**Project Documentation: Backend for Notes Application**

**Project Overview**

This project is a backend service for a Notes application, built using Node.js and Express.js. It provides user authentication, CRUD operations for managing notes, and ensures secure access using JWT-based authentication. The backend is connected to MongoDB for data storage and follows RESTful API principles.

**Tech Stack**

* **Backend:** Node.js, Express.js
* **Database:** MongoDB (using Mongoose ODM)
* **Authentication:** JWT (JSON Web Tokens)
* **Security:** bcrypt.js for password hashing, express-validator for input validation, dotenv for environment variable management

**Features**

1. **User Authentication:**
   * User registration with secure password hashing
   * User login with JWT authentication
   * Middleware to protect routes requiring authentication
2. **Notes Management:**
   * Create, Read, Update, and Delete (CRUD) operations for notes
   * Notes are linked to authenticated users
   * Secure access to user-specific notes
3. **Middleware Implementation:**
   * Authentication middleware to validate JWT tokens
   * Error handling middleware for standardized API responses

**Folder Structure**

/backend

│-- config/ # Database connection configuration

│-- middleware/ # Middleware files (authentication, error handling)

│-- models/ # Mongoose models (User, Note)

│-- routes/ # Express route handlers (auth, notes)

│-- .env # Environment variables

│-- index.js # Entry point of the backend server

│-- package.json # Node.js dependencies and scripts

**Installation and Setup**

**1. Clone the Repository**

git clone <repository\_url>

cd backend

**2. Install Dependencies**

npm install

**3. Configure Environment Variables**

Create a .env file in the root directory and add:

MONGO\_URI=<your\_mongodb\_connection\_string>

JWT\_SECRET=<your\_secret\_key>

PORT=5000

**4. Start the Server**

For production mode:

npm start

For development mode (with live reloading):

npm run dev

**API Endpoints**

**1. Authentication Routes (/api/auth)**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /register | Register a new user |
| POST | /login | User login |

**2. Notes Routes (/api/notes)**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | / | Fetch all user notes |
| POST | /create | Create a new note |
| PUT | /:id | Update a note by ID |
| DELETE | /:id | Delete a note by ID |

**Error Handling**

* Proper error messages are returned for invalid inputs, authentication failures, and unauthorized access.
* A centralized error handler ensures consistent API responses.

**Future Enhancements**

* Implement user roles (admin, user)
* Add file upload support for notes
* Implement rate limiting for API security

This document provides a clear understanding of the backend architecture, features, and usage. Let me know if you need any modifications! 🚀