**Activity 2: Notes API + UI**

Personal notes app with complete user authentication (login, registration, password reset via email) and full CRUD functionality. SQLite-based—runs immediately without XAMPP installation. MySQL-ready for production deployment.

**Members:**

Andaya, Rhodel

Benito, Patrick Raymond P.

Cielo, Ivan Louis D.

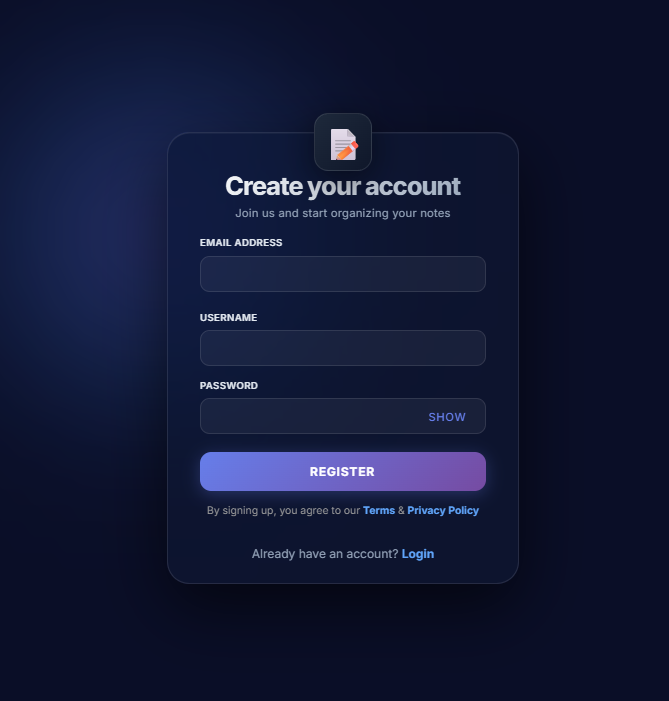
De Leon, Justin Carl V.

Dela Cruz, Althea

Dela Cruz, Filbert B.

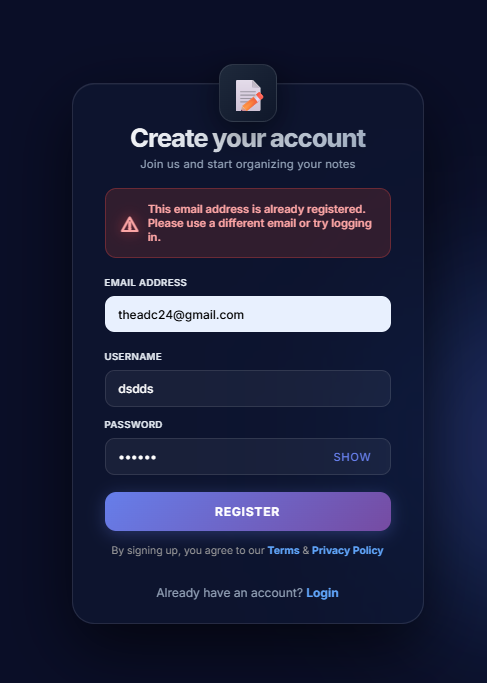
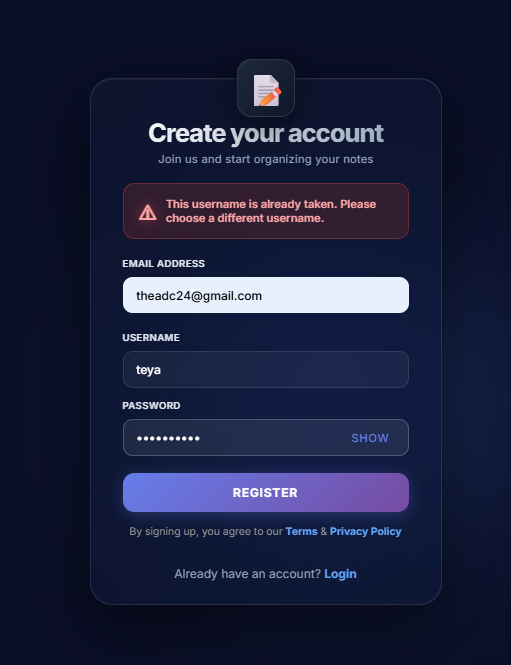
**Steps in showing functionality:**

**Registration Functionality**



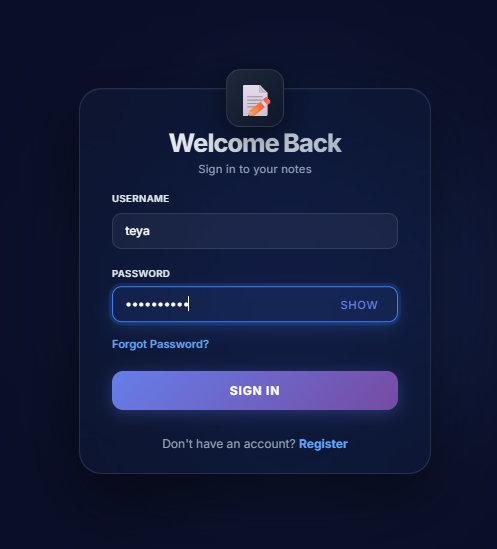
The application begins with a user registration screen where new users can create an account. Users are required to provide three pieces of information: an email address, a username, and a password. The password field includes a "SHOW" toggle button for visibility control. After filling out the form, users click the "REGISTER" button to create their account. By registering, users agree to the Terms & Privacy Policy. For existing users, a "Login" link is provided at the bottom to access their accounts directly.

**Registration with Validation**



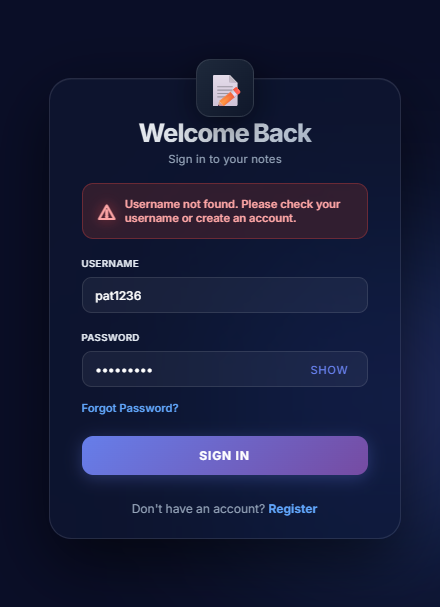
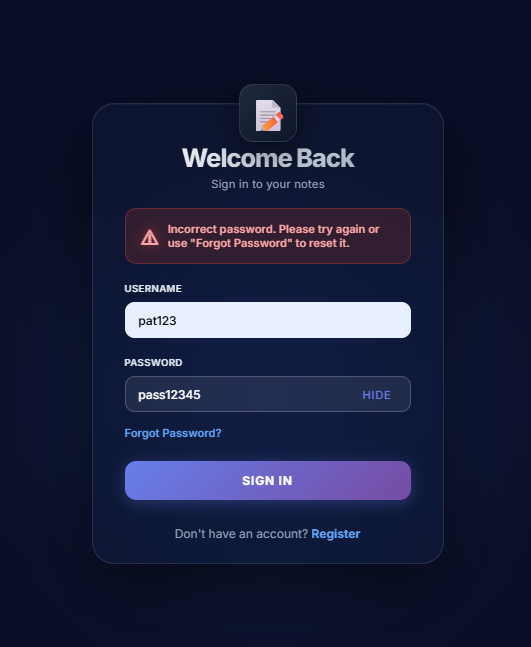
The registration form includes real-time validation to ensure unique usernames and email addresses. If a user attempts to register with a username that already exists in the system, an error message appears stating "This username is already taken. Please choose a different username." Similarly, if an email address is already registered, the system displays an error message: "This email address is already registered. Please use a different email or try logging in." These validation checks prevent duplicate accounts and guide users to either choose alternative credentials or log in to their existing account. The form retains entered information while highlighting the specific field that needs correction.

**Login Functionality**



After successful registration, users are redirected to the login screen with the message "Welcome Back - Sign in to your notes." Users can log in using their registered username (or email) and password. The password field includes a "SHOW" toggle for visibility control. A "Forgot Password?" link is available for account recovery. For new users who haven't registered yet, a "Register" link at the bottom redirects them to the account creation page.

**Login Functionality**

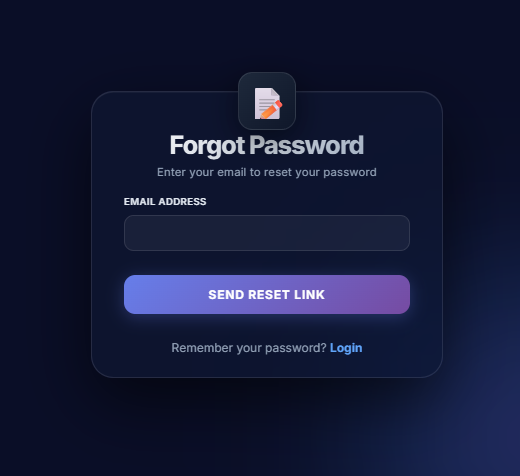
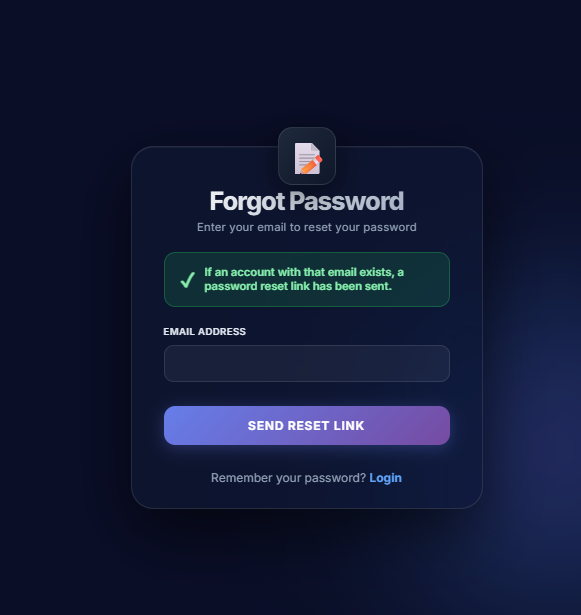


After successful registration, users are redirected to the login screen with the message "Welcome Back - Sign in to your notes." Users can log in using their registered username (or email) and password. The password field includes a "SHOW/HIDE" toggle for visibility control.

The login form includes validation to verify credentials:

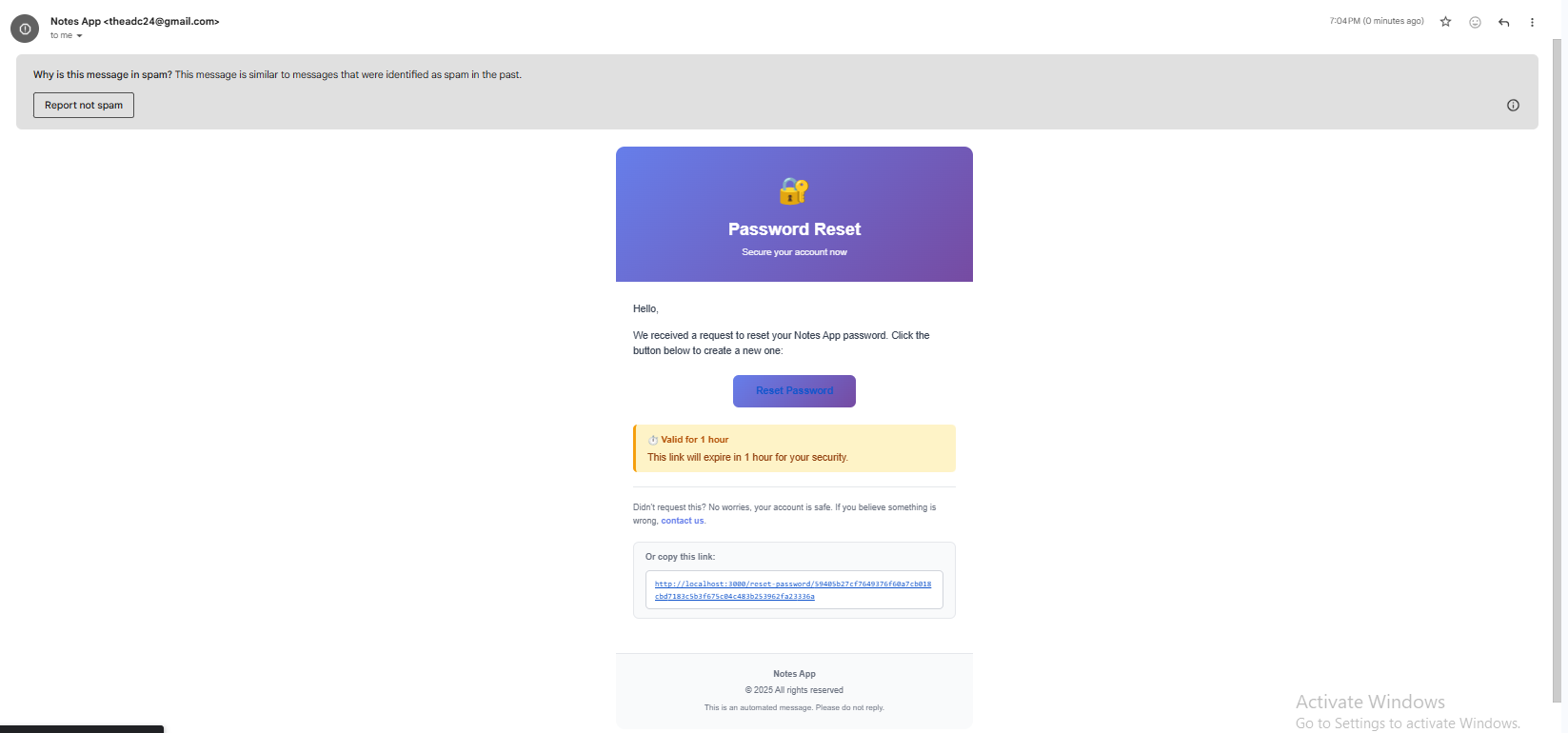
* If the username entered does not exist in the system, an error message displays: "Username not found. Please check your username or create an account."
* If the username exists but the password is incorrect, an error message appears: "Incorrect password. Please try again or use 'Forgot Password' to reset it."

**Password Recovery Functionality**



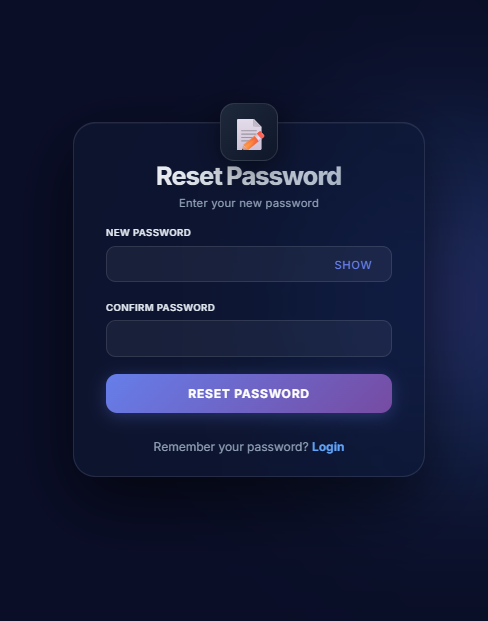
When users click the "Forgot Password?" link from the login screen, they are directed to a password recovery page. Users must enter their registered email address to initiate the password reset process. After clicking the "SEND RESET LINK" button, the system verifies if an account exists with that email address. If the email is registered in the system, a success message appears: "If an account with that email exists, a password reset link has been sent." This message protects user privacy by not explicitly confirming whether an email is registered. Users who remember their password can return to the login screen via the "Login" link at the bottom.

**Email Password Reset Link**



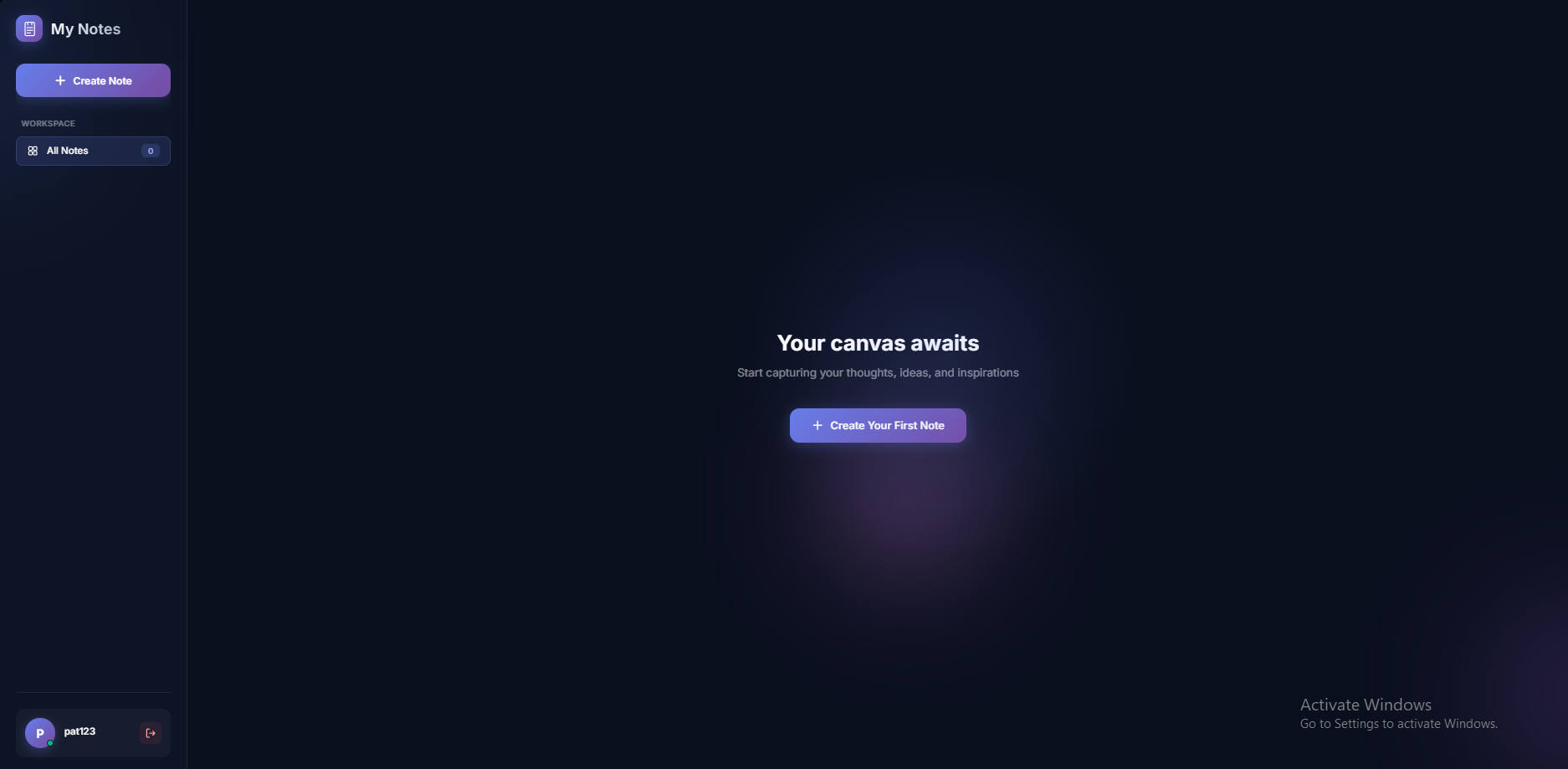
Upon successful submission of a password reset request, the system sends an automated email to the user's registered email address. The email includes a "Reset Password" button and an alternative copy-able link for password recovery. For security purposes, the reset link is valid for only 1 hour and will expire after that time. The email also includes a reassurance message stating that if the user didn't request the reset, their account remains safe.

**Reset Password Page**



When users click the reset link from their email, they are directed to a "Reset Password" page where they can create a new password. The form requires users to enter their new password twice: once in the "NEW PASSWORD" field and again in the "CONFIRM PASSWORD" field to ensure accuracy. The new password field includes a "SHOW" toggle for visibility control. After entering matching passwords and clicking the "RESET PASSWORD" button, users are automatically redirected to the login page where they can sign in using their newly created password.

**Dashboard - Main Interface**

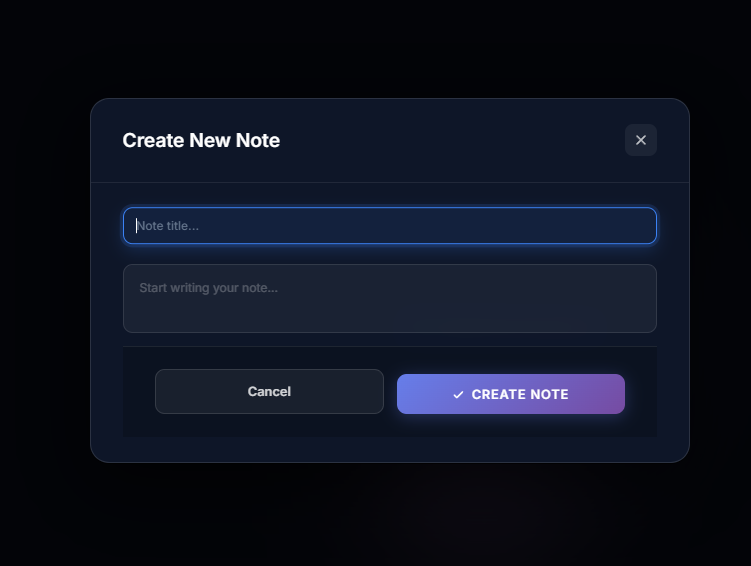


After successful login, users are directed to the main dashboard displaying an empty state with the message "Your canvas awaits - Start capturing your thoughts, ideas, and inspirations." The interface features a left sidebar containing:

* "My Notes" header at the top
* A "+ Create Note" button for quick note creation
* A "WORKSPACE" section showing "All Notes" with a counter displaying "0" notes

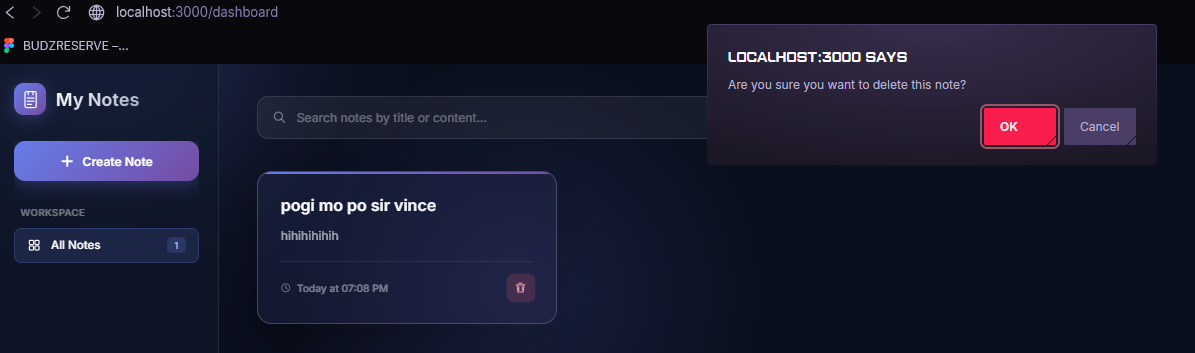
The center of the screen displays a "+ Create Your First Note" button, inviting new users to begin creating their first note. At the bottom left corner, the user's profile information is displayed, showing their username with a logout option.

**Create New Note Modal**



Clicking "+ Create Note" opens a modal with a "Note title..." input field and a text area for content. Users click "CREATE NOTE" to save or "Cancel" to close without saving. A close button (X) is available in the top-right corner.

**Dashboard with Created Notes**

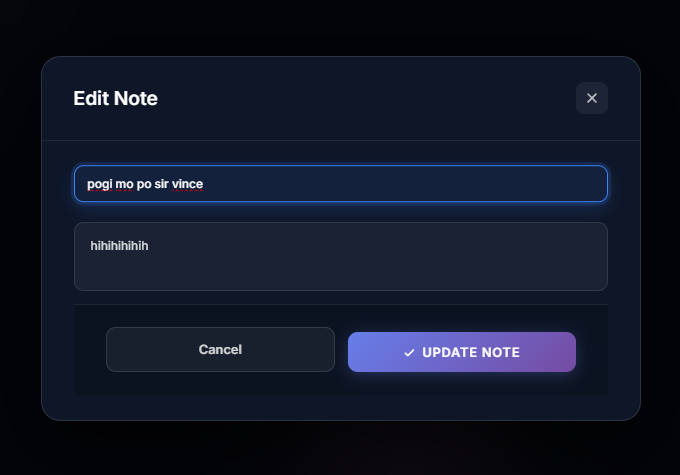


Once a note is created, it becomes visible on the dashboard. The interface now displays:

* A search bar at the top for finding specific notes
* The created note card showing the note title, a preview of the content, and a timestamp
* A delete icon (trash bin) on each note card for removing notes
* The sidebar's "All Notes" section now displays a counter showing the total number of notes created

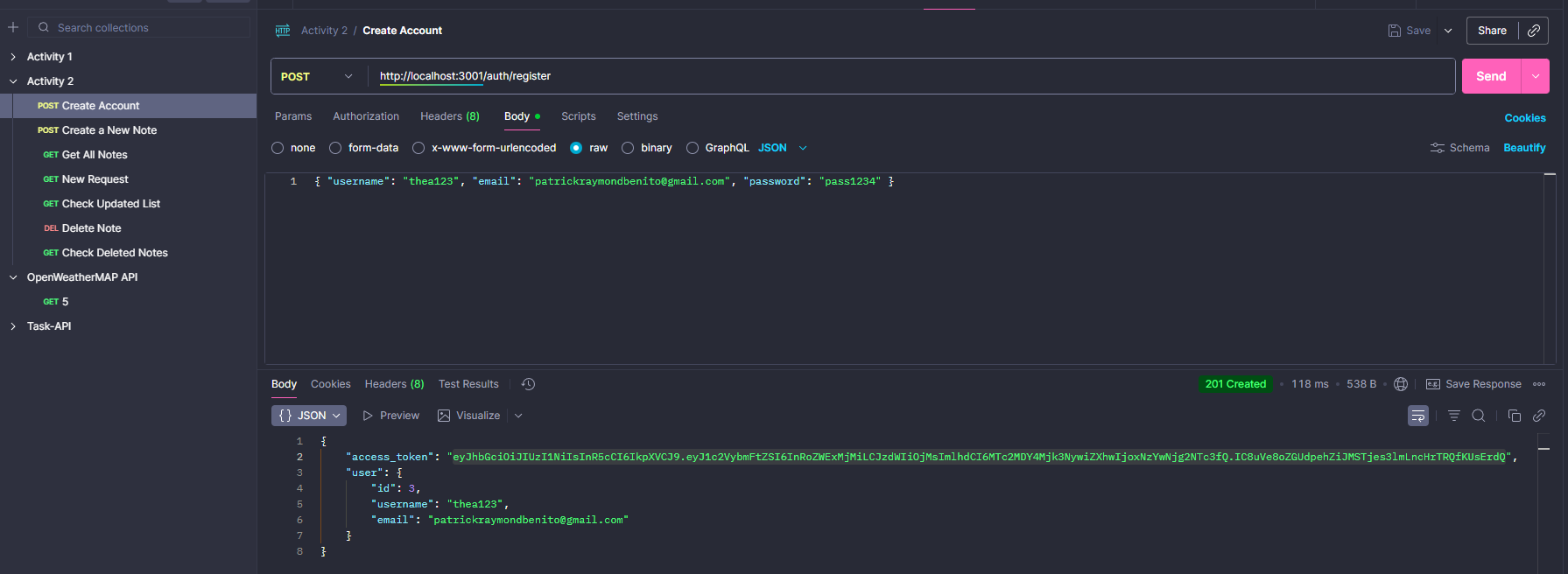
When users click the delete icon, a confirmation dialog appears with the message "LOCALHOST:3000 SAYS - Are you sure you want to delete this note?" Users can click "OK" to confirm deletion or "Cancel" to keep the note.

**Edit Note Modal**

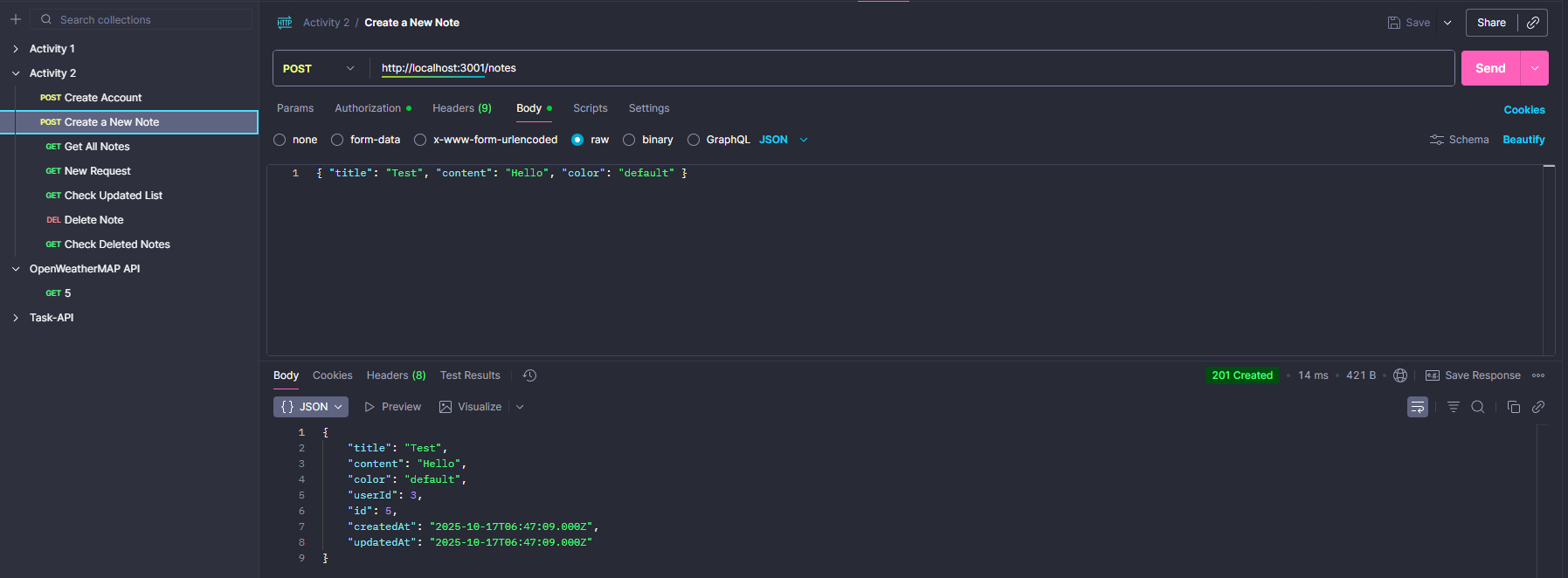


When users click on a note card from the dashboard, an "Edit Note" modal appears displaying the existing note's title and content in editable fields. Users can modify the note title in the top input field and edit the note content in the larger text area below. After making changes, users can click "UPDATE NOTE" to save the modifications or "Cancel" to discard changes and close the modal. A close button (X) is also available in the top-right corner.

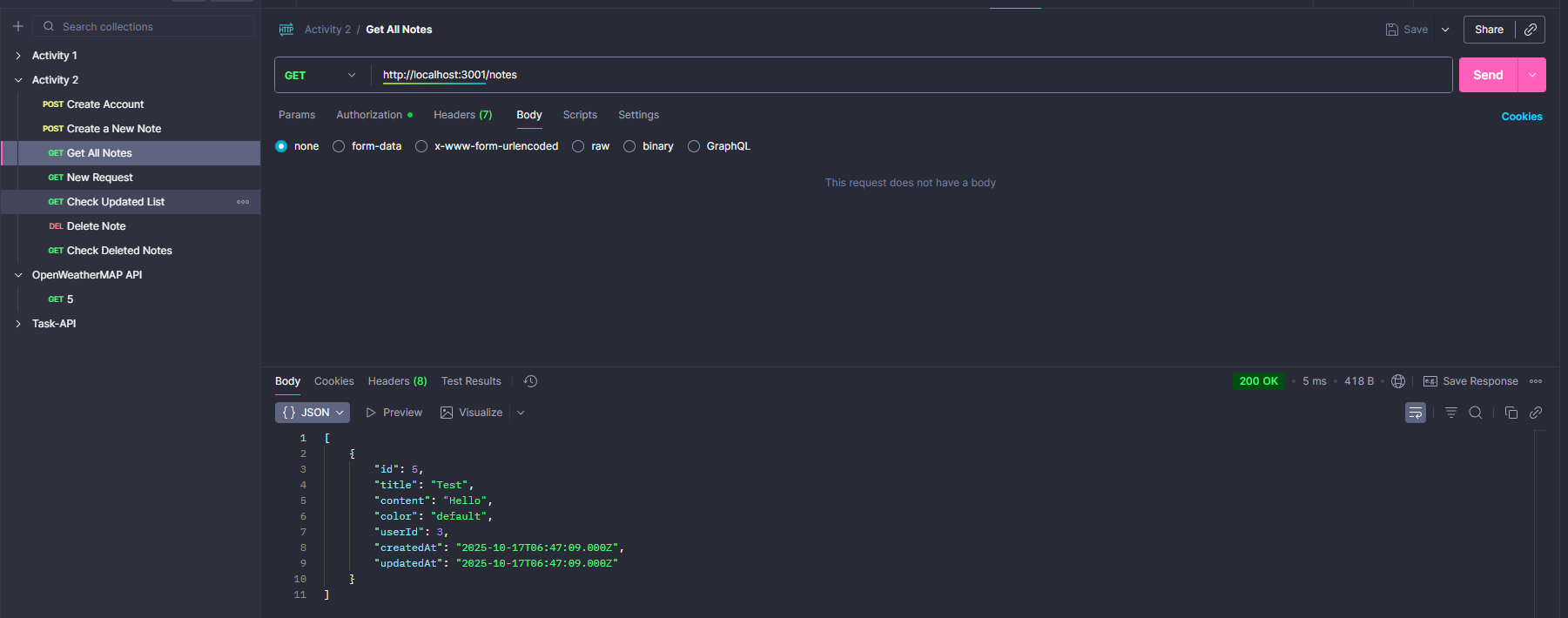
**API Connectivity Testing:**

**Create Account Endpoint**

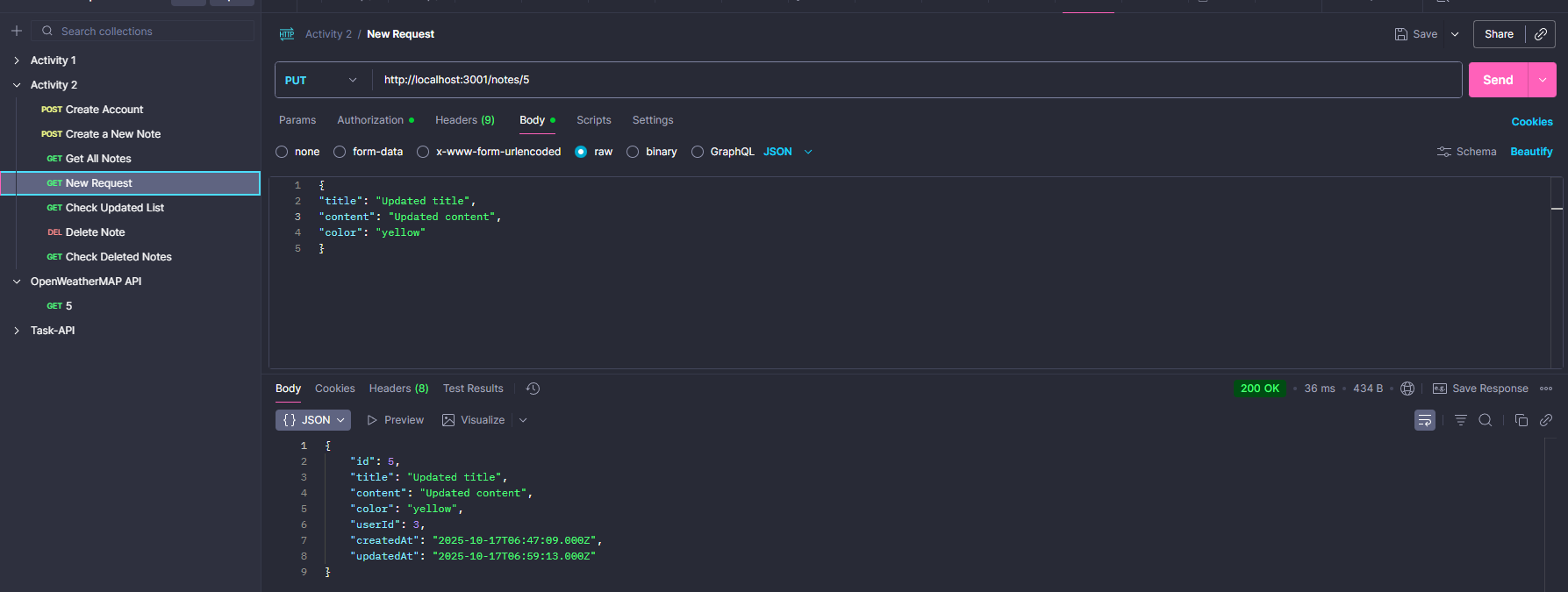
The Create Account endpoint is tested using Postman with a POST request to http://localhost:3001/auth/register. The request body contains user registration data in JSON format with username, email, and password. Upon successful registration, the API returns a 201 Created status with a response containing an access token for authentication and the created user's information (id, username, email).

**Create a New Note Endpoint**

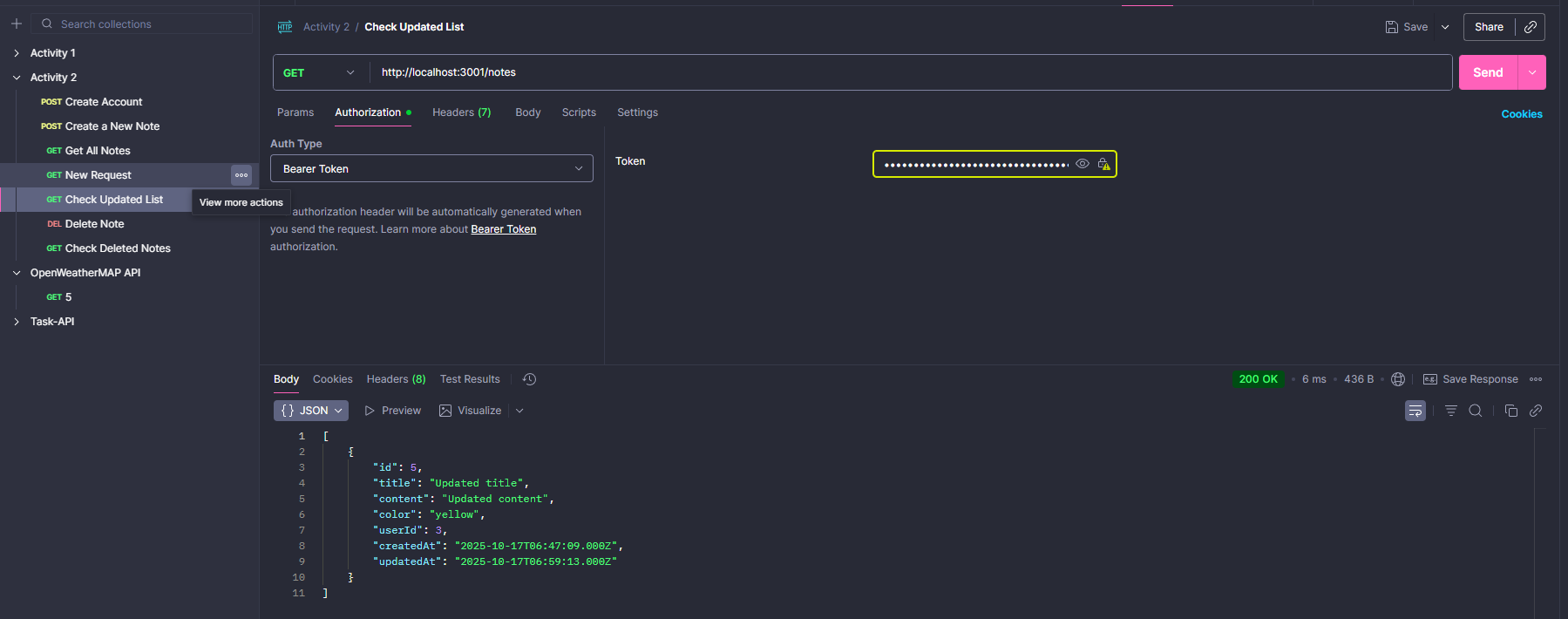
The Create a New Note endpoint is tested using Postman with a POST request to http://localhost:3001/notes. The request requires authorization and sends a JSON body containing note details: title, content, and color. Upon successful creation, the API returns a 201 Created status with a response containing the complete note object including the generated id, userId, createdAt timestamp, and updatedAt timestamp.

**Get All Notes Endpoint**

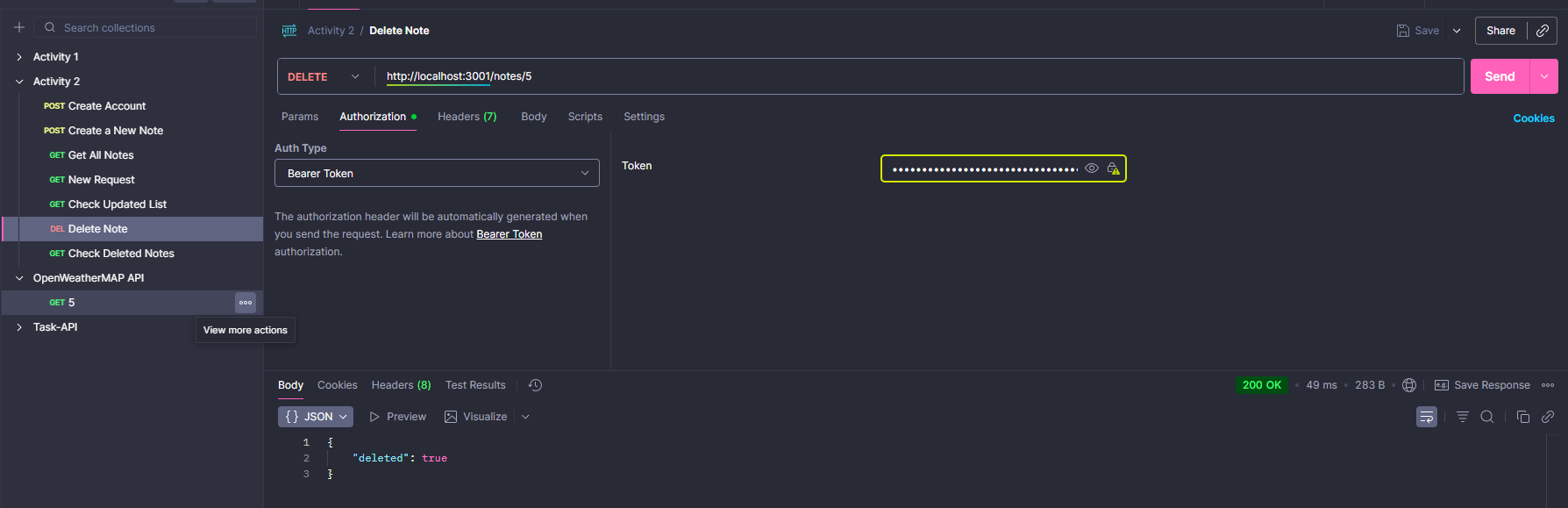
The Get All Notes endpoint is tested using Postman with a GET request to http://localhost:3001/notes. This request requires authorization and does not include a request body. Upon successful retrieval, the API returns a 200 OK status with a JSON array containing all notes belonging to the authenticated user. Each note object includes id, title, content, color, userId, createdAt timestamp, and updatedAt timestamp.

**Update Note Endpoint**

The Update Note endpoint is tested using Postman with a PUT request to http://localhost:3001/notes/5. This request requires authorization and sends a JSON body containing the updated note details: title, content, and color. Upon successful update, the API returns a 200 OK status with the complete updated note object including id, the modified fields, userId, createdAt timestamp, and a new updatedAt timestamp

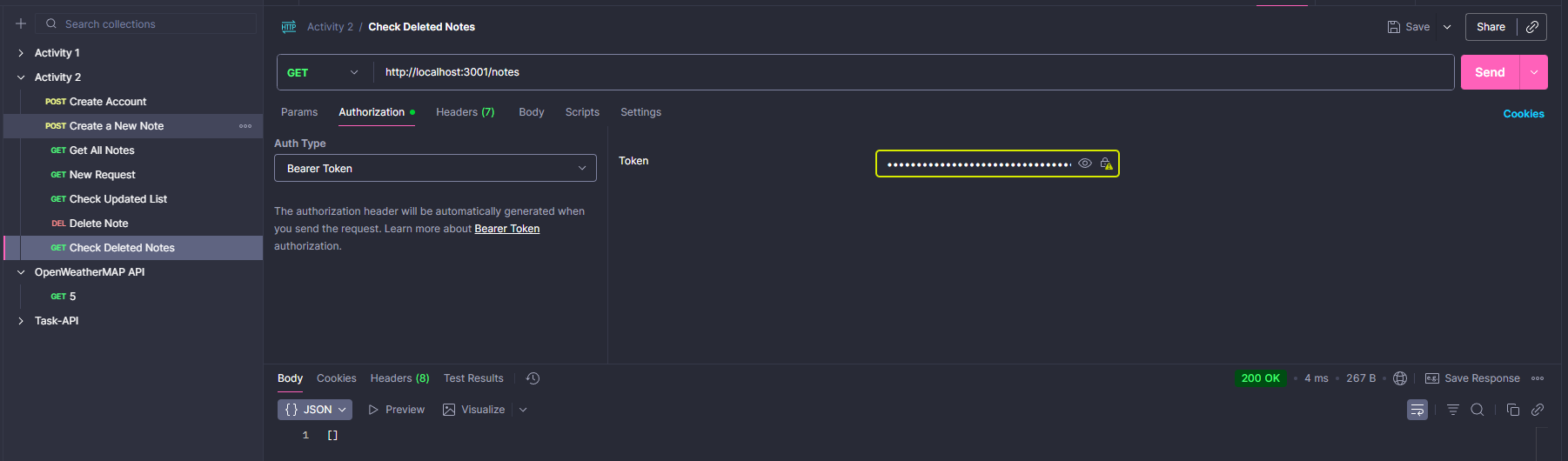
**Check Updated List Endpoint**

A GET request to http://localhost:3001/notes verifies the updated note information. The endpoint uses Bearer Token authentication and returns a 200 OK status displaying the note with updated fields: id, title, content, color, userId, createdAt, and updatedAt.

**Delete Note Endpoint**

A DELETE request to http://localhost:3001/notes/5 removes the specified note from the database. The endpoint uses Bearer Token authentication and returns a 200 OK status with a JSON response {"deleted": true} confirming successful deletion.

**Check Deleted Note Endpoint**



The Check Deleted Notes endpoint is tested using Postman with a GET request to<http://localhost:3001/notes>. This request requires Bearer Token authorization to authenticate the user. Upon successful retrieval, the API returns a 200 OK status with an empty array, indicating that there are no deleted notes in the system.

# **Project Setup Guide**

This guide will walk you through setting up and running the Notes Application on your local machine.

## **Prerequisites**

* **Node.js 18+** and **npm** - [Download here](https://nodejs.org/)

## **Setup Instructions**

### **Step 1: Backend Setup**

#### **1.1 Install dependencies**

cd backend

npm install

#### **1.2 Create environment file**

Create a file named .env inside the backend/ folder:

DB\_TYPE=sqlite

DB\_DATABASE=database.sqlite

JWT\_SECRET=your\_secret\_key\_here

EMAIL\_USER=your\_email@gmail.com

EMAIL\_PASSWORD=your\_google\_app\_password

FRONTEND\_URL=http://localhost:3000

DB\_NAME=notes\_app

#### **1.3 Configure Gmail App Password**

To enable password reset emails:

1. Go to [Google Account Security](https://myaccount.google.com/)
2. Click **Security** → Enable **2-Step Verification**
3. Go to **App passwords** under "How you sign in to Google"
4. Create a new app password (name it "Notes App")
5. Copy the 16-character password
6. Paste it in .env as EMAIL\_PASSWORD (remove all spaces)

**Update your .env:**

* Replace your\_email@gmail.com with your Gmail address
* Replace your\_secret\_key\_here with any random string
* Replace your\_google\_app\_password with the app password

#### **1.4 Start the backend**

npm run start:dev

Server runs at http://localhost:3001

### **Step 2: Frontend Setup**

Open a new terminal:

cd frontend

npm install

npm start

App opens at <http://localhost:3000>

## **Troubleshooting**

**Backend won't start?** Check port 3001 is available

**Frontend won't start?** Check port 3000 is available

**Emails not sending?** Verify Gmail App Password is correct and 2-Step Verification is enabled

**Database errors?** Delete backend/database.sqlite and restart