongoDB, simplifying data manipulation. Middleware such as express-validator is used for request validation, while bcryptjs ensures password encryption for secure authentication.

One of the core functionalities of the Notes App back-end is its authentication mechanism. Users must sign up and log in to access their notes. Each API request is validated using JWT authentication, ensuring that only authorized users can manage their data.

The Notes App back-end also includes error handling mechanisms to provide informative responses in case of invalid requests or server issues. This ensures a smooth user experience by preventing unauthorized access and handling common issues gracefully.

Designed to be modular and scalable, this back-end can be extended to include additional features such as note sharing, categorization, and reminders. It serves as an excellent foundation for developers who want to build a fully functional note-taking application.

Project Outcome :-

With the Notes App, Amit has streamlined his study process. He no longer worries about losing important information, as all his notes are securely stored and easily accessible. The app's intuitive interface and useful features have

enhanced his productivity, helping him focus more on learning and less on managing scattered notes.

Key Features:

1. User Dashboard: A central dashboard where users can view, create, edit, and delete their notes in an organized manner.
2. Note Creation: Users can create and save notes with a title, description, and optional tags for easy categorization.
3. Edit & Update Notes: Existing notes can be updated with new content without creating a new entry.
4. Delete Notes: Users can remove notes that are no longer needed.
5. User Authentication: Secure login and signup system using JWT (JSON Web Token) authentication.
6. Secure Data Storage: All user notes are securely stored in a MongoDB database with unique user associations.
7. Tagging System: Notes can be categorized using tags for easier organization and filtering.
8. Search & Filter: Users can quickly find notes by searching based on title, content, or tags.
9. Date Tracking: Each note includes a timestamp to track when it was created or last updated.

Project Overview :-

NoteNest App Documentation

Overview

NoteNest is a secure note-taking application designed to help users store, manage, and retrieve their notes efficiently. The application ensures user authentication, allowing only authorized users to access their personal notes.

Features

- User authentication (Sign up, Login, Logout) using JWT
- Secure storage of notes
- CRUD operations (Create, Read, Update, Delete)
- Persistent storage using MongoDB
- User-friendly interface for managing notes
- Responsive design for mobile and desktop users

Tech Stack

- Frontend: React.js (optional, if implemented later)
- Backend: Node.js with Express.js
- Database: MongoDB (Mongoose ORM)
- Authentication: JWT (JSON Web Token)
- Deployment: (e.g., AWS, Heroku, Vercel)

API Endpoints

Authentication

- POST /api/auth/signup – Register a new user
- POST /api/auth/login – Authenticate and obtain token
- POST /api/auth/logout – Logout user

Notes Management

- POST /api/notes – Create a new note
- GET /api/notes – Fetch all notes for logged-in user
- GET /api/notes/:id – Fetch a specific note
- PUT /api/notes/:id – Update a note
- DELETE /api/notes/:id – Delete a note

Security Measures

- Encrypted passwords using bcrypt
- JWT-based authentication for secure access
- Input validation to prevent SQL/NoSQL injections
- CORS handling for API security

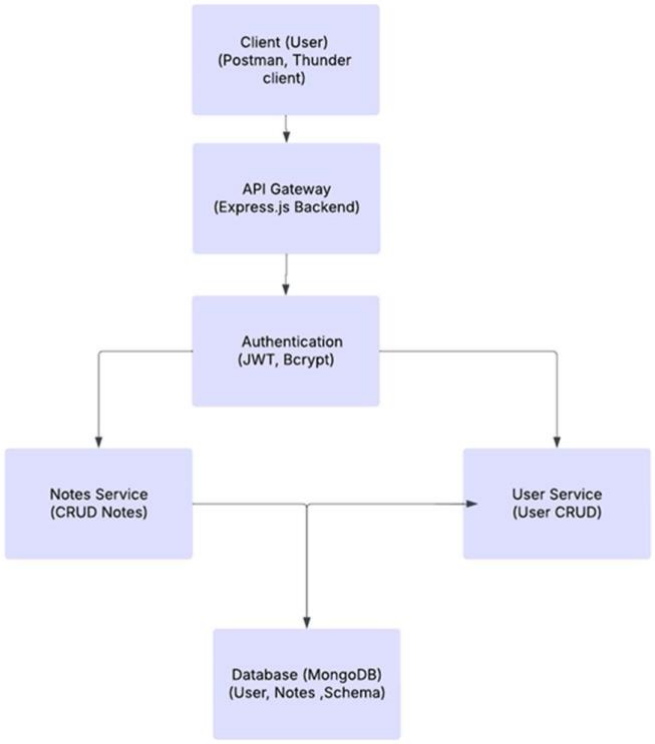
Future Enhancements

- Add cloud storage for media attachments
- Implement real-time collaboration
- Introduce categories/tags for notes
- Dark mode UI support

Conclusion

NoteNest provides a secure and efficient platform for managing notes. With authentication and CRUD functionalities, it serves as a useful tool for personal and professional note-taking.

Project Flow Diagram :-


Client (User)
(Postman, Thunder client)
API Gateway
(Express.js Backend)
Authentication
(JWT, Bcrypt)
Notes Service
(CRUD Notes)
User Service
(User CRUD)
Database (MongoDB)
(User, Notes ,Schema)

:-