PostCare.AI - TASK 2(Rohit Arrunachalam)

Link to Github Repo

Link to Demo Video

Task 2: Implement the Prototype with React and MongoDB

Objective:
For this task, you will take your Figma prototype into development by implementing it using
React for the front end. You'll also set up a MongoDB database to store prototype chats locally
on your machine. The goal is to create a working prototype that can be run locally, demonstrating the functionality you've designed. Remember, the focus is on the front end; you
are not required to implement any language model capabilities.
Requirements:

1. React Implementation: Develop the front end of your redesigned ChatGPT prototype using
   React. Your implementation should be responsive and adhere closely to your Figma design in
   Task 1. The application should be set up to run locally on a machine using tools such as Visual
   Studio Code (VScode) for development.
2. MongoDB Database Setup: Install and set up a MongoDB database on your local machine

to store the chat interactions. Design the schema in a way that supports future enhancements,
such as adding a language model capability (The language model capabilities are optional
but a good to have).

3. API Calls: Implement API calls that simulate fetching and sending messages to the chat
interface. These should interact with your MongoDB database to store and retrieve chat data.
The focus is on demonstrating how the system could work with actual data, rather than on the
chat logic itself.
4. GitHub: Push your code along with sample MongoDB database to a public GitHub
repository.

Solution:

- Features
    - Authentication
    - Email Verification
    - Chat
    - Responsive
    - Clear Chats
    - OpenAI API Integration
    - Extra Features
        - Search Bar for searching through Chat History
        - Related Questions Feature (Gives follow-up questions related to the prompt)
        - Folders (Group multiple chats into one folder for quicker access)

        - Healthcare Related (Send prompts with sub string #Symptoms, #Appointment, #Remainder, #Emergency to get response related to the prompt)
            - Symptoms Checker
            - Pill Remainder
            - Appointment Scheduler
            - Emergency Tele Assistant
            - Only Restricted to HealthCare related queries

- Tech Stack
    - React
    - Node.js
    - TailwindCSS
    - MongoDB
    - Redux
    - Nodemailer
    - Vite

- Step to Run the Application
    - client
        - Run the following commands in the /client directory

```
cd client
npm i
npm run dev
```

    - server

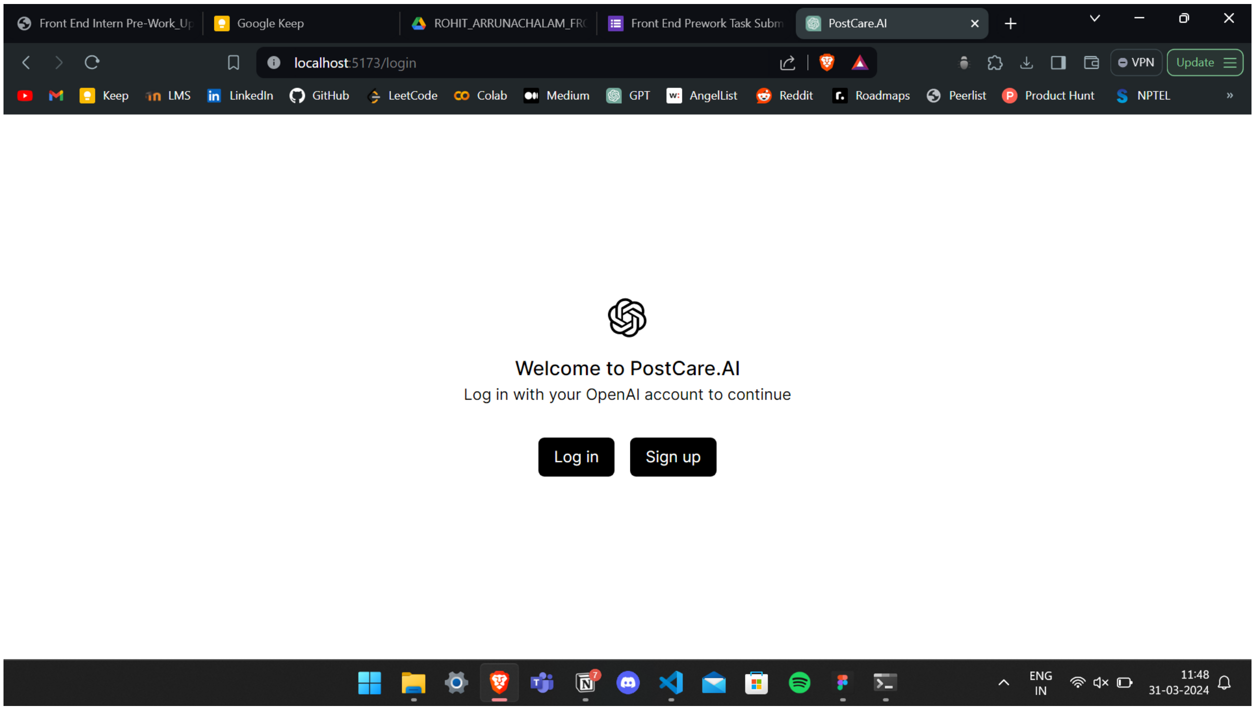
- Run the following commands in the /server directory

```
cd server
npm i
npm start
```

- And set up a .env file in the /server directory

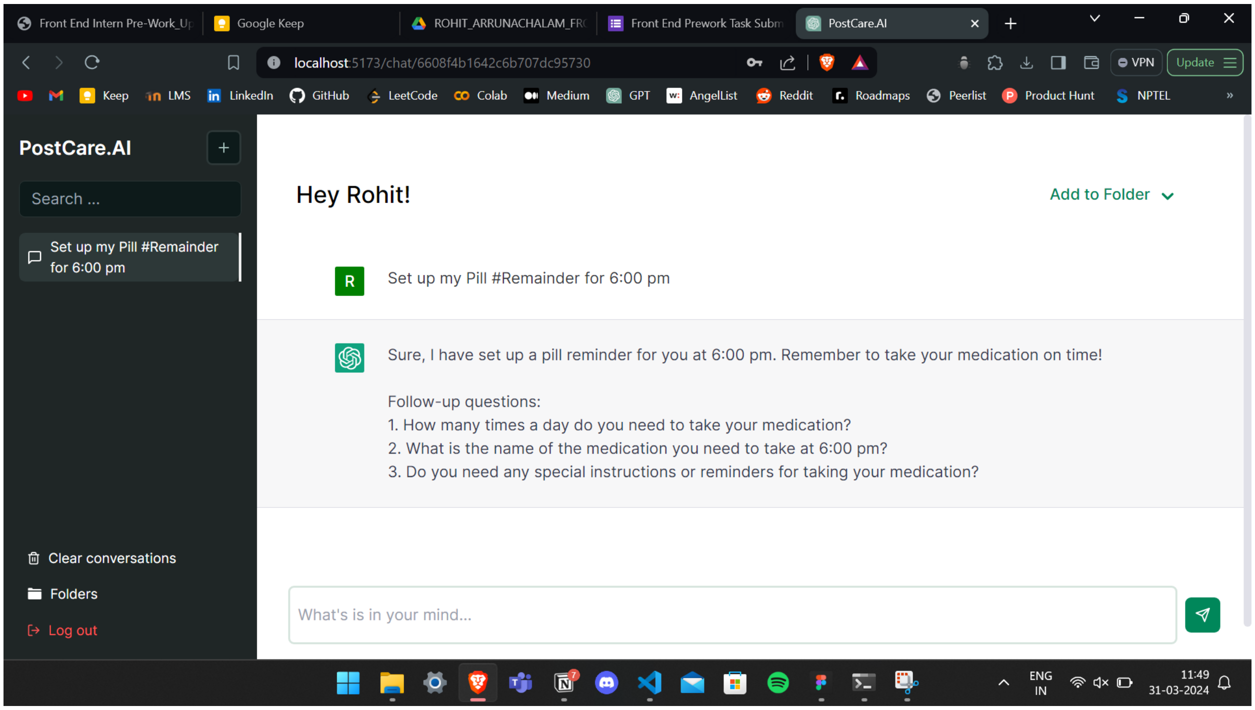
```
PORT = 5000
MONGO_URL = "mongodb://localhost:27017"
SITE_URL = "http://localhost:5173"
JWT_PRIVATE_KEY = "some-private-key"
OPENAI_API_KEY = "your-open-api-key"
MAIL_EMAIL = "mail-id-for-mail-verification"
MAIL_SECRET = "mail-app-password"
```

- Screenshots



Welcome to PostCare.AI

Log in with your OpenAI account to continue

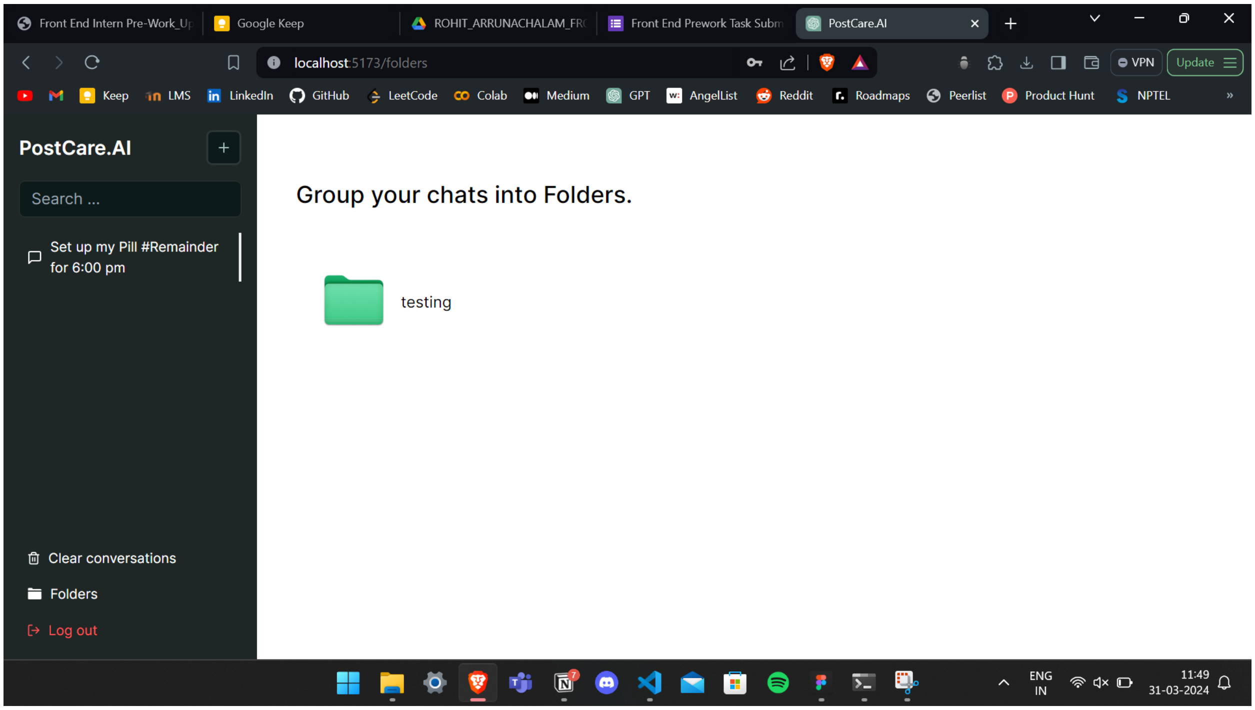


Hey Rohit!

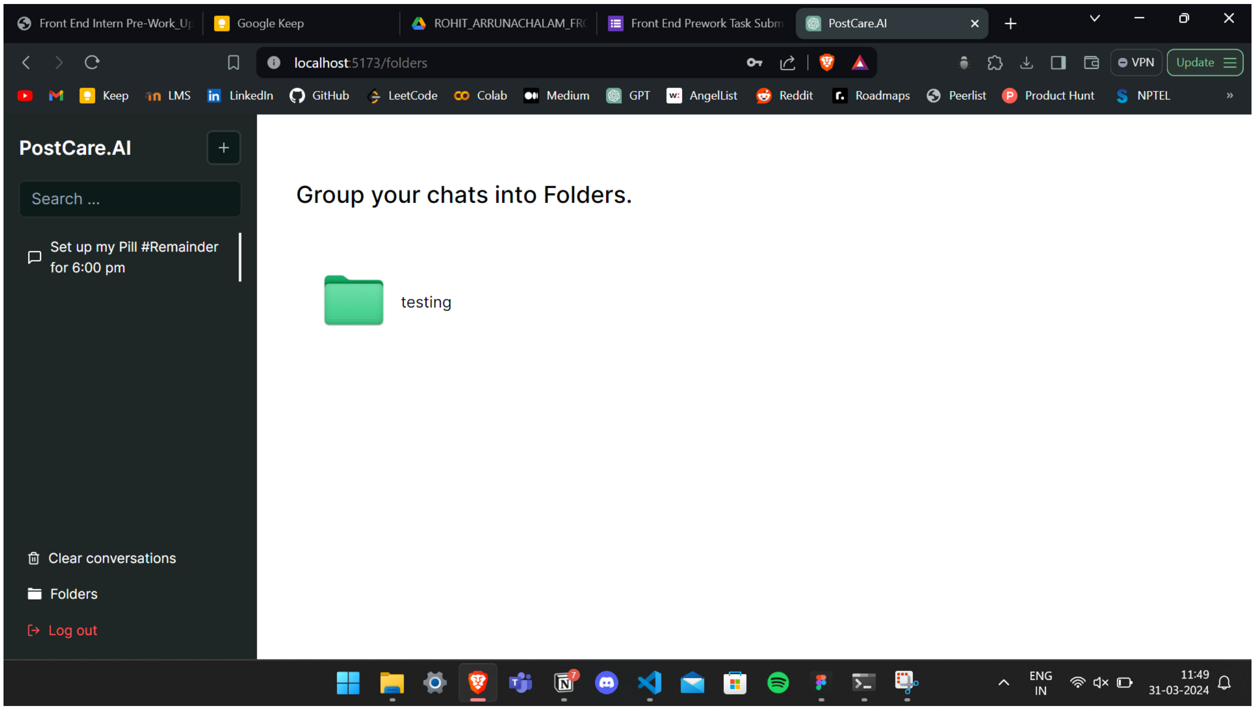
Set up my Pill #Remainder for 6:00 pm

Sure, I have set up a pill reminder for you at 6:00 pm. Remember to take your medication on time!

Follow-up questions:
1. How many times a day do you need to take your medication?
2. What is the name of the medication you need to take at 6:00 pm?
3. Do you need any special instructions or reminders for taking your medication?



Group your chats into Folders.



Set up my Pill #Remainder for 6:00 pm

Group your chats into Folders.

testing