User Management API with MongoDB Integration

This project builds on the initial User Management API by adding MongoDB as a database, providing persistent storage for user data. Mongoose is used to manage the connection, define schemas, and validate data.

Key Aspects:

- **MongoDB Connection**: Connects Node.js to MongoDB through Mongoose, enabling persistent data storage.
- **User Schema**: Defines a structured schema with Mongoose, setting required fields and data types to enforce data integrity.
- **CRUD Routes**:
  - **GET /users** – Fetches all users from MongoDB.
  - **GET /users/**
    – Retrieves a user by their MongoDB ObjectId.
  - **POST /user** – Adds a new user and saves it to the database.
  - **PUT /user/**
    – Updates a user's information.
  - **DELETE /user/**
    – Deletes a user by ObjectId.

Challenges:

- **Error Handling**: Requires careful async error handling for database interactions.
- **ObjectId Usage**: Manages MongoDB's ObjectId format, different from simple integer IDs.
- **Middleware Enhancements**: Handles more complex request logging and input validation.

This transition also involves advanced middleware to manage requests and respond with relevant status codes. The additional complexity of handling asynchronous database operations adds a level of difficulty, making it important to implement proper async/await handling and error-catching mechanisms.

GitHub Link:

https://github.com/Ashmita2282/userAPI-MongodbMngt

Project Structure

```
UserAPIManagement/
├── controller/
│   └── userController.js
├── middleware/
│   └── logger.js
├── routes/
│   └── userRoutes.js
├── config/
│   └── database.js
├── models/
│   └── userModel.js
├── nodes/
│   └── various file
├── .env
├── index.js
└── package.json
```

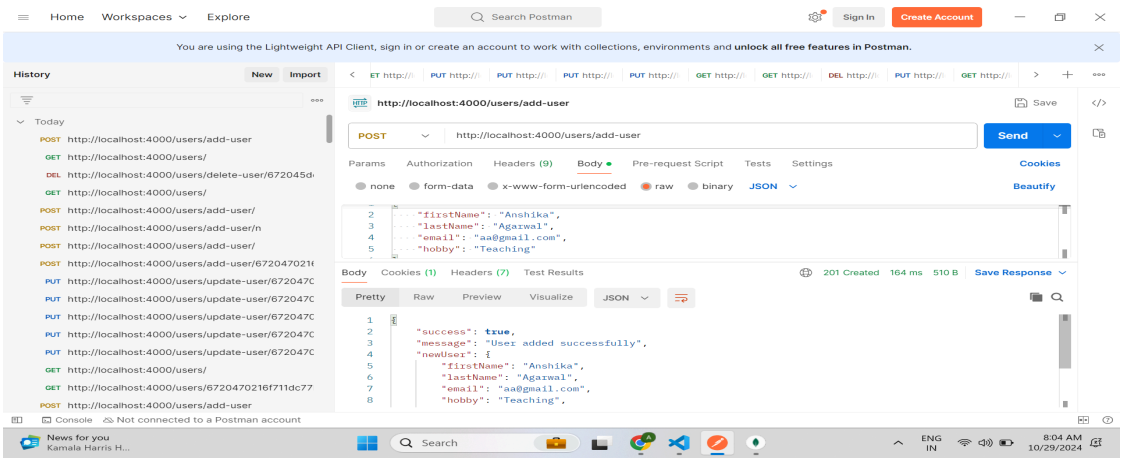
ScreenShots —>

Having screenshot of PostMan Testing and also For MongoDB Compass for all the API's whether it is adding user → POST, Updating User →PUT , Deleting User → DELETE, fetch all user details → GET , fetch only one user by id → GET

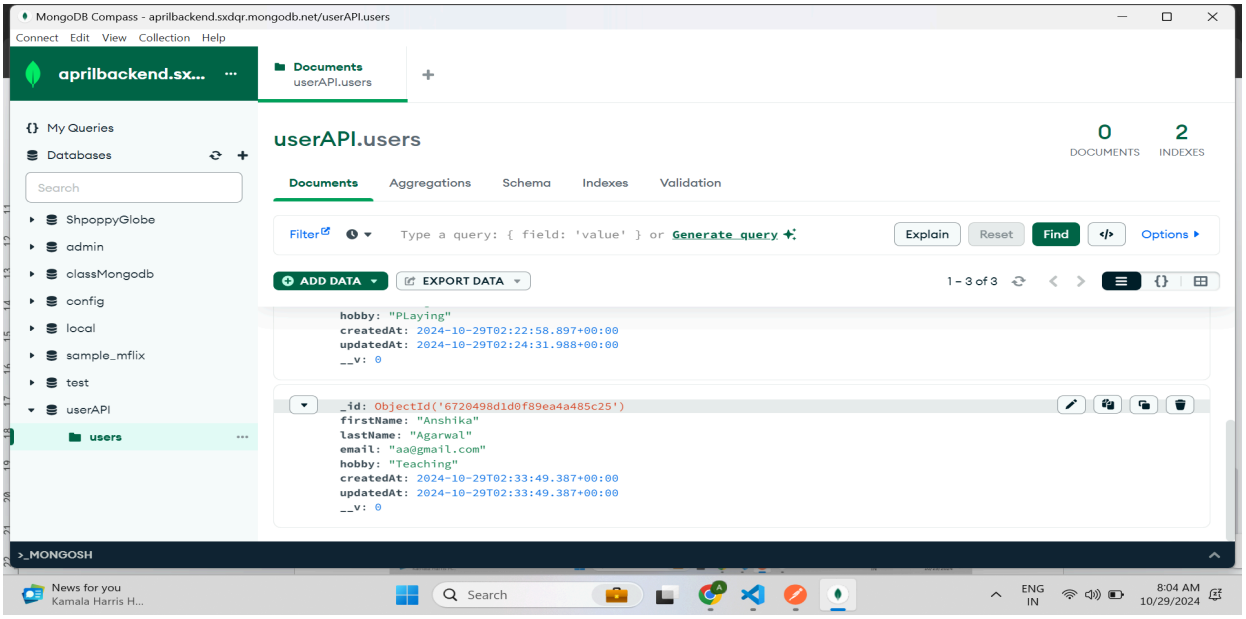
API CAll for Add → POST

1. Added new user with name : Anshika and hobby : teaching with Unique email Id too

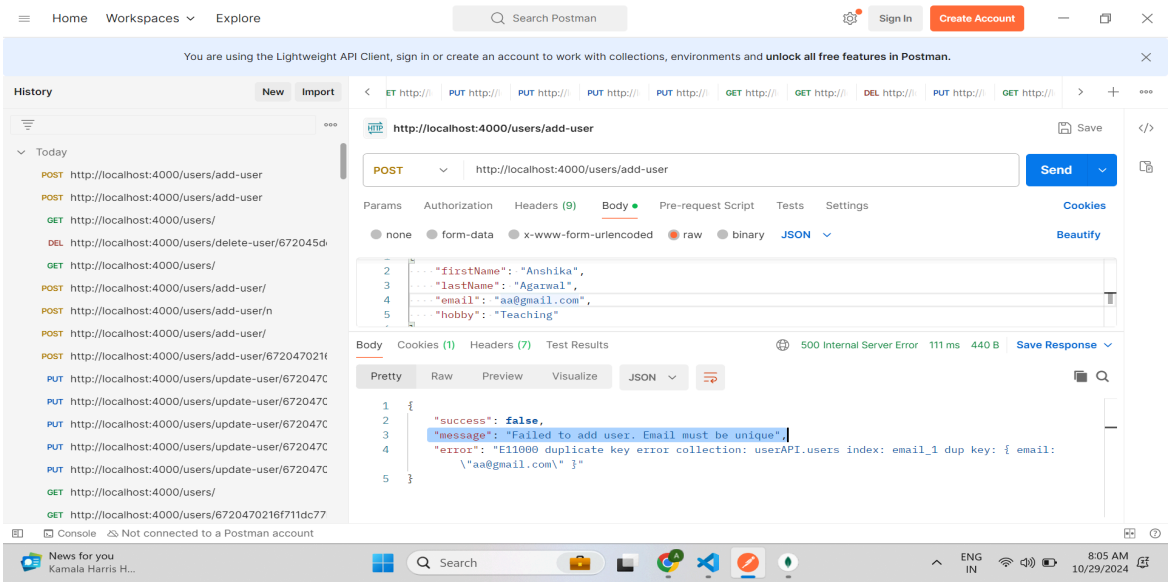
POSTMAN SS



MONGODB COMPASS Showing added user

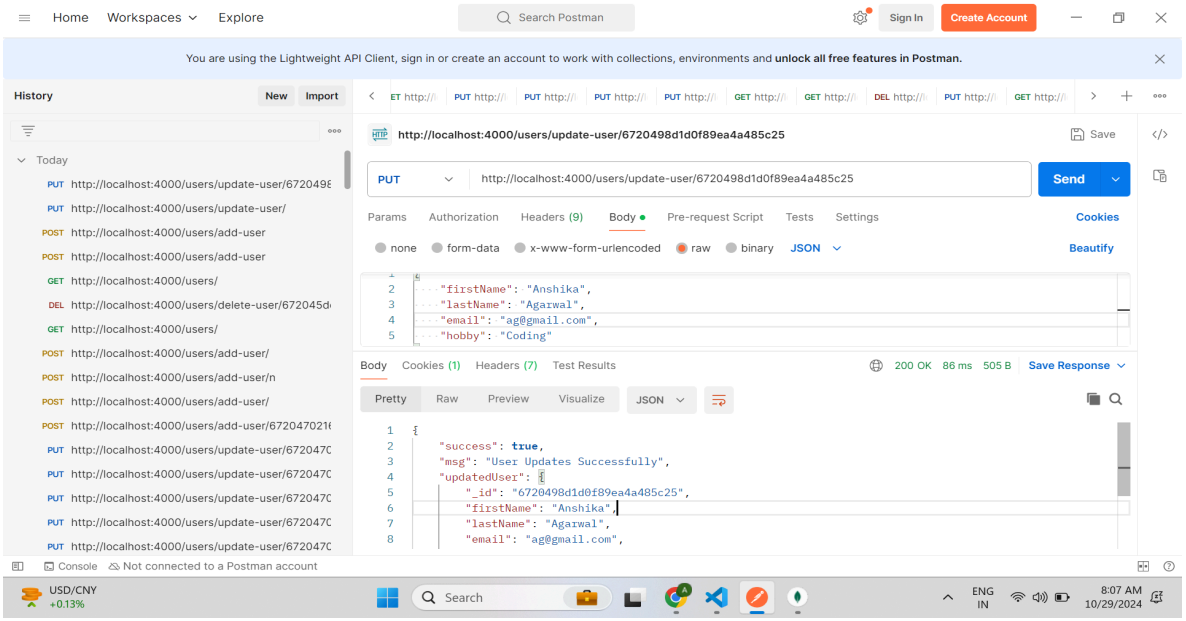


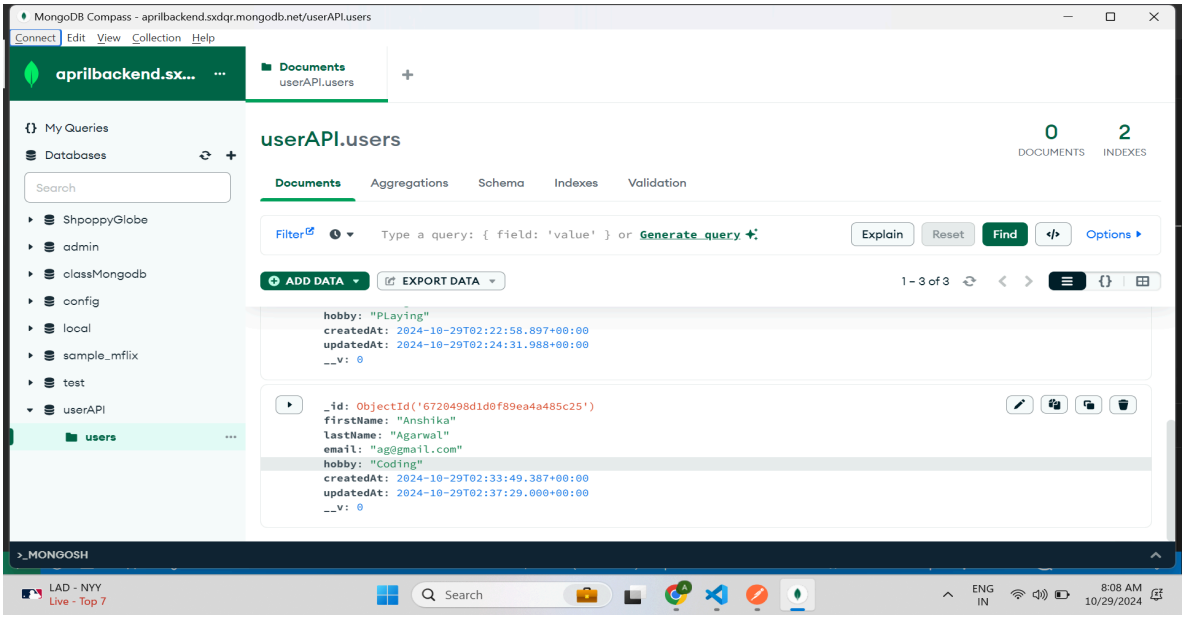
Error for email should be unique (Email Validation)



API CAll for Update User by id → PUT

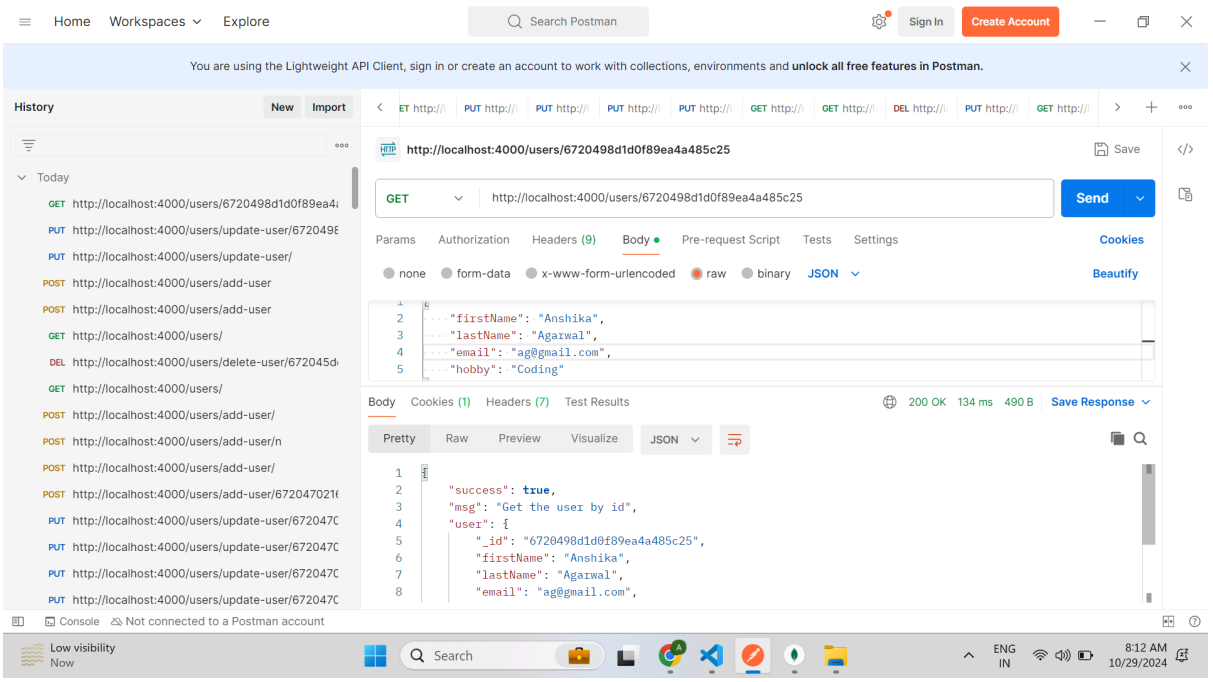
1. Updated user with email :ag@gmail.com and hobby: coding

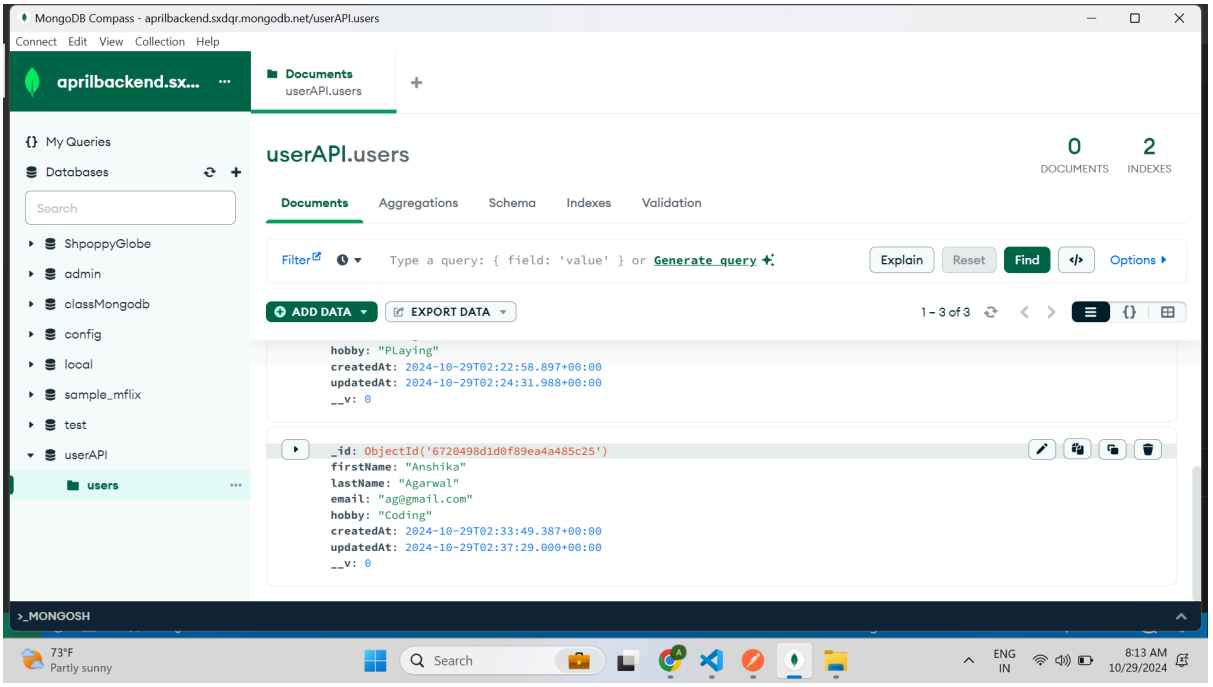




API CAll for Fetch User by id → GET

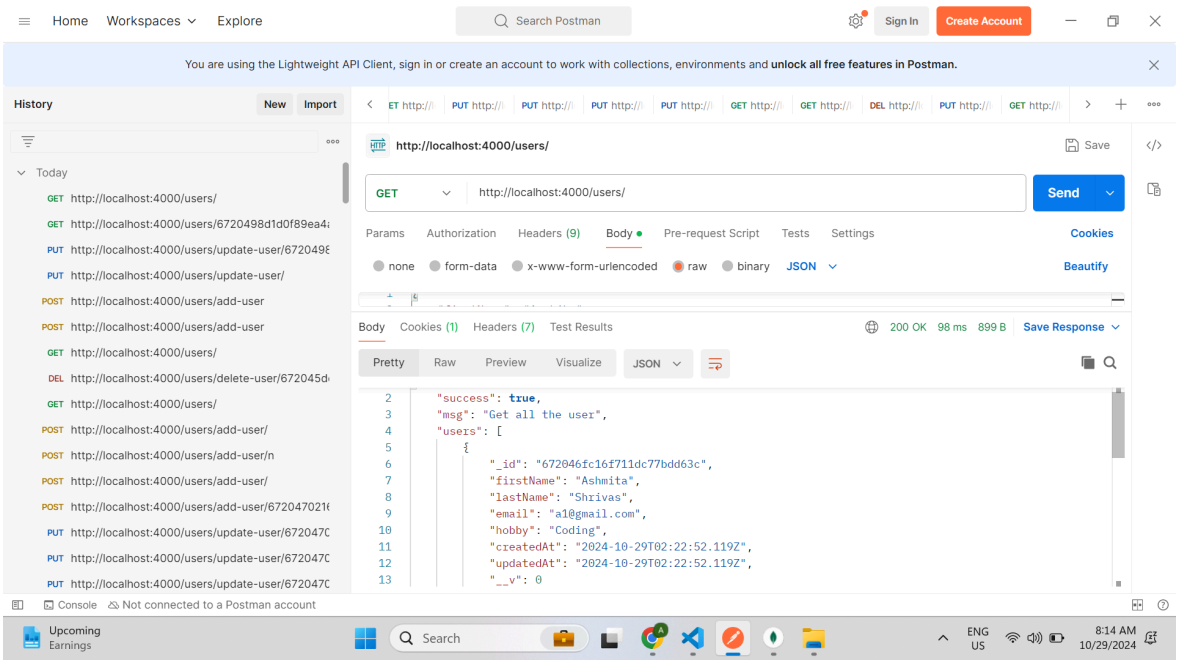
1. Get the user with id

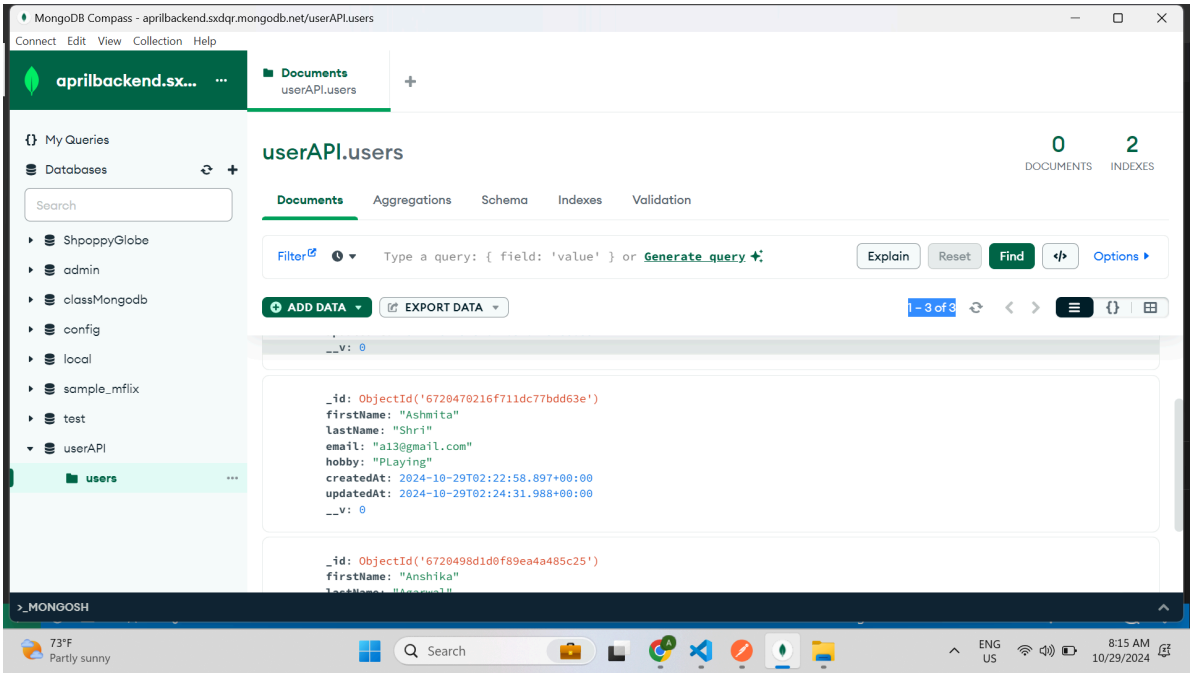




API CAll for Fetch All User → GET

1. Get all user





API CAll for Delete User by id → Delete

1. Deleted user with name : Anshika email :ag@gmail.com and hobby: coding