**MAGIZH Hospital Management System**

**Technologies used:**

**Frontend**-visual studio and React

**Backend**-Intellij Springboot

**Database** -MySQl and MYSQL workbench

**VideoLink-** **https://drive.google.com/file/d/1SWkahe279HiVk2Nhv2AlnY\_68LNdrObZ/view?usp=drive\_link**

Magizh Hospital Management System (MHMS) is a software solution designed to streamline the operations and processes within a healthcare facility, such as hospitals, clinics, and medical centers. It integrates various administrative, financial, and clinical functions to enhance efficiency, accuracy, and quality of patient care. Here are some key components and features typically found in a Hospital Management System:

1. **Patient Management**: Registration for new patient and login using registered details. Once entered into the patient dashboard patients can view list of doctors and book appointment with slots and date convenient to patient, view/cancel appointment and view prescription.
2. **Doctor management**: Doctors can login but cannot be registred from frontend for security purpose. Only admin can add doctors login credentials, Doctors can view patient list, add prescription, and cancel appointment from their side.
3. **Admin Components**: The system display doctors list, add doctors, update doctors, delete doctors and patients into database.
4. **User Accounts**: Registration and login functionality for passengers to create accounts, manage their bookings, view booking history, and update personal information.
5. **Security**: Implement Jwt security measures to protect user data, secure payment transactions, and prevent unauthorized access to the system.

**Project setup:**

**Steps to download frontend project from GitHub in VS code**

1. You can do this by pressing **Ctrl + Shift + P** (Windows/Linux) or **Cmd + Shift + P** (Mac). This will open a command palette at the top of the editor.
2. **Clone Repository**: In the command palette, type **Git: Clone** and select the option that appears. Alternatively, you can press **Ctrl + Shift + P** (Windows/Linux) or **Cmd + Shift + P** (Mac) and type **Git: Clone** and press Enter.
3. You will be prompted to enter the URL of the GitHub repository you want to clone. Paste the URL and press Enter.
4. Choose the local directory where you want to clone the repository. Once selected, press Enter.
5. If the repository is private or requires authentication, you may be prompted to provide your GitHub username and password or personal access token. Enter the required credentials and press Enter.
6. VSCode will start cloning the repository from GitHub to your local machine. You will see a progress indicator in the status bar at the bottom of the editor.
7. Once the cloning process is complete, VSCode will open the cloned repository in a new window automatically.
8. Enter cd busbooking
9. And type npm run dev
10. Open browser with http://127.0.0.1:5173/

**Steps to download backend project from GitHub in VS code**

1. **Open the Project from Version Control**: In the IntelliJ IDEA welcome screen, or if you have an existing project open, you can go to **File** > **New** > **Project from Version Control** > **Git**.
2. **Clone Repository**: In the URL field, enter the URL of the GitHub repository you want to clone. You can find this URL on the GitHub repository page. Then click **Clone**.
3. **Select Destination Folder**: Choose the local directory where you want to clone the repository. Once selected, click **Clone**.
4. **Authenticate (if required)**: If the repository is private or requires authentication, IntelliJ IDEA will prompt you to provide your GitHub username and password or personal access token. Enter the required credentials and click **OK**.
5. **Wait for Cloning to Complete**: IntelliJ IDEA will start cloning the repository from GitHub to your local machine. You will see a progress indicator in the bottom right corner of the IntelliJ IDEA window.
6. **Open Cloned Project**: Once the cloning process is complete, IntelliJ IDEA will open the cloned project in a new window automatically.
7. **Configure Project Settings (if necessary)**: Depending on the project, you may need to configure additional settings such as SDK, project structure, dependencies, etc.
8. **Run BusReservationBackendApplication file**
9. **Its runs in http://localhost:8080**

***Application.properties***

spring.application.name=hospitalManagement  
spring.jpa.hibernate.ddl-auto=update  
spring.datasource.url=jdbc:mysql://localhost/hospital\_management  
spring.datasource.username=root  
spring.datasource.password=kvlali2310  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
spring.jpa.show-sql:true

**MySQL work bench set up(**And add all the comments given below😊)

create database hospital\_management;

use hospital\_management;

insert into users(email,name,password,username)values("admin@gmail.com","admin","$2a$10$53BR2W9D0JXlXQJQKlSIM.alYjsArrV6oA.U5c7aMwJRzVxG3RXB6","admin");

insert into users(email,name,password,username)values("john@gmail.com","user","$2a$10$x4OkEBWj64beItTVQgbcFOjEi9bcqqySHQQ8SYwCV9.nHuUb07X5.","john");

insert into users(email,name,password,username)values("doctor@gmail.com","user","$2a$10$vwZ1ZkLEmODyWtaYntrcI.2OzpasCIp3V9AJBJ4bXM/pu1QiVQd4W","doctor");

insert into roles(name)values("ROLE\_ADMIN");

insert into roles(name)values("ROLE\_USER");

insert into roles(name)values("ROLE\_DOCTOR");

insert into users\_roles values(1,1);

insert into users\_roles values(1,2);

insert into users\_roles values(2,2);

insert into users\_roles values(3,3);

**Login credentials**

***Admin***

Username: admin

Password: admin

***Doctor***

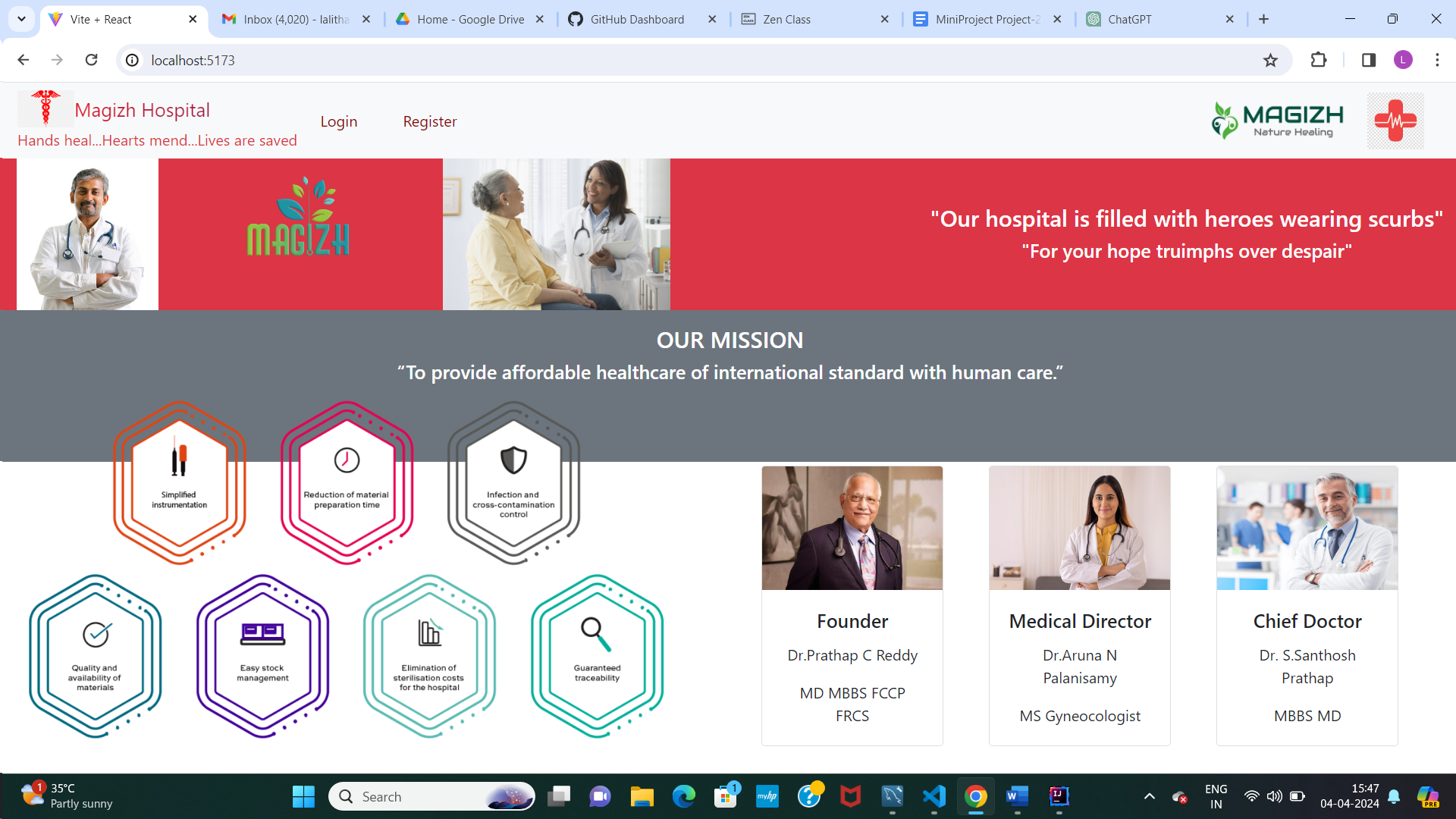
Username: doctor

Password: doctor

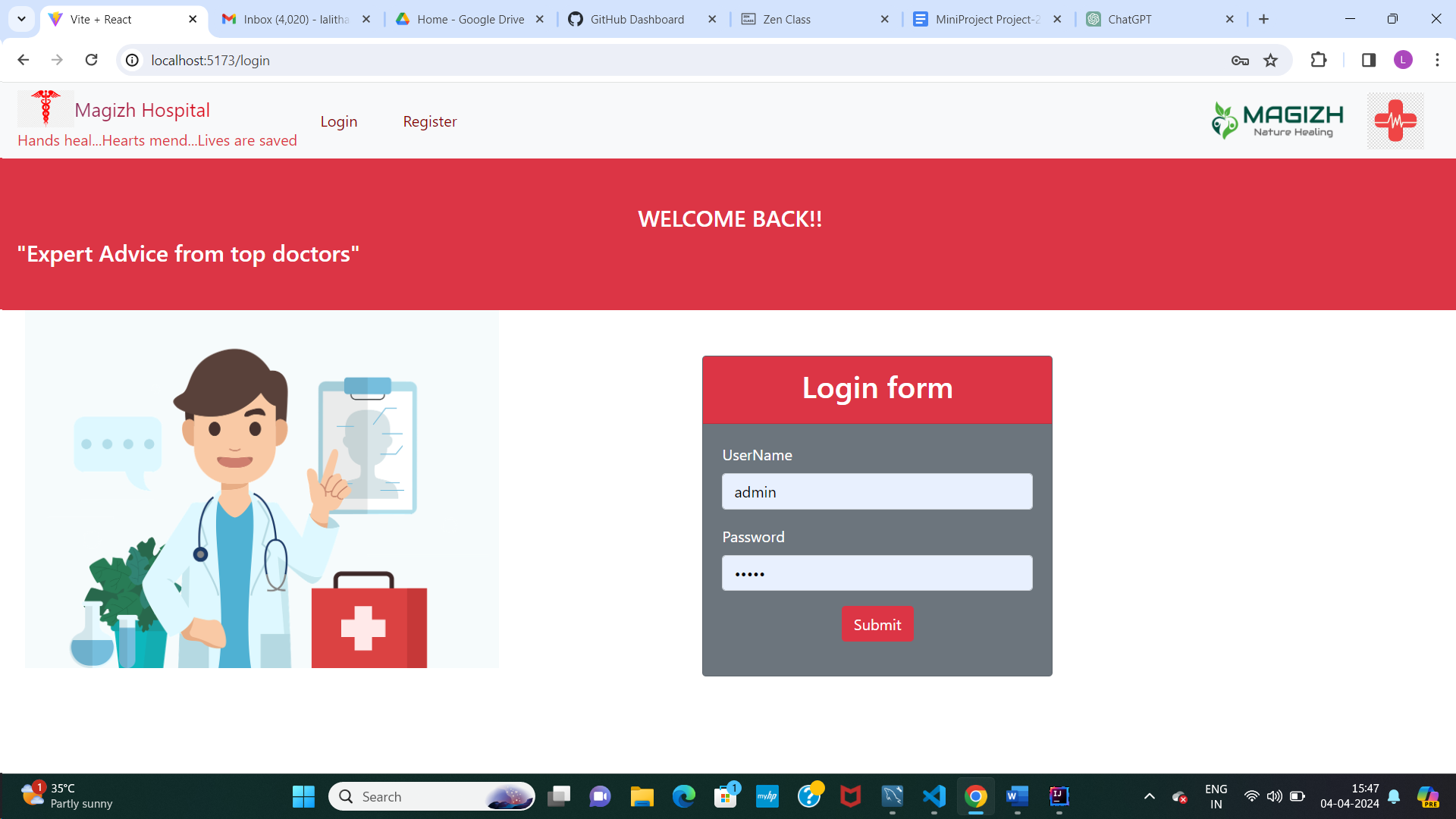
**You can add user by registering the user to login but admin and doctor credentials cannot be changed.**

**ScreenShots of my Magizh Hospital Management System**

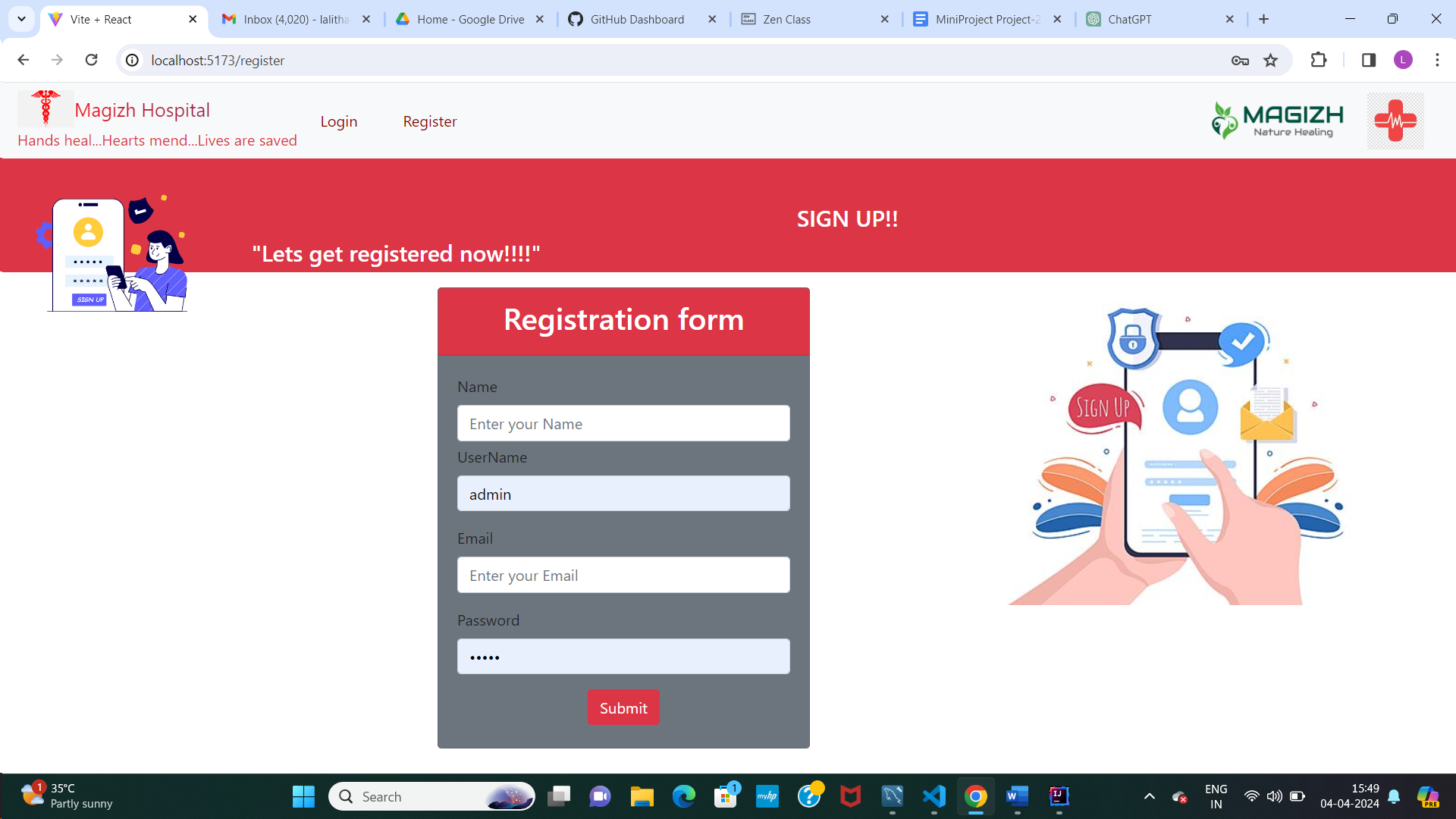
***Homepage(“/”)***



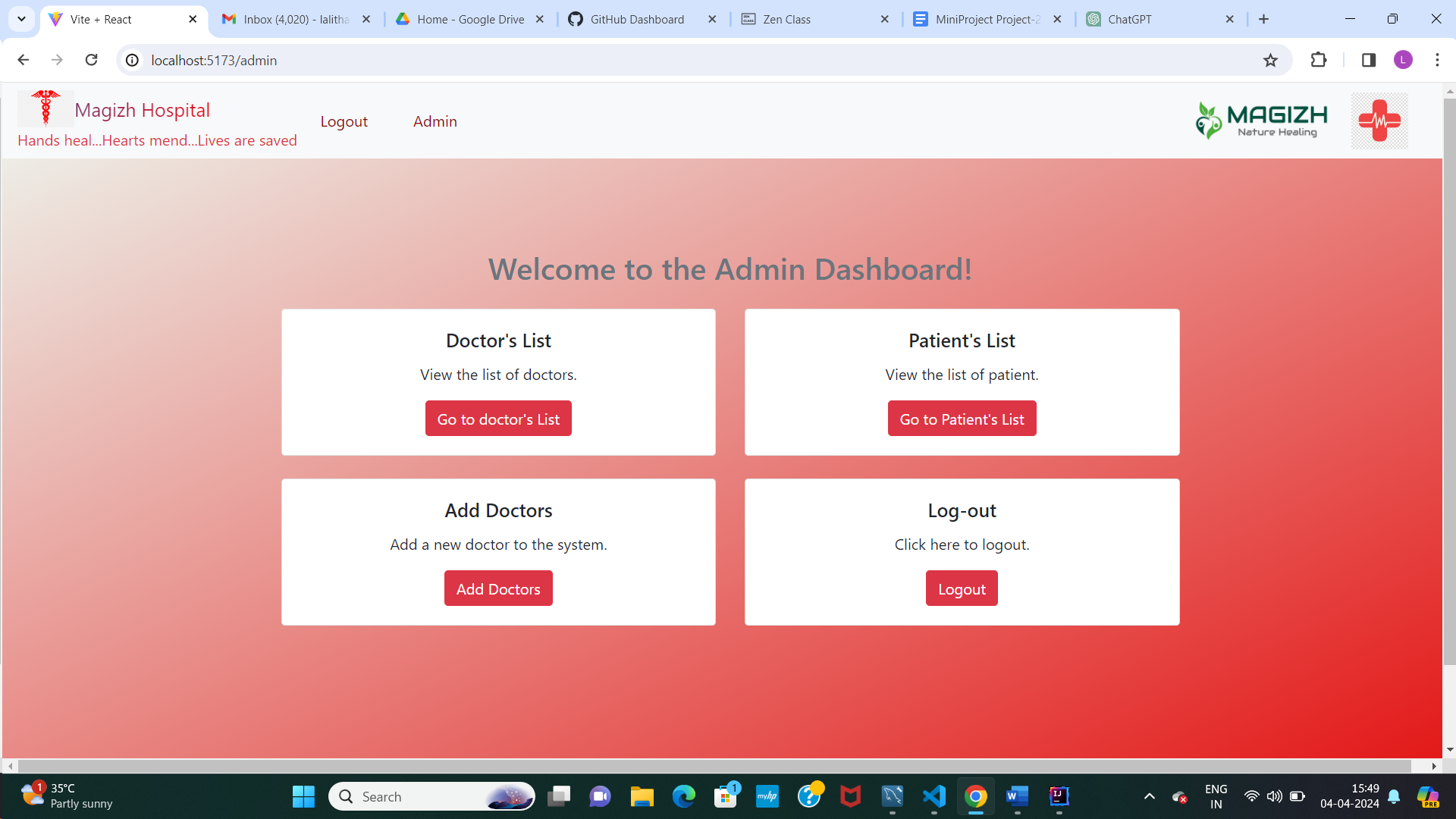
***Login page***



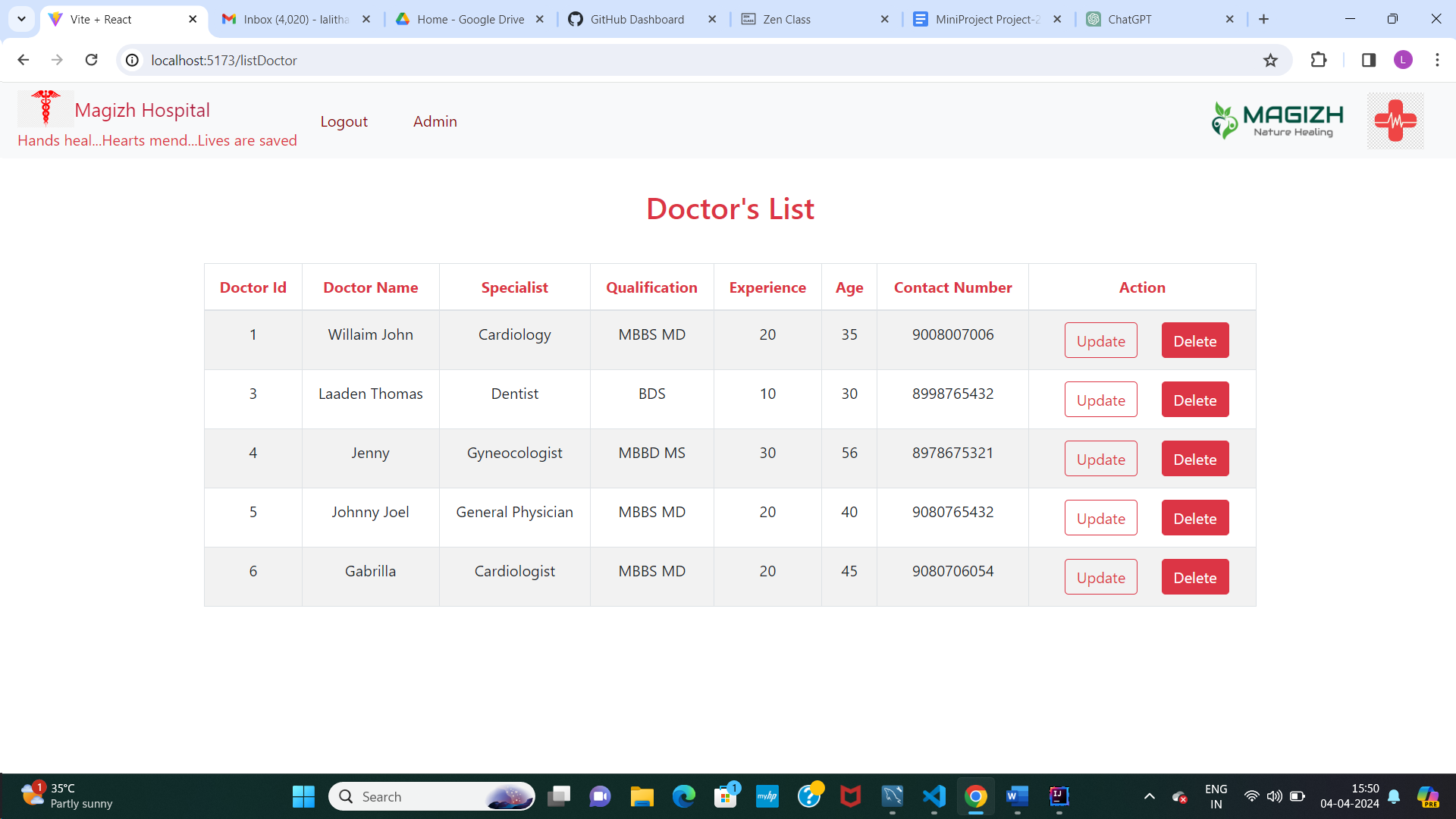
***Register Page***



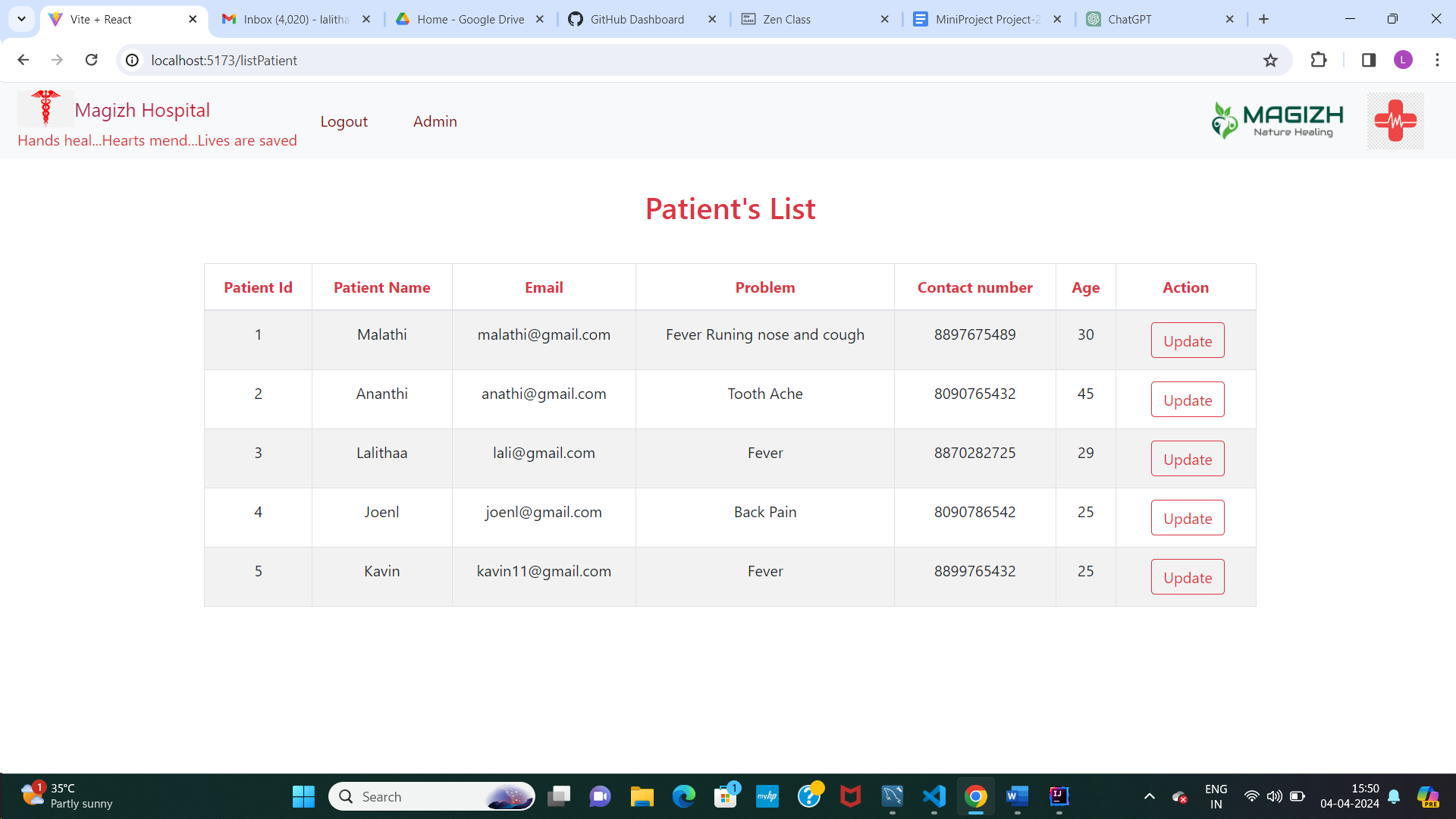
***Admin page***



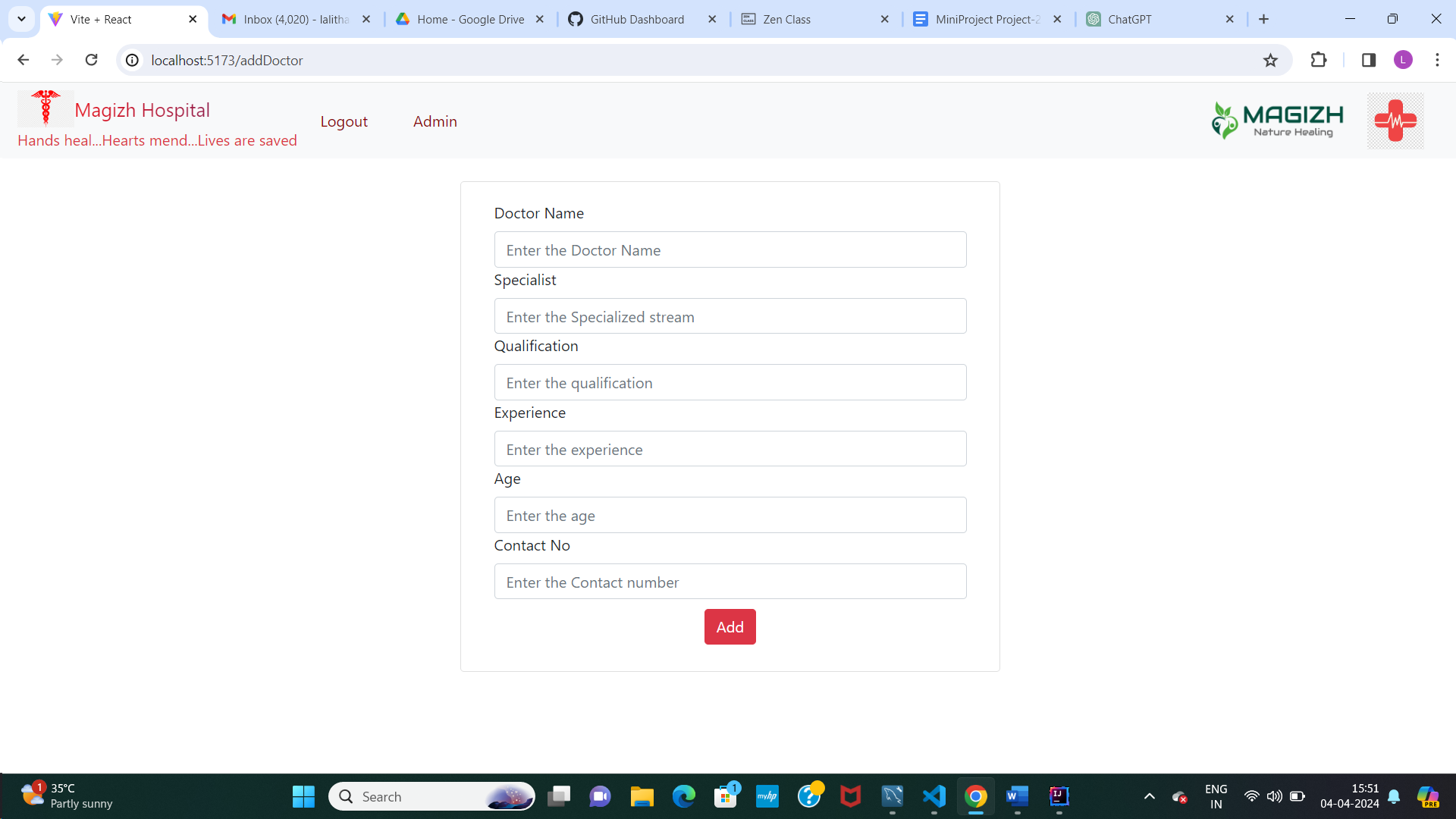
***Doctors list***



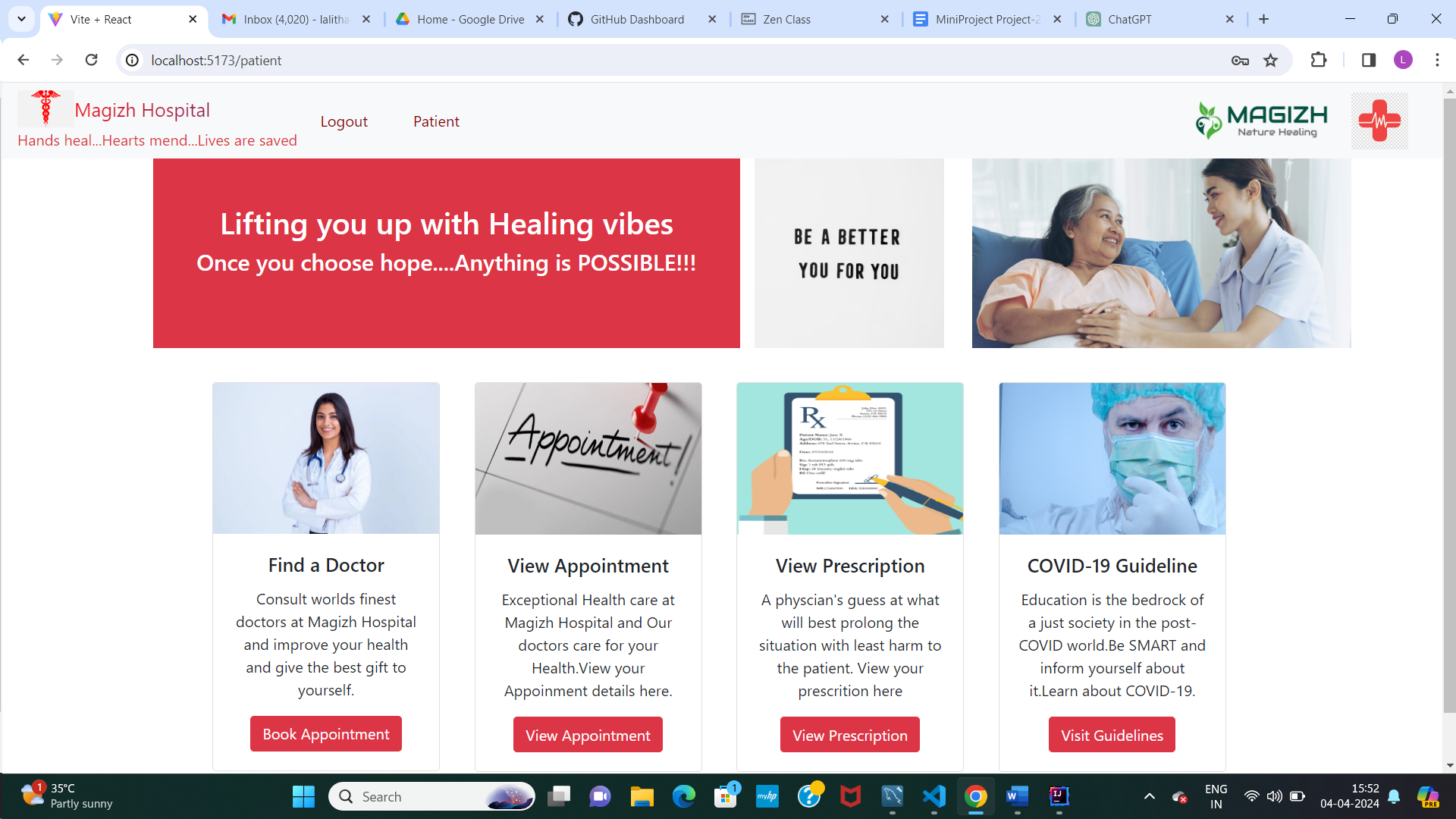
***Patient list***



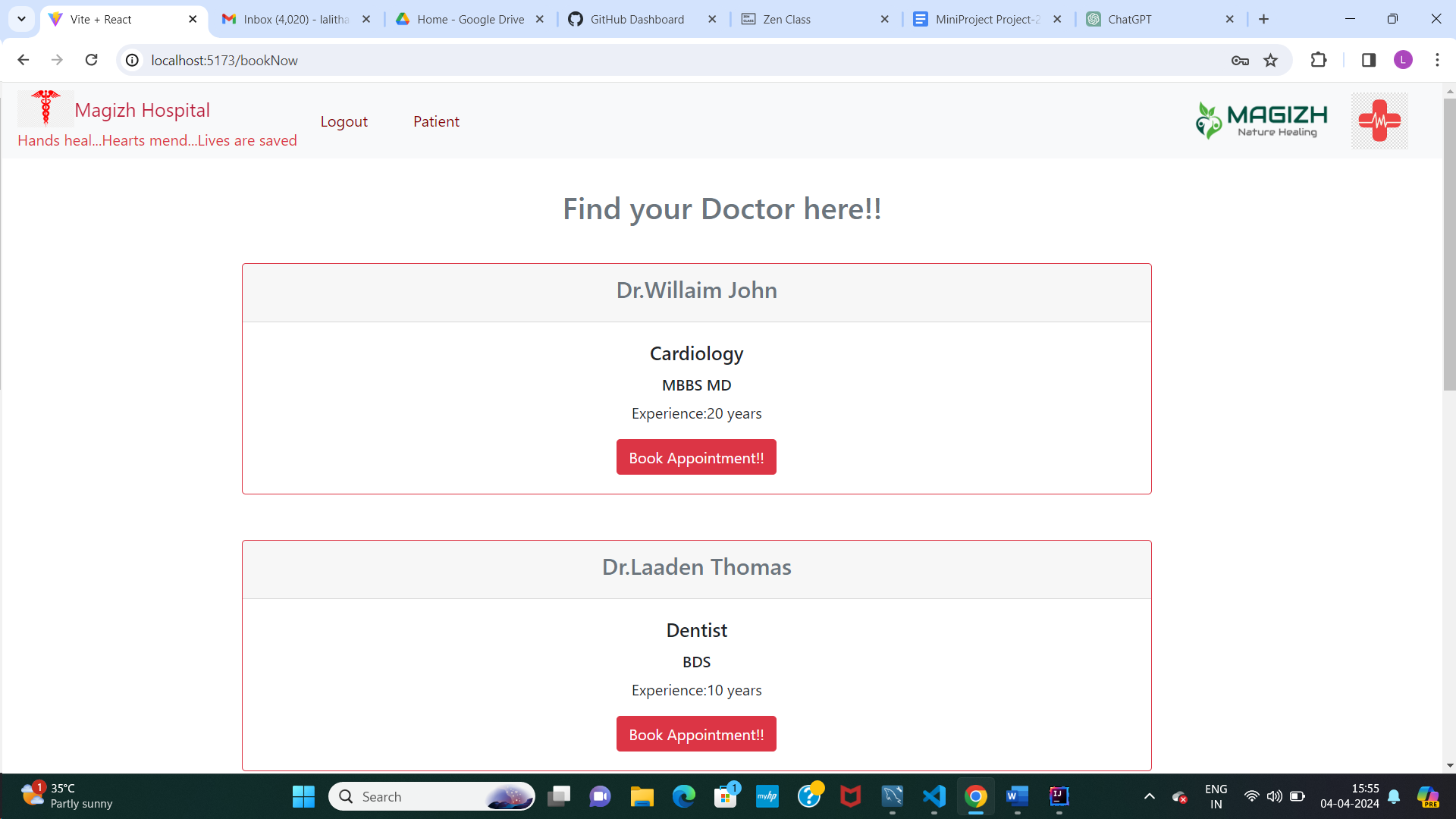
***Add doctor***



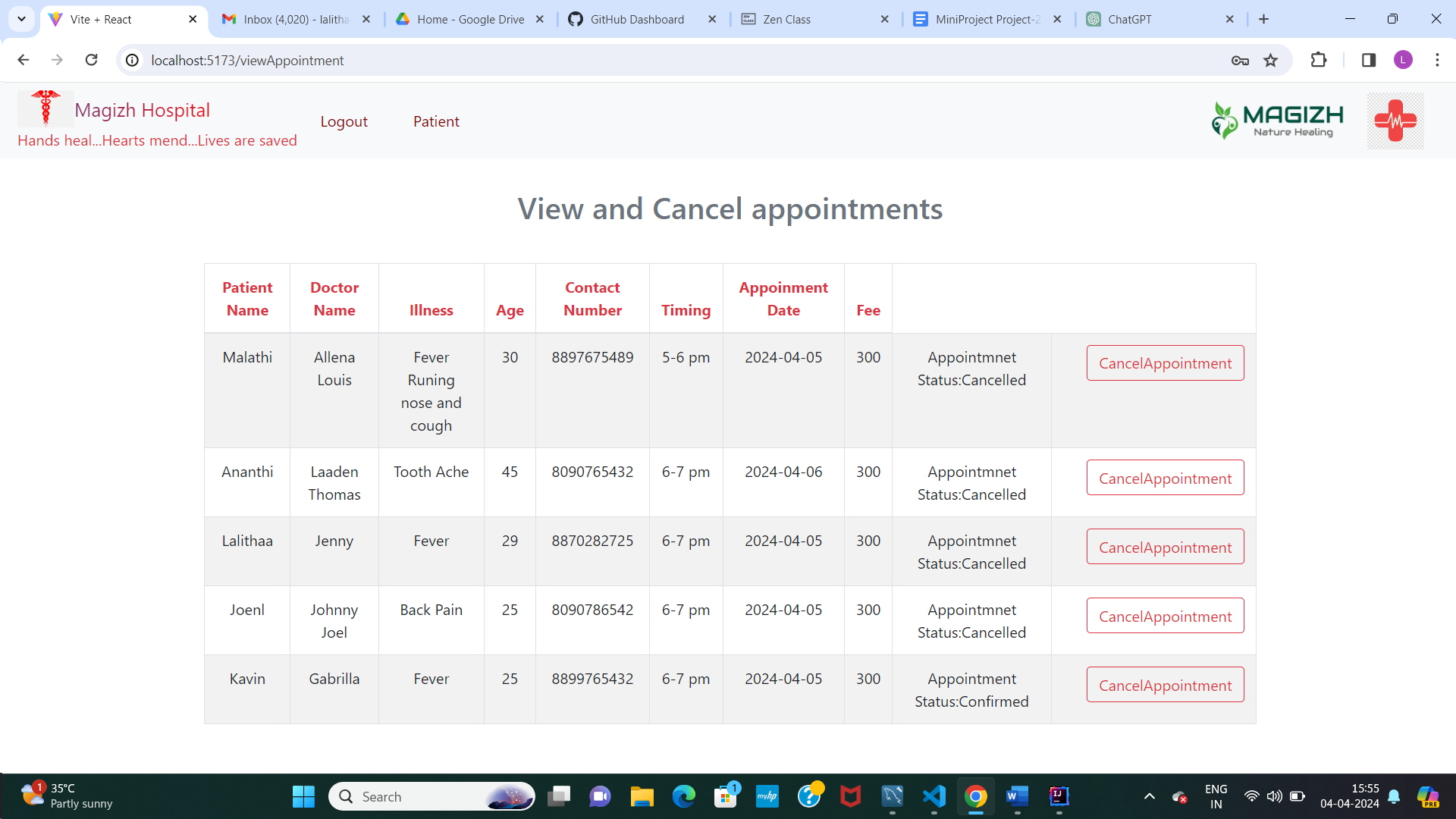
***Patient dashboard***



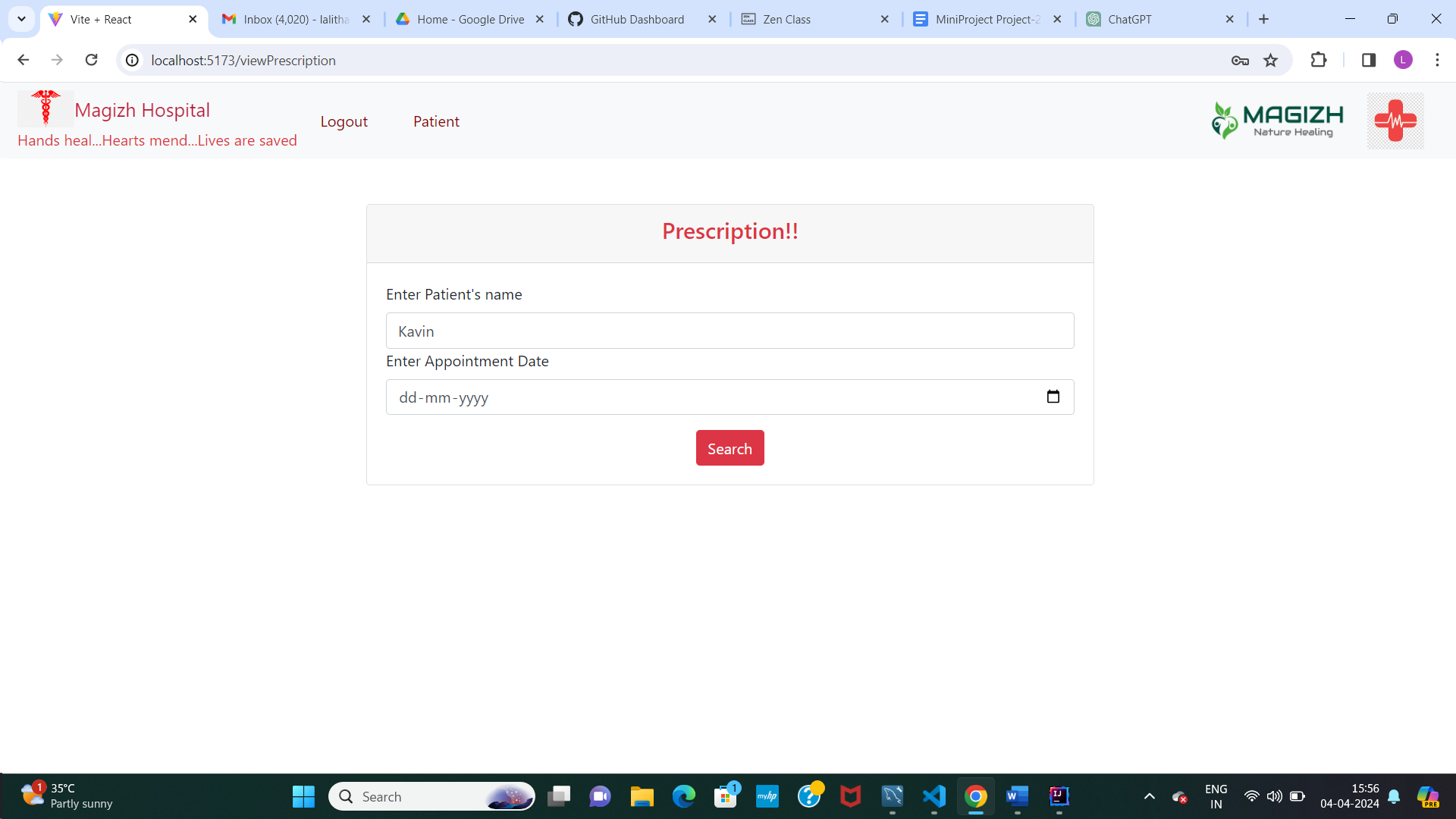
***Book appointment***

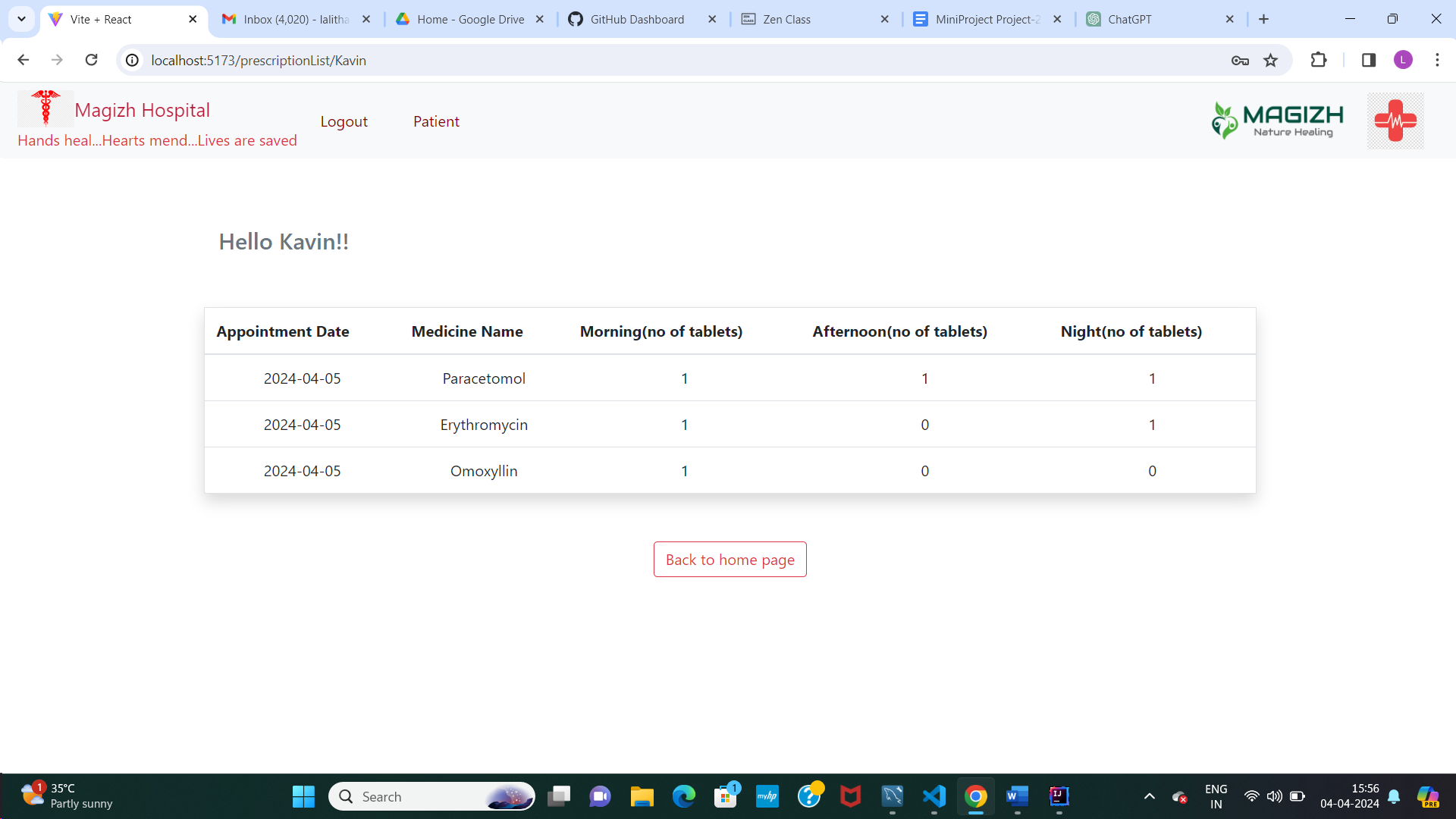


***View appointment***

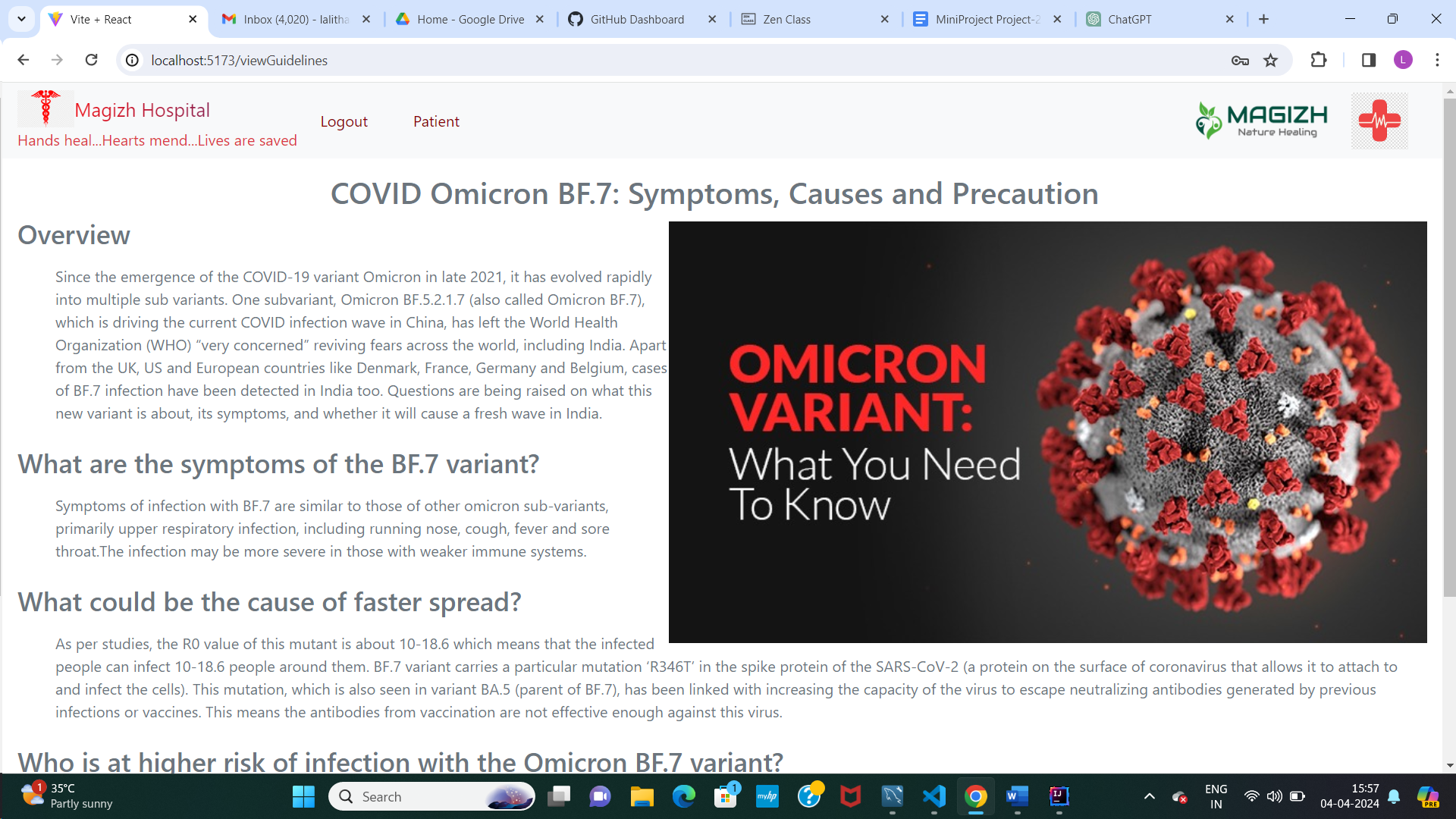


***View prescription***

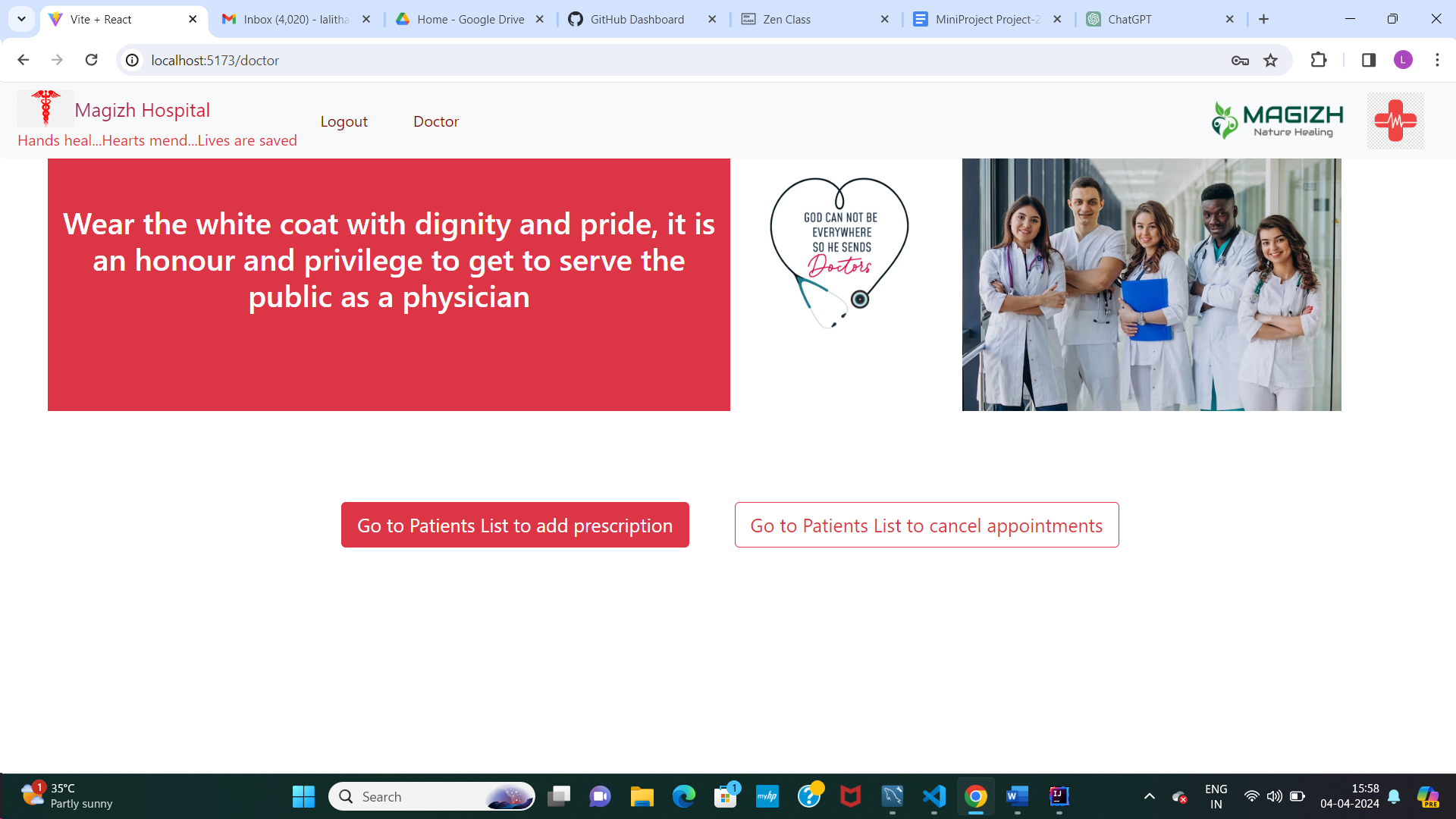




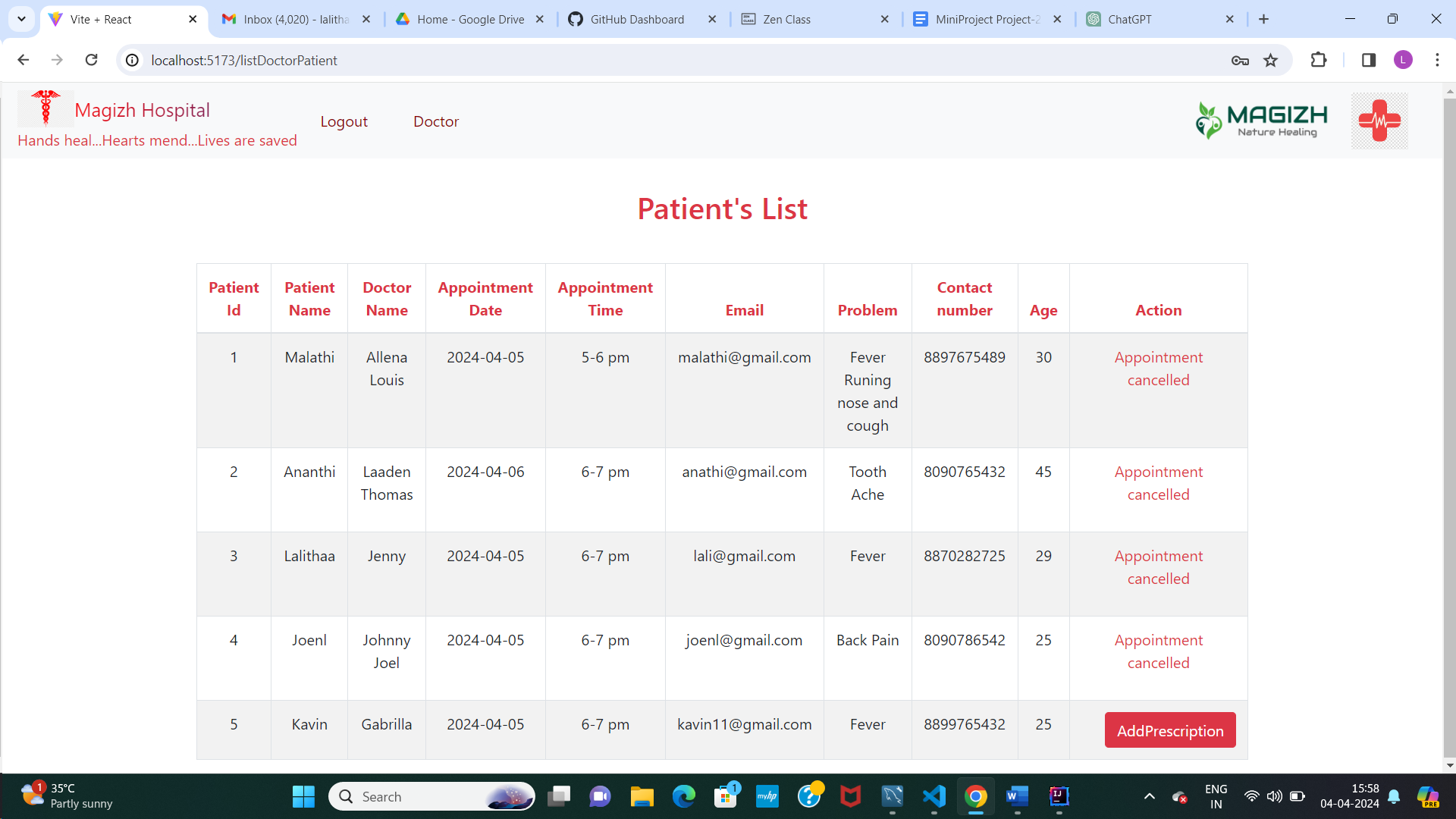
***Visit guideline***

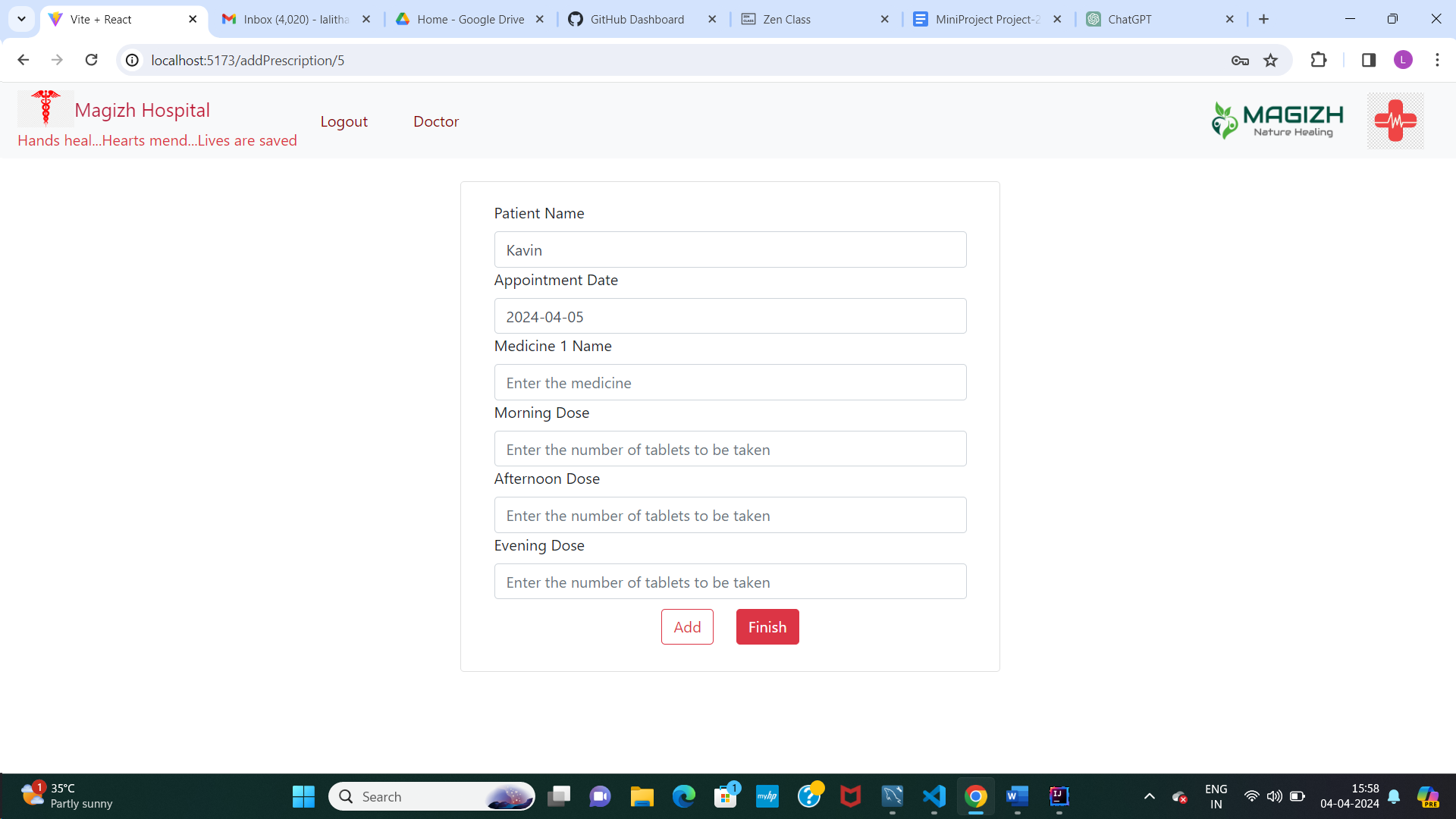


***Doctor dashboard***

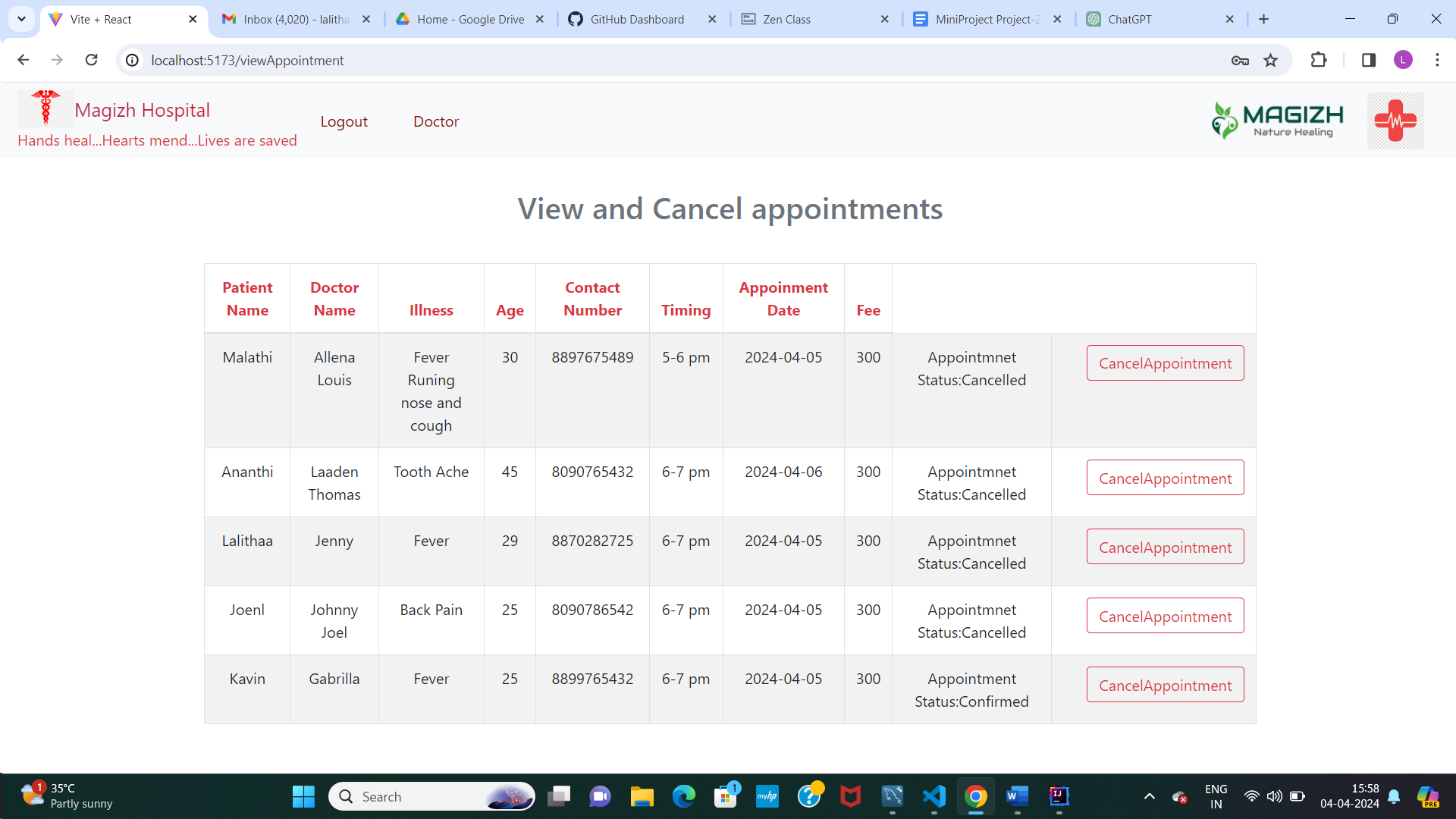


***Add prescription***





***Cancel appointment***



***Form validation***

