# Case Study On

# MediBook – Doctor Appointment Booking System

# INTRODUCTION

- MediBook is a web-based doctor appointment booking system that automates appointment scheduling.
- It allows patients to book, view, and cancel appointments online.
- The system reduces long waiting times and manual hospital processes.
- It minimizes administrative workload and improves appointment management.
- MediBook provides an efficient and user-friendly solution for patients and doctors.

# ABSTRAT

The **MediBook – Doctor Appointment Booking System** is developed to provide a seamless and user-friendly platform for managing doctor appointments online. Patients can register, log in, view available doctors, book appointments, view their booked appointments, and cancel appointments when required. Doctors can register, log in, and view the list of patients who have booked appointments with them, including appointment date and time. The application is developed using Spring Boot, Spring Data JPA (Hibernate), MySQL, Postman, and HTML/CSS/JavaScript for frontend interaction.This system ensures secure, role-based access and real-time appointment handling using REST APIs.

# CLIENT REQUIREMENT

- A web-based Doctor Appointment Booking System named MediBook.
- A system that allows patients to register, log in, and manage their appointments online.
- A doctor module where doctors can register, log in, and view appointments booked by patients.
- An appointment management module that enables patients to book, view, and cancel appointments.
- A role-based dashboard system for patients and doctors with separate functionalities.
- A platform that displays available doctors along with their specialization details.
- A scheduling mechanism to avoid appointment conflicts and overlapping bookings.

# TECHNICAL FEATURES

- RESTful APIs using Spring Boot
- Layered Architecture (Controller, Service, Repository)
- Session-based login handling (Session Storage)
- JSON-based data exchange
- CRUD operations
- Exception handling
- Role-based dashboard navigation

# TECHNOLOGIES AND TOOLS USED

**Backend**
Java 17
Spring Boot 3
Spring Data JPA (Hibernate)
Maven
REST APIs
Apache Tomcat (Embedded)

**Database**
MySQL 8.0

**Frontend**
HTML
CSS
JavaScript

**Tools**
Spring Tool Suite (STS) 4
Postman
MySQL Workbench
Google Chrome

# SYSTEM REQUIREMENTS

**Software Requirements**

Operating System: Windows 10+

Java JDK 17

MySQL Server 8.0

Spring Tool Suite (STS)

Web Browser (Chrome)

**Hardware Requirements**

Processor: Intel i3 or above

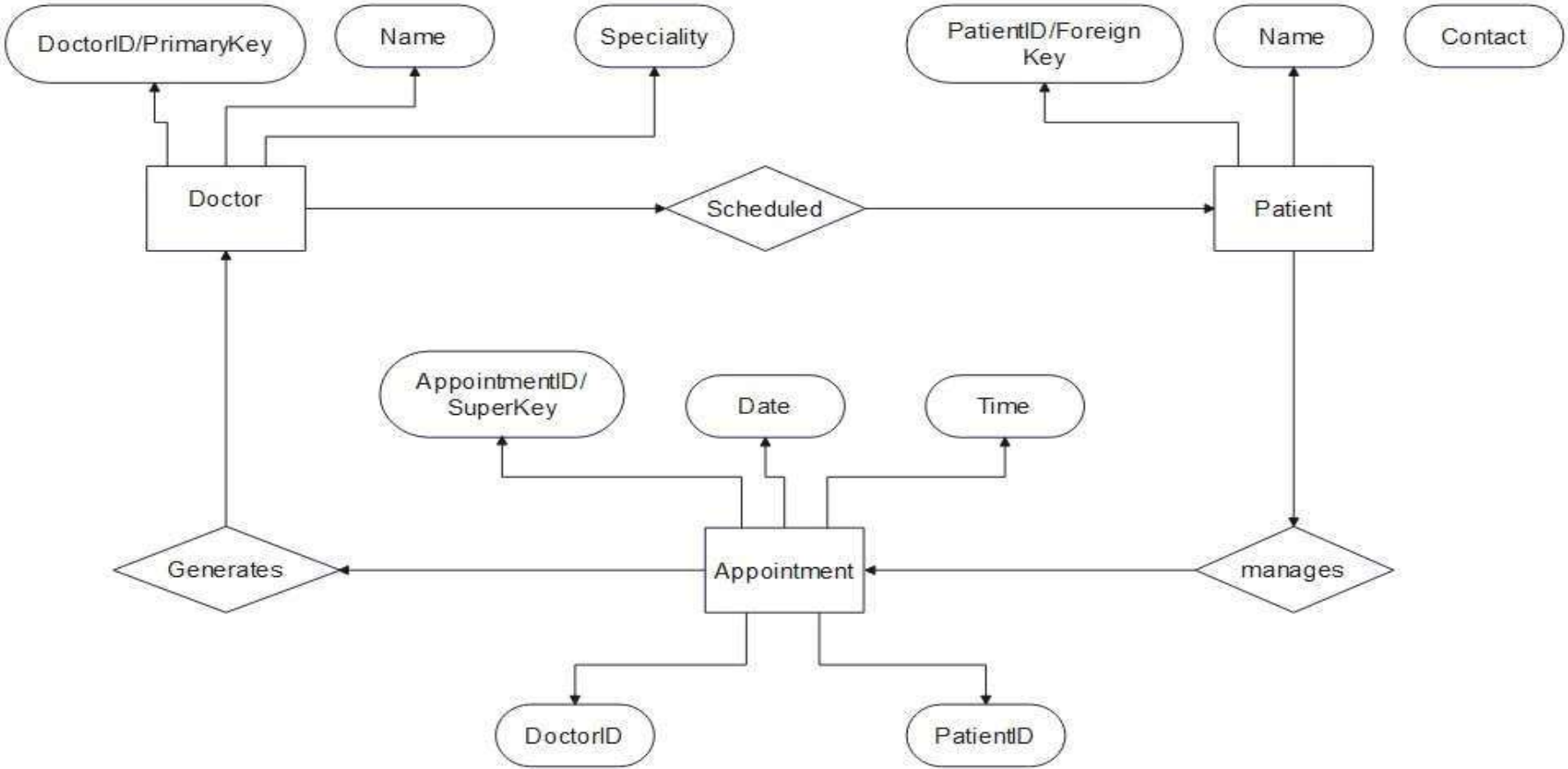
RAM: Minimum 4 GB

Hard Disk: 10 GB free space

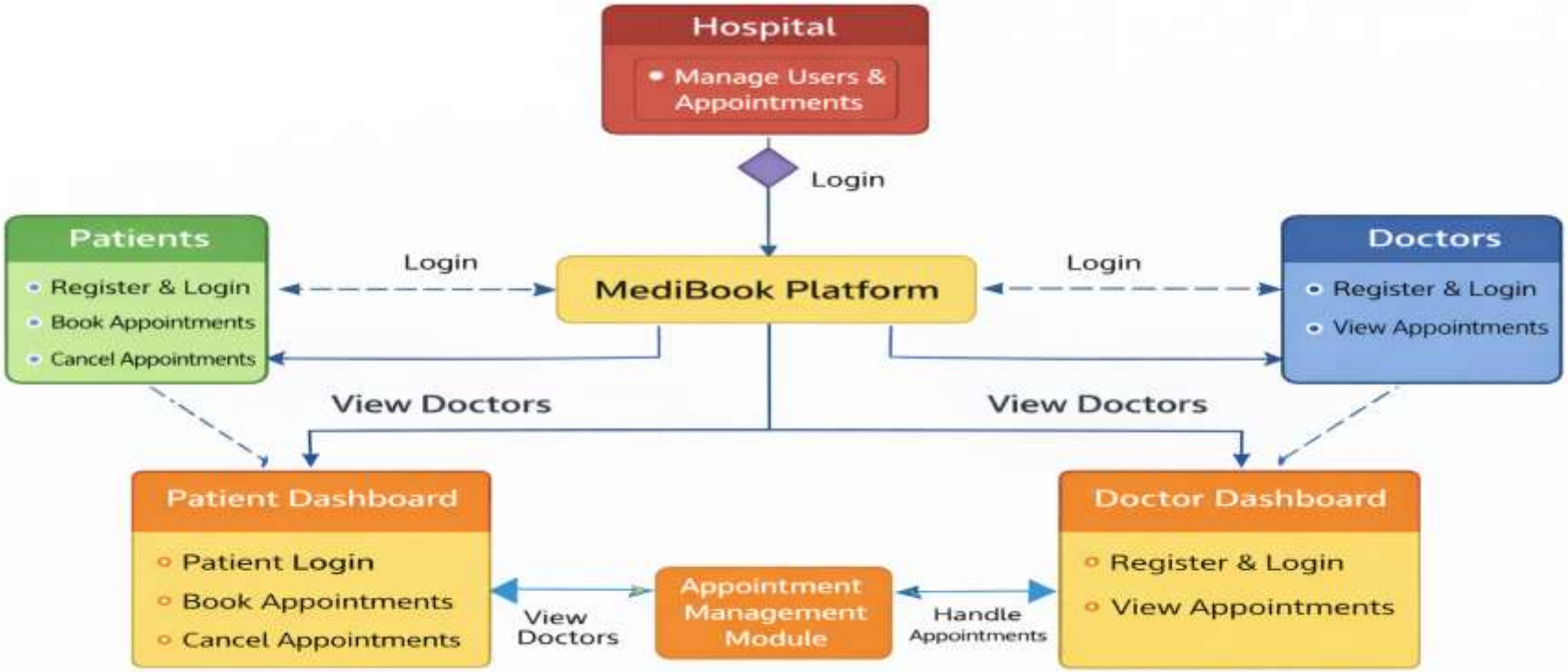
# PROJECT MODULE

- Patient Module
- Doctor Module
- Appointment Module
- Home / Navigation Module

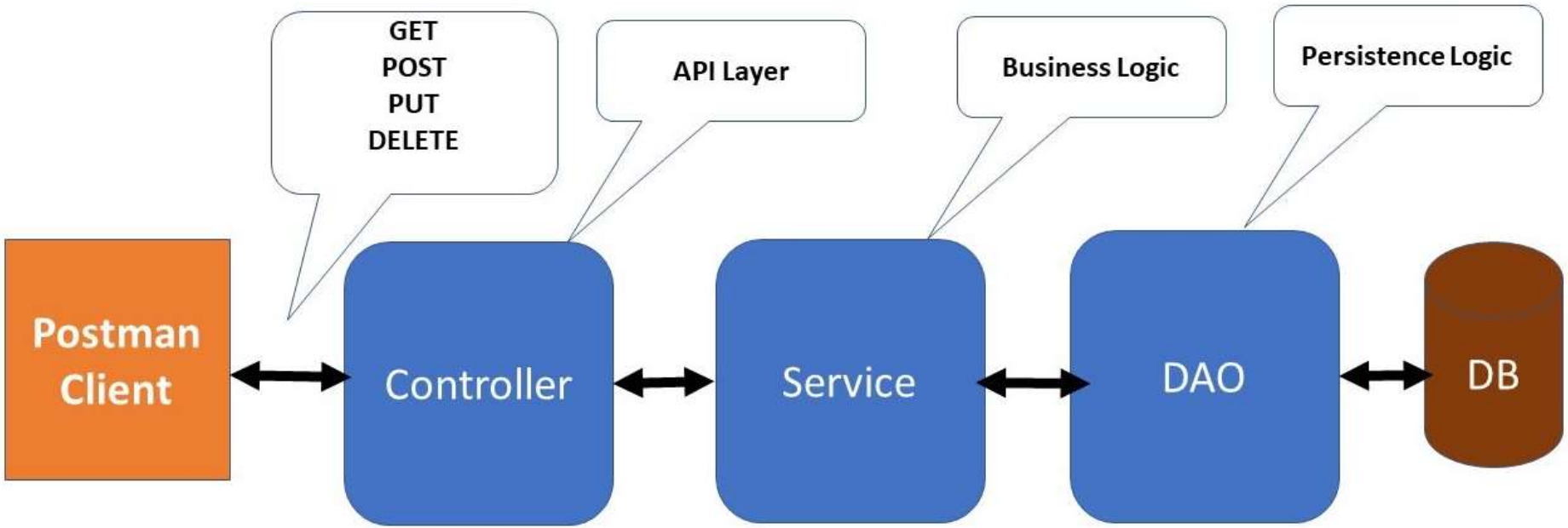
# ER DIAGRAM


DoctorID/PrimaryKey
Name
Speciality
PatientID/Foreign Key
Name
Contact
Doctor
Scheduled
Patient
AppointmentID/ SuperKey
Date
Time
Generates
Appointment
manages
DoctorID
PatientID

# Client – Server Architecture



# Spring Boot APP



# HOME MODULE

Acts as the entry point of the MediBook system.

Provides navigation options for:

- Patient
- Doctor
- Hospital

Redirects users based on login status.

Improves usability by guiding users to the correct dashboard.

Ensures easy access to different system roles.

# HOSPITAL MODULE

- Allows hospital staff to view registered doctors.
- Displays doctor details such as name and specialization.
- Helps in managing overall appointment flow.
- Acts as a supervisory interface for hospital operations.
- Improves coordination between doctors and patients

# DOCTOR MODULE

- Allows doctors to register and log in to the system.
- Displays a personalized Doctor Dashboard.
- Enables doctors to view:
- List of patients who booked appointments
- Appointment date and time
- Helps doctors manage their daily schedules efficiently.
- Reduces manual tracking of patient appointments.

# PATIENT MODULE

- Allows patients to register and log in securely.
- Displays available doctors with specialization details.
- Enables patients to:
- Book appointments
- View booked appointments
- Cancel appointments if needed
- Prevents overlapping appointments.
- Provides a user-friendly interface for appointment management.

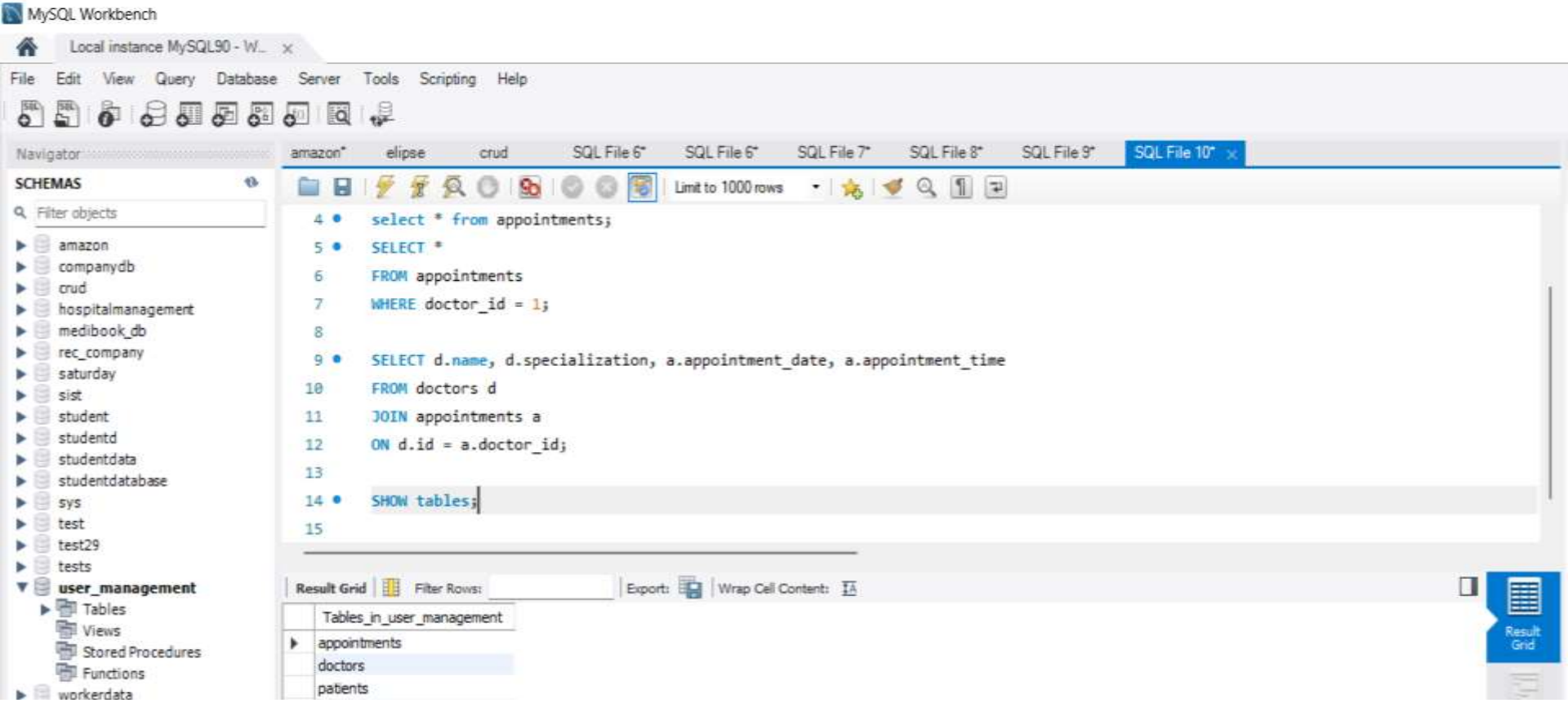
# Http Request Methods

HTTP request methods are used to perform different operations such as fetching data, creating records, and deleting records in the MediBook system.

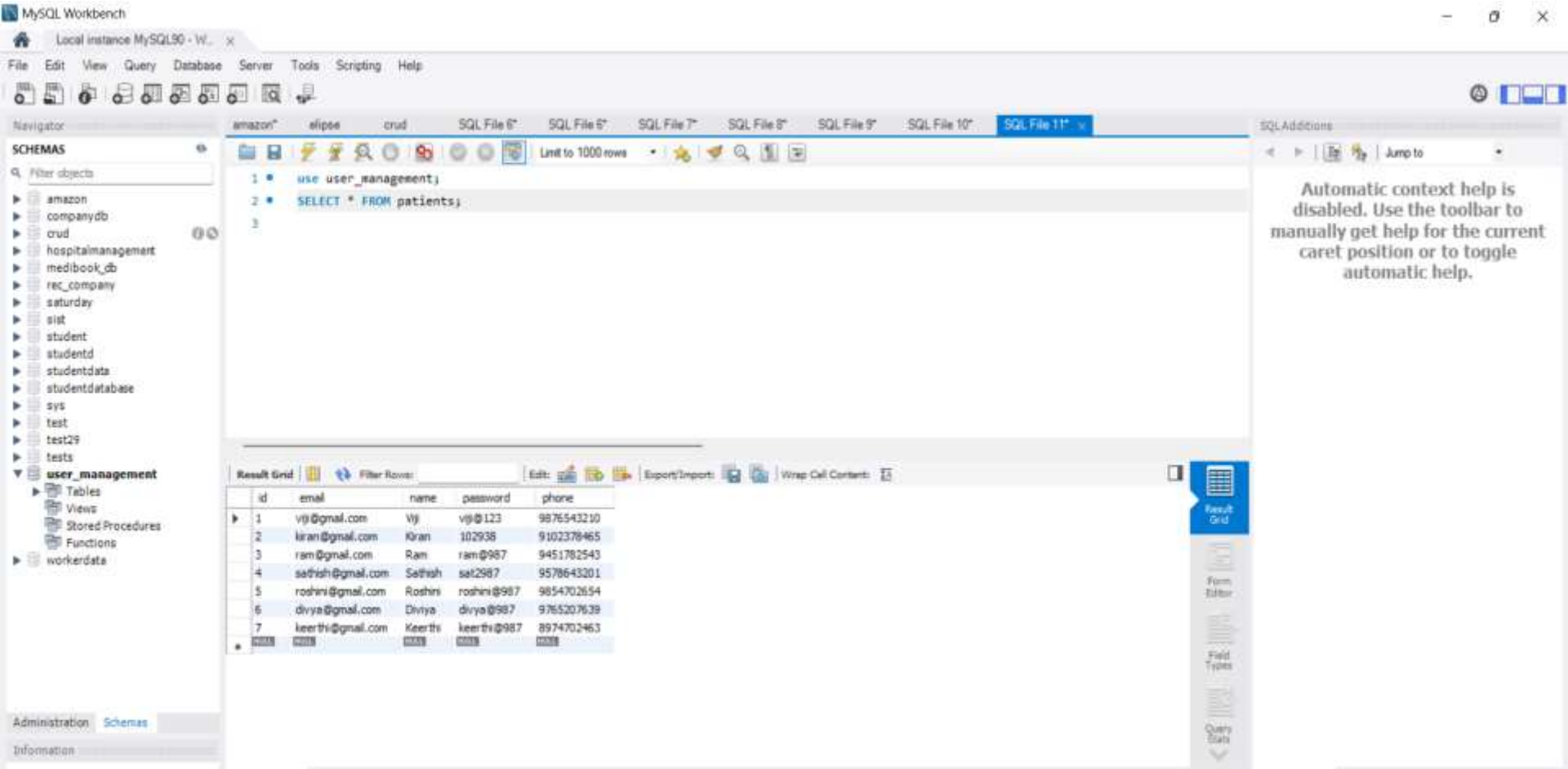
| | | |
|---|---|---|
| GET | http://localhost:8080/api/doctors | View doctors |
| POST | http://localhost:8080/api/appointments/book | Book appointment |
| GET | http://localhost:8080/api/appointments/patient | View patient appointments |
| GET | http://localhost:8080/api/appointments/doctor/id | View doctor appointments |
| DELETE | http://localhost:8080/api/appointments/id | Cancel appointment |

# DATA DICTIONARY

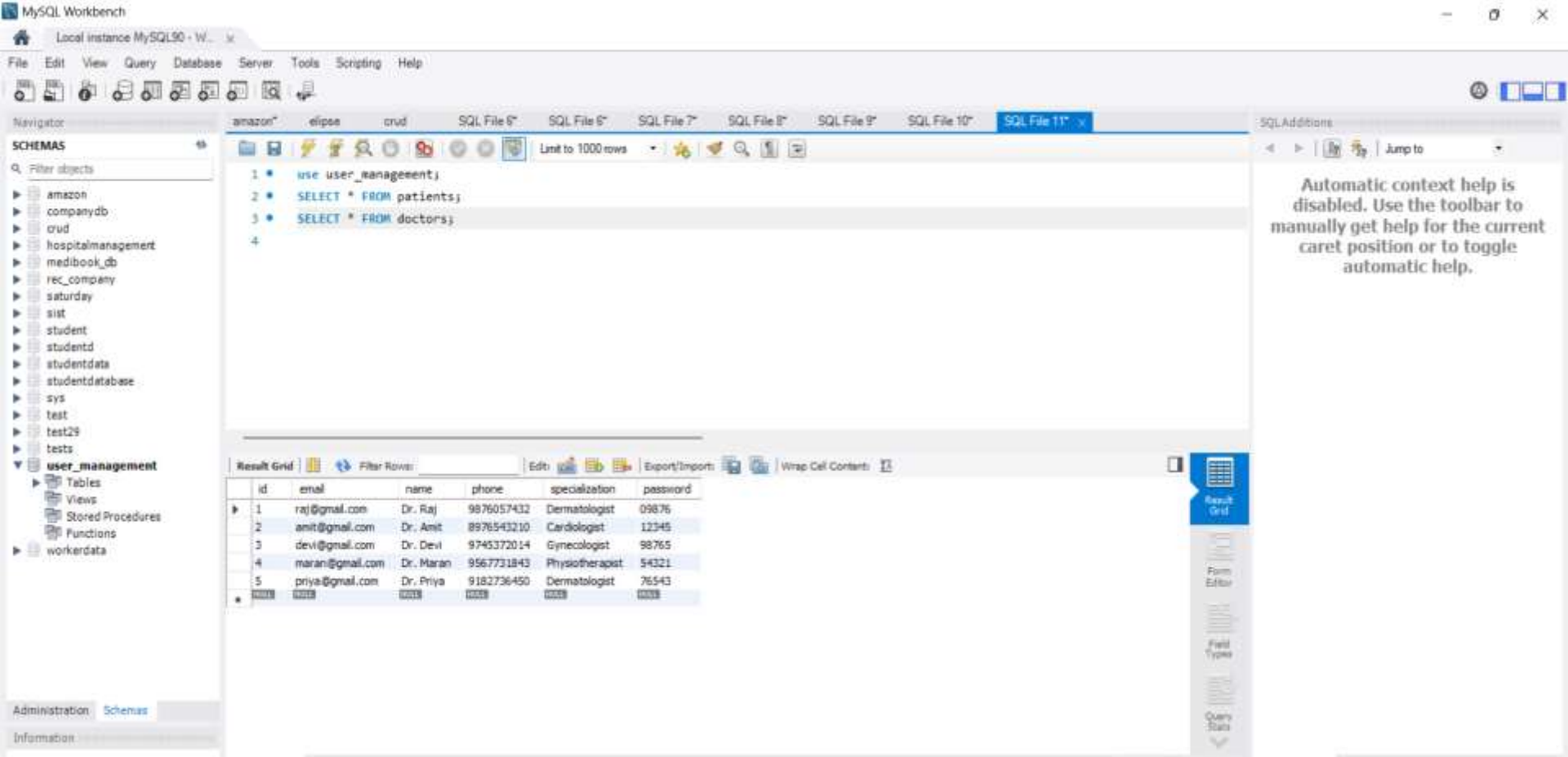
# TABLES OF DATABASE



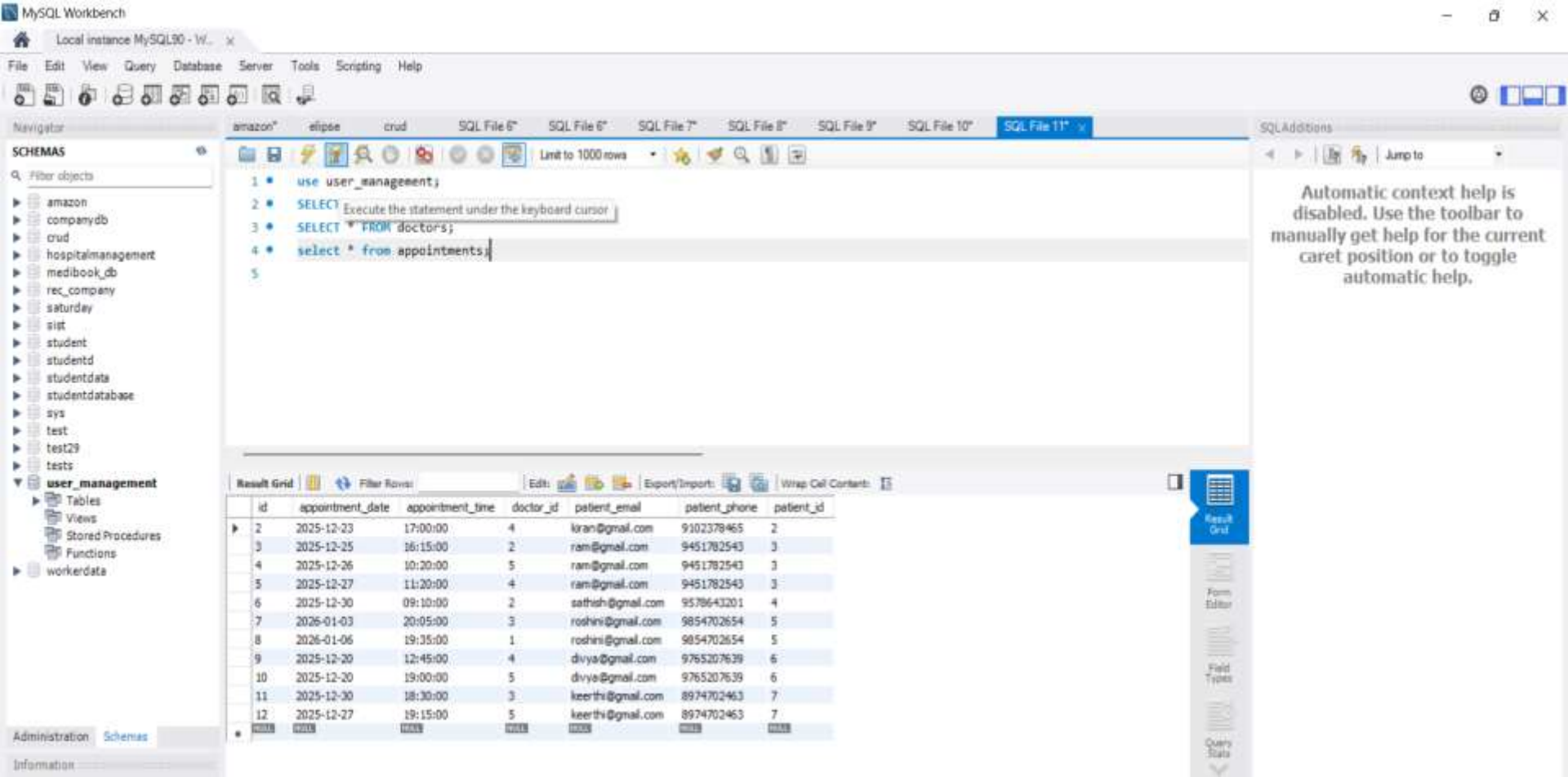
# PATIENT DATABASE



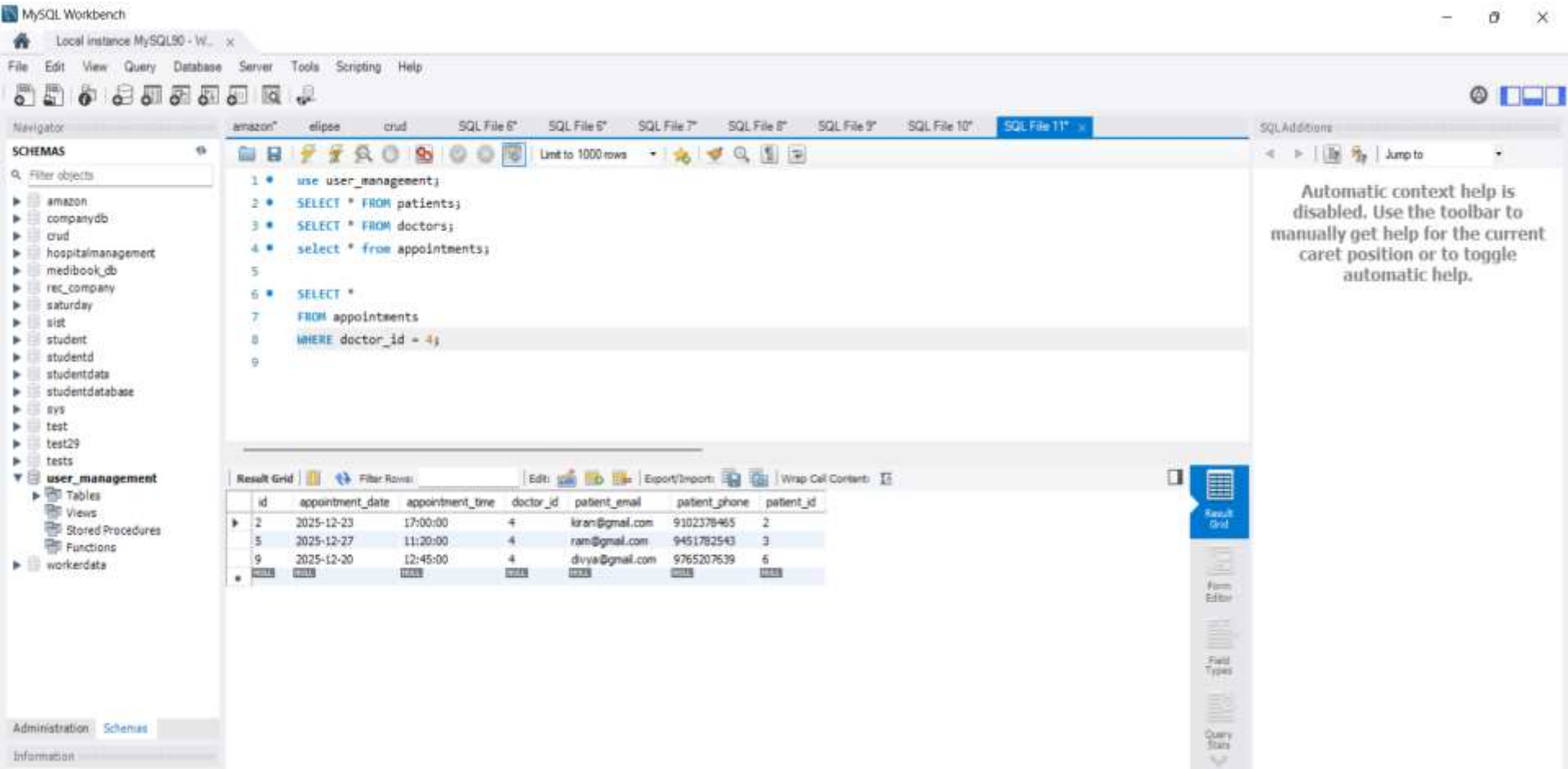
# DOCTOR DATABASE



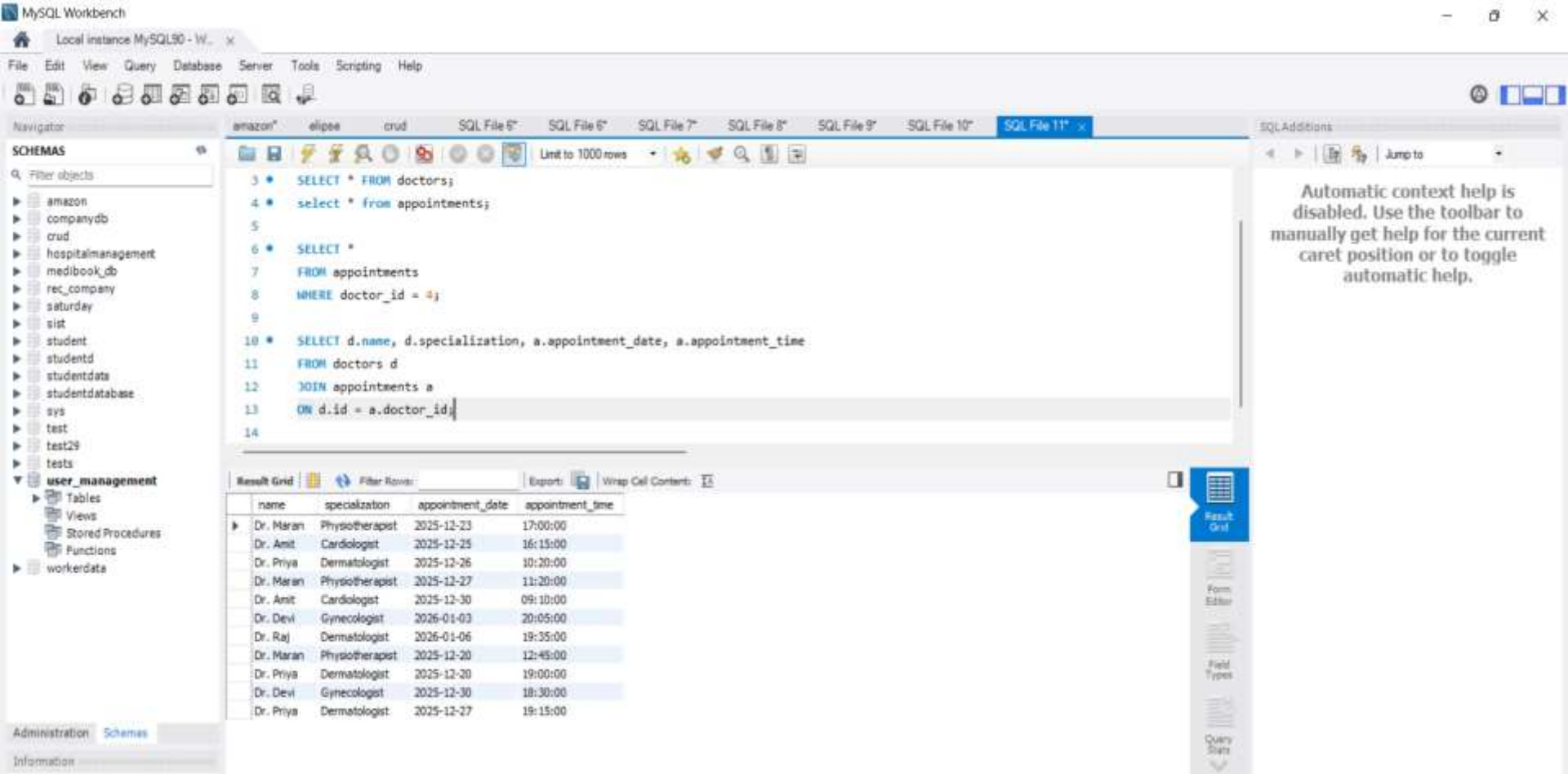
# APPOINTMENT DATABASE



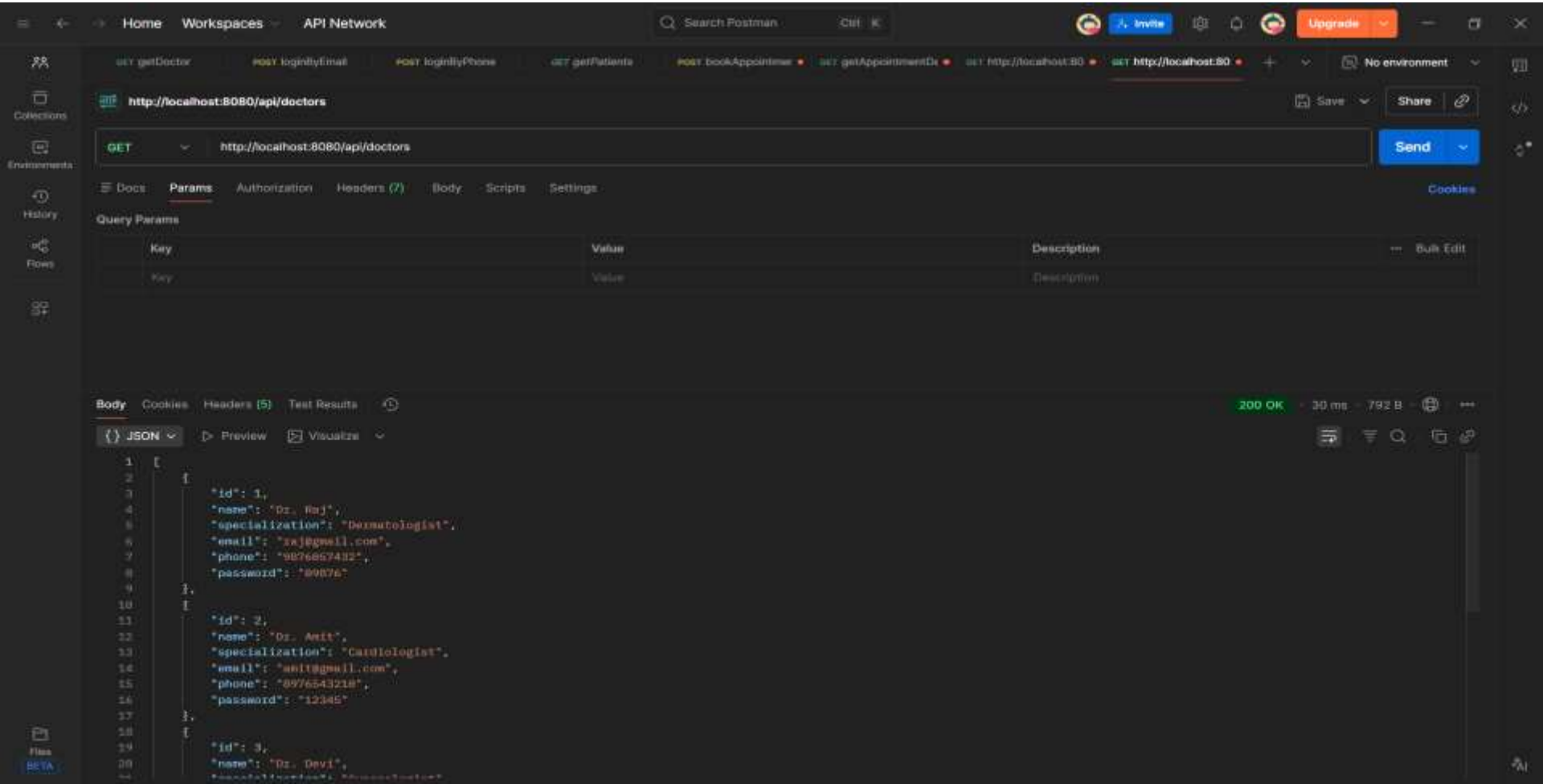
# Appointments for a Specific Doctor



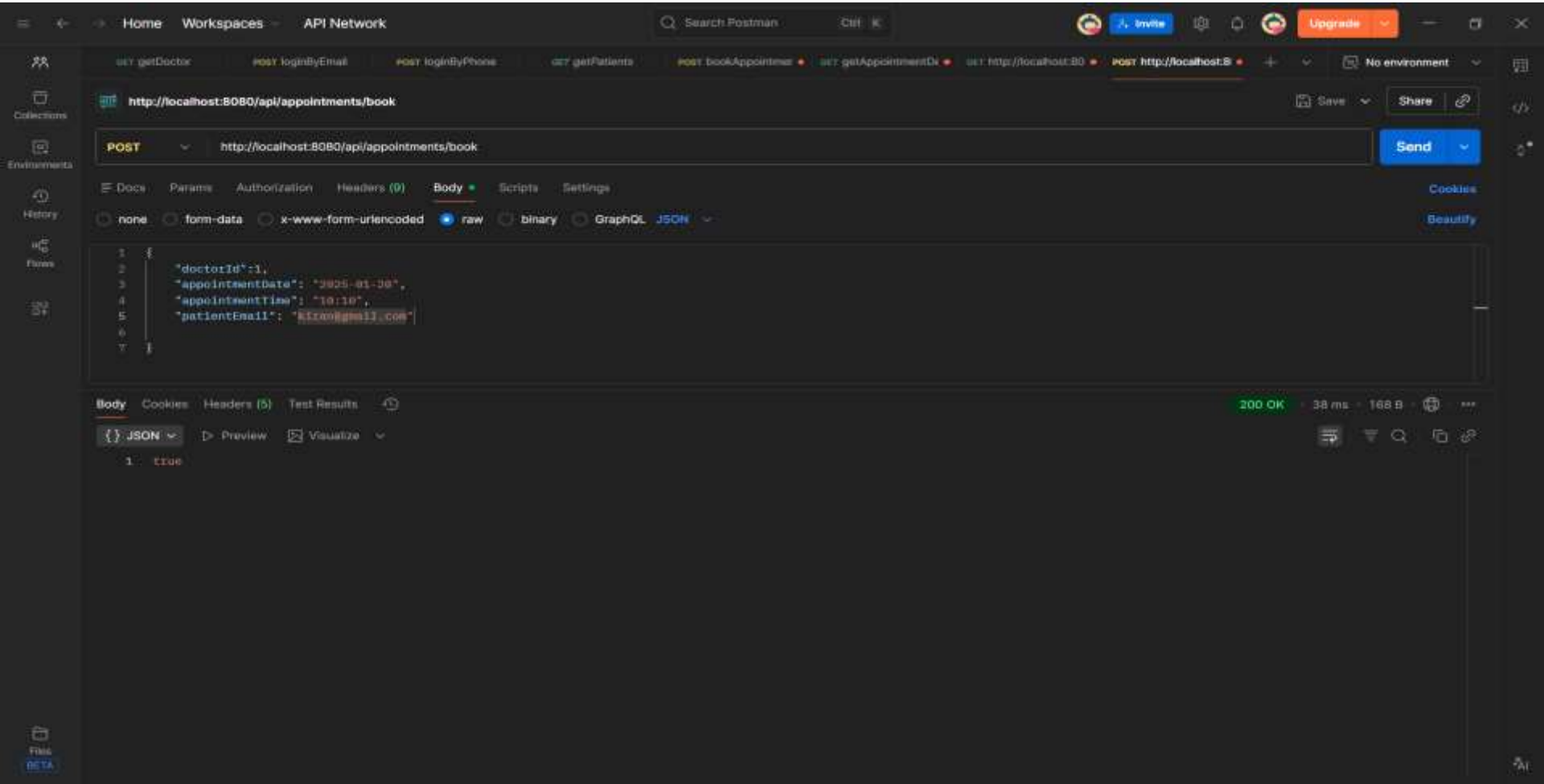
# HOSPITAL DATABASE



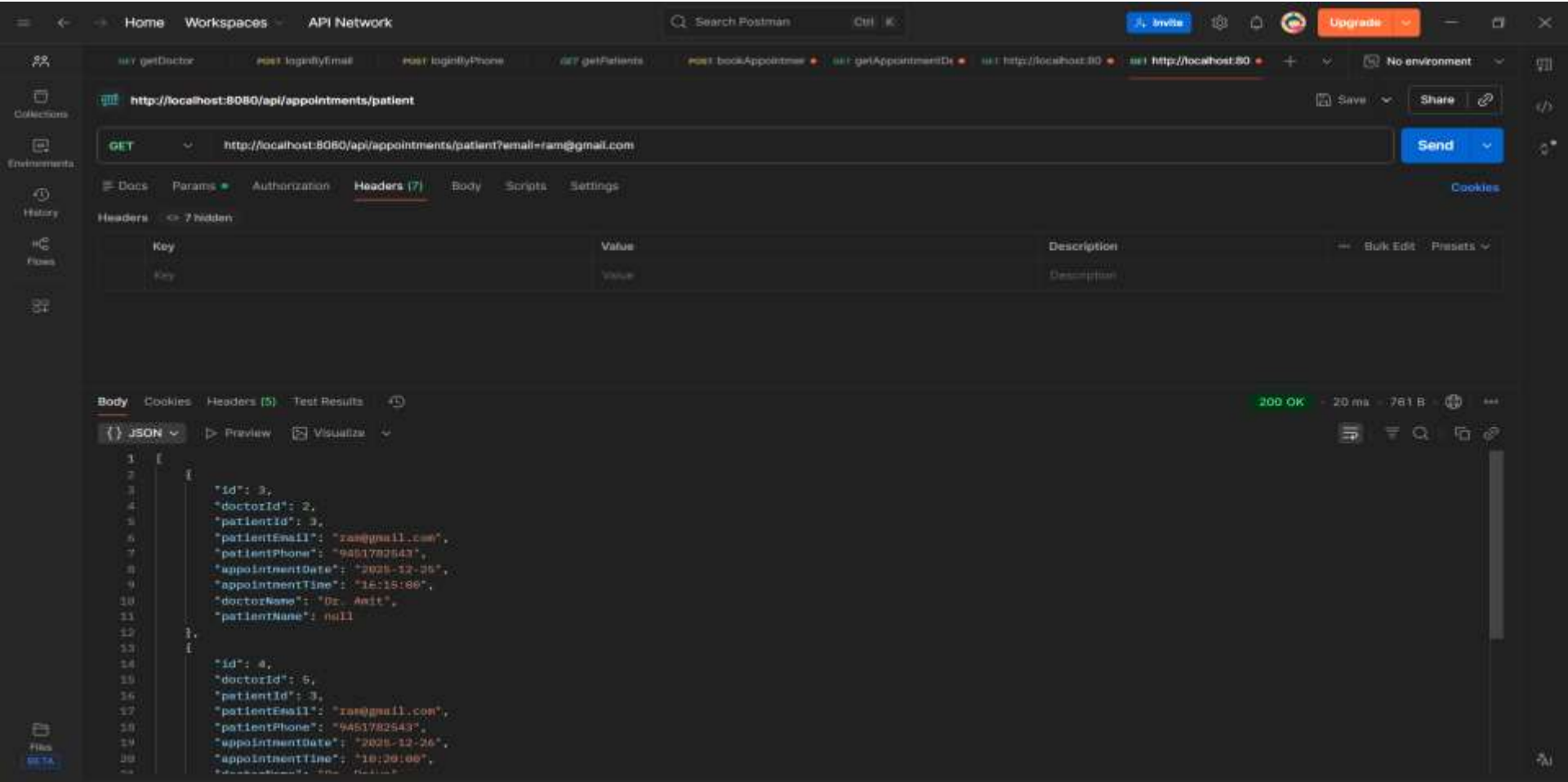
# GET METHOD FOR VIEWING DOCTORS



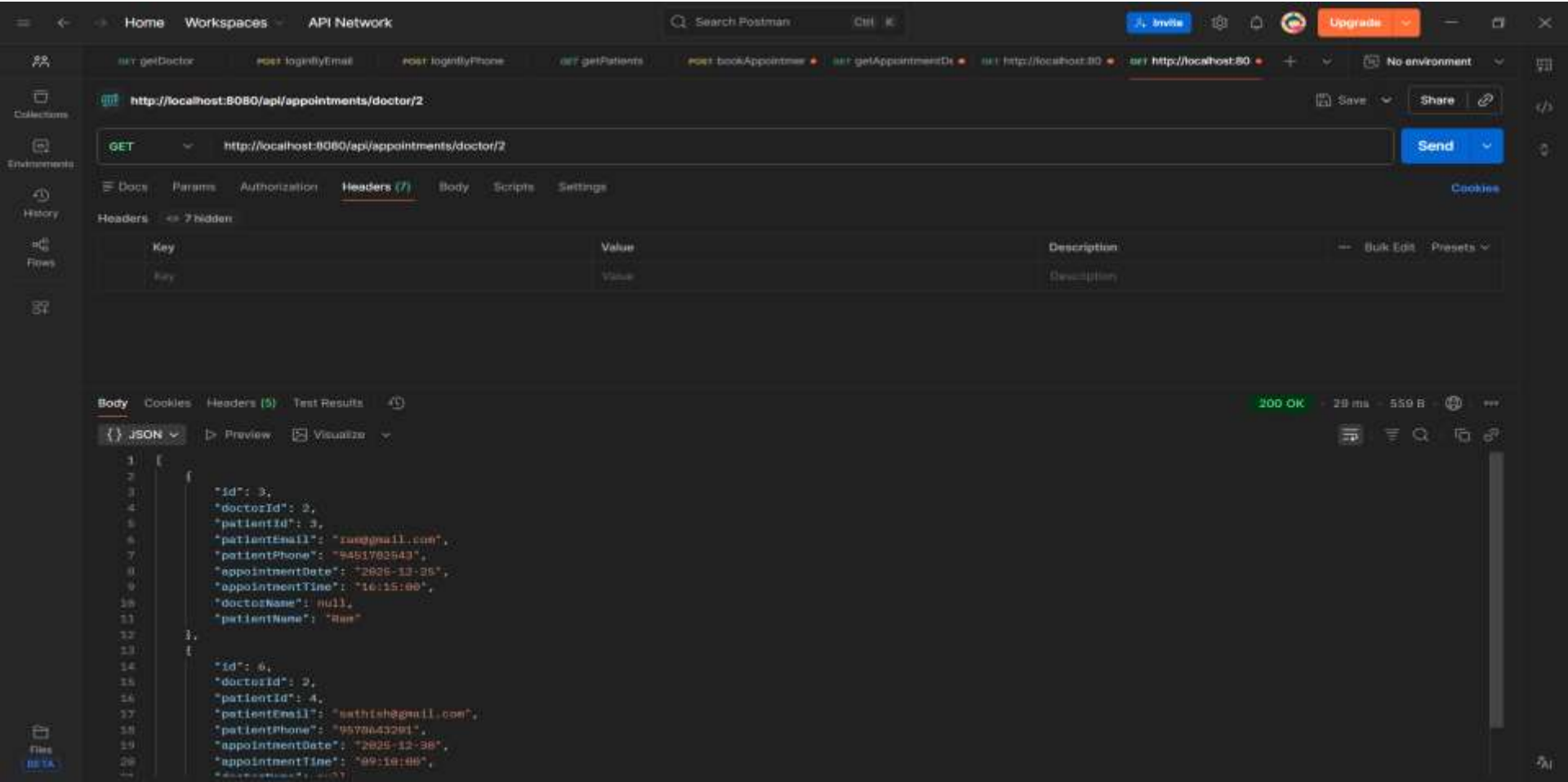
# POST METHOD FOR BOOKING APPOINTMENT



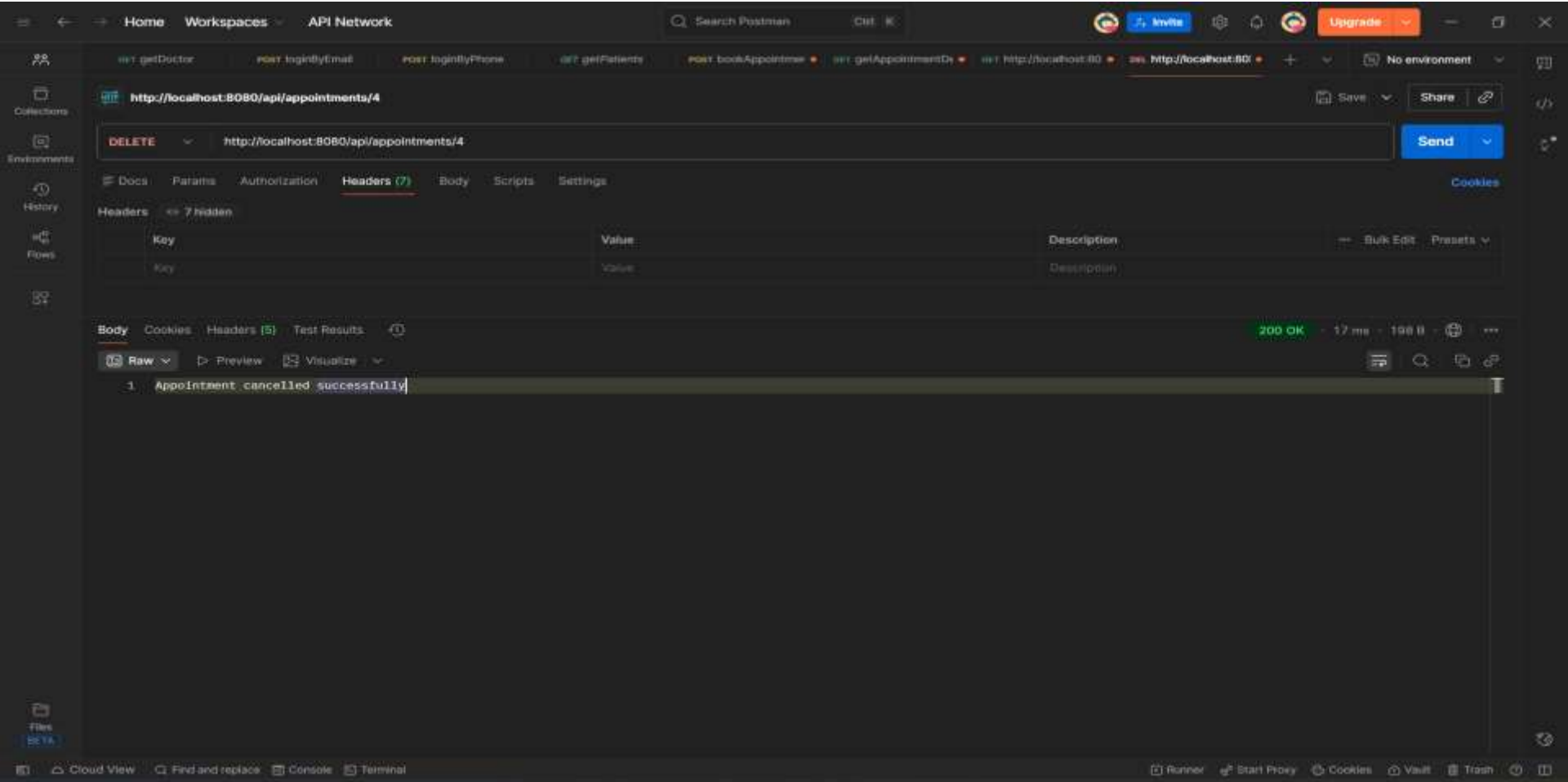
# GET METHOD FOR VIEWING PATIENT APPOINTMENTS



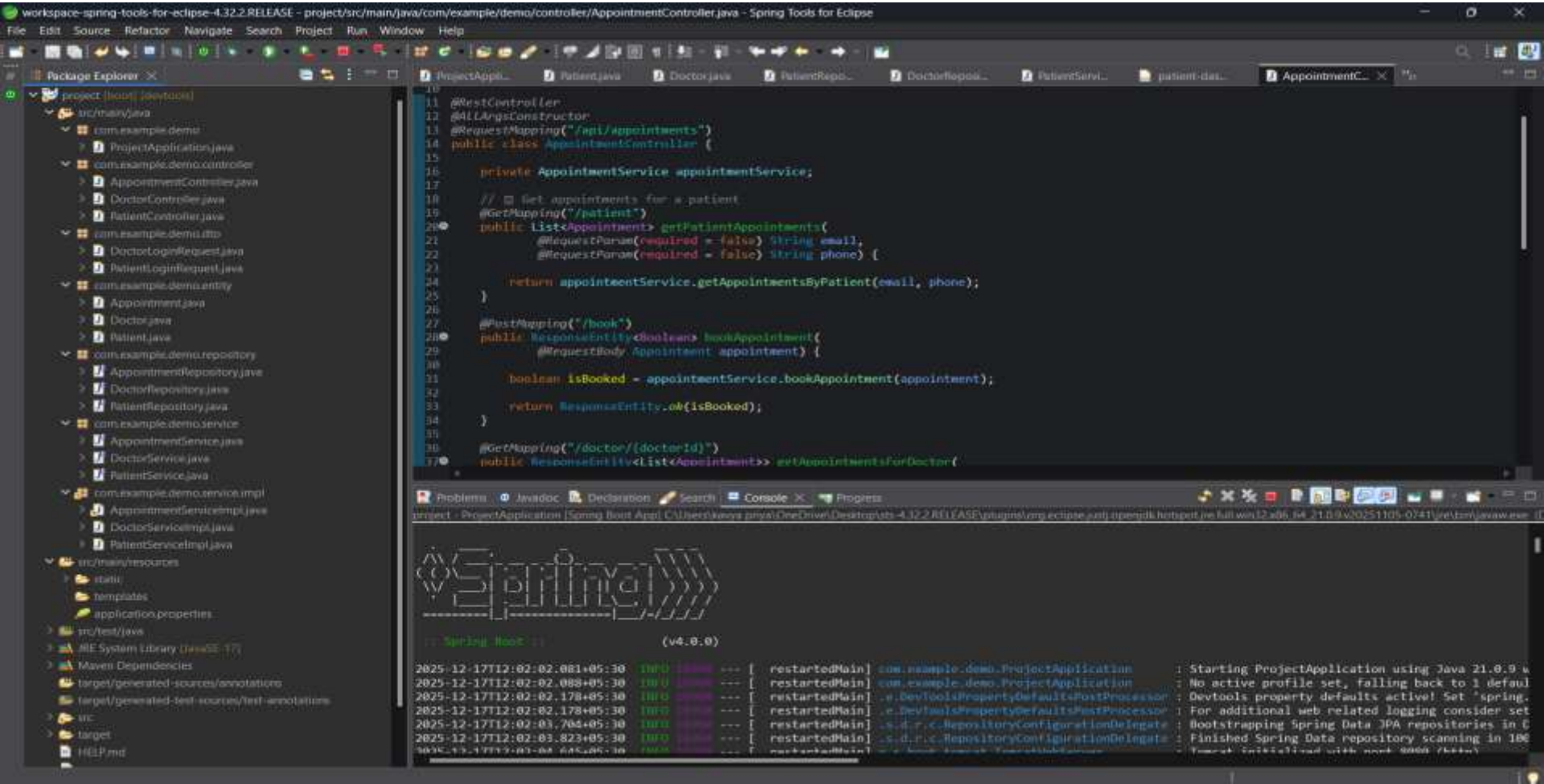
# GET METHOD FOR VIEWING DOCTOR APPOINTMENTS



# DELETE METHOD FOR CANCELLING APPOINTMENT



# WORKING OF SPRING TOOL



# WEBSITE DRIVE LINK

Link: https://drive.google.com/file/d/1wN6ZVlBeZeLfZAu-ZThr-UuoN0sXMshQ/view?usp=sharing