Week 3 Status Report: MyPrivNote

Project Overview

The **MyPrivNote** project is a web application that allows users to create and retrieve private notes through a unique URL. It is intended to mimic the functionality of PrivNote, enabling users to share sensitive information securely. The current focus is on setting up the backend with basic CRUD operations for note creation and retrieval.

Current Status

1. Project Initialization

- The project was initialized using **Node.js**, and a Git repository was set up to track progress.
- Dependencies like **Express.js**, **Mongoose**, and **dotenv** were installed to handle the server, database, and environment configuration.

2. MongoDB Setup and Connection

- A MongoDB instance was created using **MongoDB Atlas**, and the connection was established via **Mongoose**.
- The database connection is working correctly, and the application can successfully interact with the MongoDB database.

3. API Endpoints

- Two API endpoints were implemented:
  - **POST /api/note**: Creates a new note with a unique URL.
  - **GET /api/note/**

: Retrieves the note using its unique URL.

- Both endpoints have been tested using **Postman** and are functioning as expected.

What Is Working

- **MongoDB Connection**: The database connection was successfully established, and the backend can communicate with MongoDB.
- **API Endpoints**: The two primary API endpoints (POST for note creation and GET for note retrieval) have been implemented and tested.
- **Data Validation**: Basic validation is in place to ensure that required fields (content and url) are provided when creating a note.

What Is Not Working

- No major issues have been encountered so far. The backend is working as intended for the basic API routes.

Planned but Not Completed

- **Note Encryption**: Future versions will add encryption to the note content before storing it in the database.
- **Expiration Feature**: Notes currently do not expire, but future releases will include a time-to-live (TTL) feature where notes auto-delete after a certain time.
- **User Authentication**: No user authentication has been implemented yet, but this is planned for later stages.

Next Steps

- Implement additional features like note encryption and expiration.
- Start working on user authentication to restrict note access to authorized users.
- Begin the front-end work to provide a user interface for interacting with the API.

Conclusion

For Week 3, the project has successfully met the foundational requirements, including project setup, database connection, and initial API functionality. The backend is operational, and the next phase will focus on additional features like encryption, authentication, and the front-end.