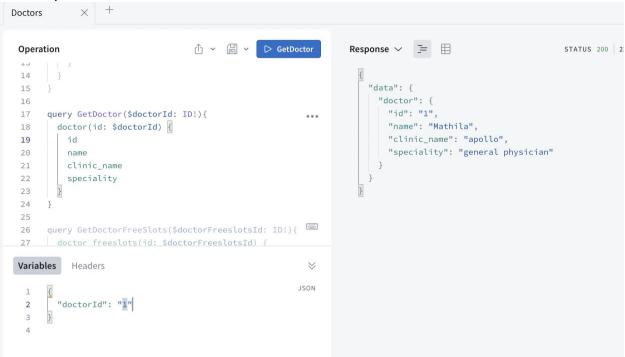
Implementation Task:

Testing for each Testcase Identifier:

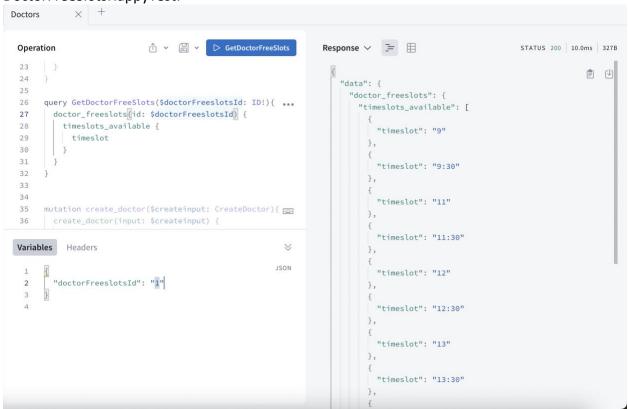
DoctorByIdHappyTest:

```
① ∨ 🖺 ∨ D GetDoctor
Operation
                                                              Response \vee = \equiv
                                                                                                       STATUS 200 | 11.0ms | 25B
                                                                                                                    "data": {
15
                                                                   "doctor": null
16
17
     query GetDoctor($doctorId: ID!){
      doctor(id: $doctorId) {
18
19
20
        name
21
       clinic_name
 22
        speciality
23
24
25
    query GetDoctorFreeSlots($doctorFreeslotsId: ID!){
27 doctor freeslots(id: $doctorFreeslotsId) {
Variables Headers
                                                     JSON
```

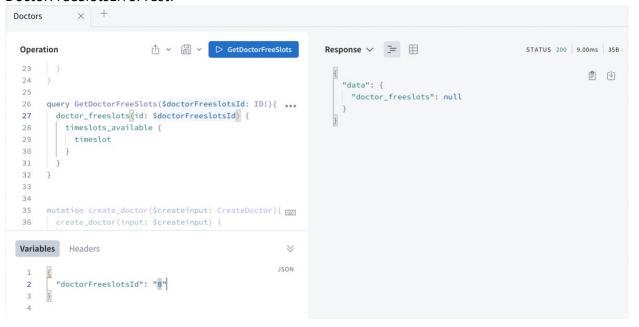
DoctorByIdErrorTest:



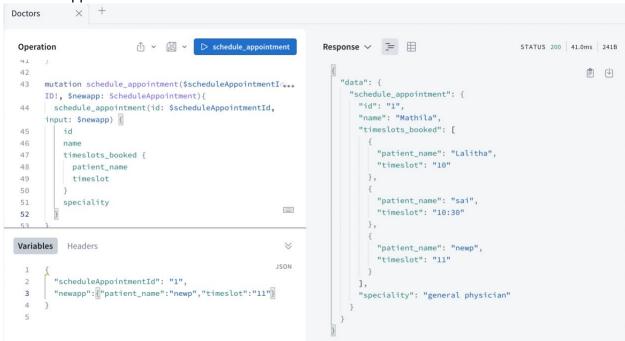
DoctorFreeSlotsHappyTest:



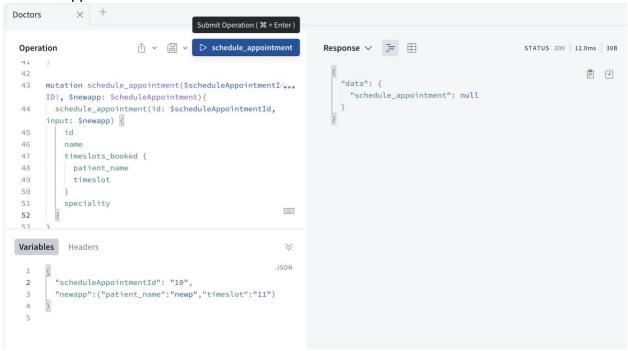
DoctorFreeSlotsErrorTest:



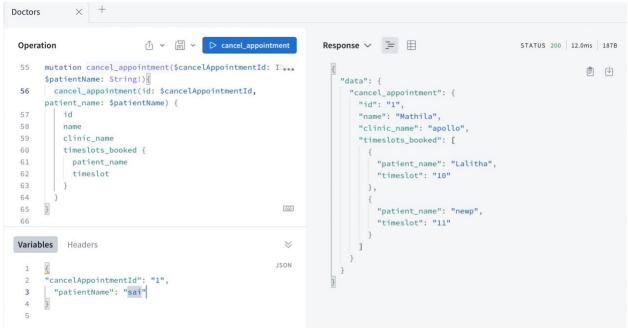
Schedule Appointment HT est:



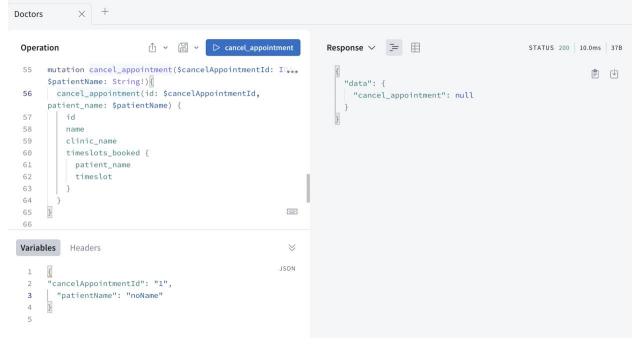
ScheduleAppointmentETest:



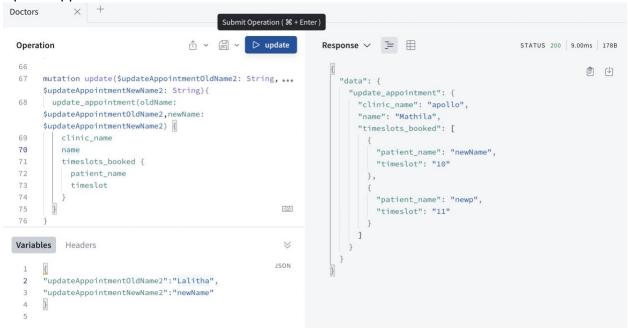
CancelAppointmentHTest:



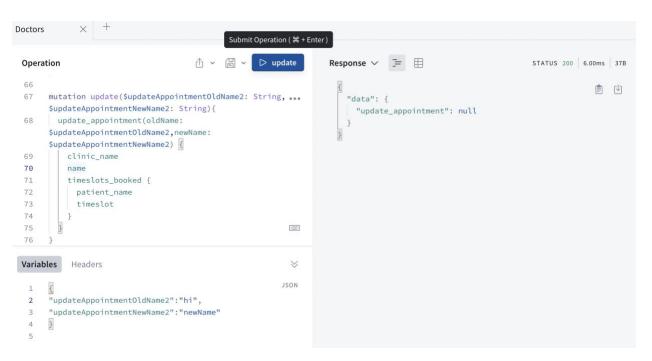
CancelAppointmentETest:



UpdateAppointmentHTest:



UpdateAppointmentETest:



Extra Queries and Mutations Tests:

```
① v 🗒 v Doctors
Operation
                                                    Response V = =
                                                                                            STATUS 200 | 36.0ms | 706B
     query Doctors {
                                                                                                          doctors {
                                                       "data": {
 2
 3
         id
                                                         "doctors": [
 4
         name
                                                             "id": "1",
 5
         clinic_name
                                                             "name": "Mathila",
 6
         speciality
         timeslots_available {
                                                             "clinic_name": "apollo",
         timeslot
                                                             "speciality": "general physician",
 8
                                                             "timeslots_available": [
 9
10
         timeslots_booked {
                                                                 "timeslot": "9"
11
         patient_name
          timeslot
12
13
                                                                "timeslot": "9:30"
14
15
16
                                                                "timeslot": "11"
17
     query GetDoctor($doctorId: ID!){
     doctor(id: $doctorId) {
18
19
       name
       clinic_name
20
                                                                "timeslot": "11:30"
21
       speciality
22
23
                                                                 "timeslot": "12"
24
    query GetDoctorFreeSlots
     ($doctorFrancloteId: IDI)
                                                                 "timeslot": "12:30"
Variables Headers
                                                                 "timeslot": "13"
```

```
① ∨ 🖺 ∨ D GetDoctor
                                                        Response \vee \equiv \boxplus
                                                                                                    STATUS 200 | 7.00ms | 95B
Operation
16
      query GetDoctor($doctorId: ID!){
                                                                                                                  17
                                                           "data": {
        doctor(id: $doctorId) {
                                                             "doctor": {
19
         name
                                                             "name": "Mathila",
 20
          clinic_name
                                                              "clinic_name": "apollo",
 21
          speciality
                                                             "speciality": "general physician"
                                               [222]
22
 23
Variables Headers
                                               \forall
                                             JSON
        "doctorId": "1"
```

Reflection Task:

What were some of the alternative schema and query design options you considered? Why did you choose the selected options?

Answer: The alternative schema and query design that was considered is to have separate doctors and events details and map a variable in between them. But the current schema and queries are considered to make the API easy to understand and implement. By having all the details in a single object is going to make the traversal and understanding easy.

Consider the case where, in future, the 'Event' structure is changed to have more fields e.g reference to patient details, consultation type (first time/follow-up etc.) and others.

O What changes will the clients (API consumer) need to make to their existing queries (if any).

Answer: Queries would remain the same but variables will be added in timeslots_booked There would be changes in timeslots_booked object which inturn has timeslot and patient_name as a list of objects.

O How will you accommodate the changes in your existing Schema and Query types?

Answer: The changes are accommodated in the schema to add the additional fields. Queries would remain the same for most of them mentioned.

The existing schema would get changed to this:

```
type Details {
    patient_name: String
    patient_age: Int
    patient_contact: String
}

type Booked {
    timeslot: String
    consultation_type:String
    patient_details:[Details]
}

type Doctor{
    id: ID!
    name: String!
    clinic_name: String!
    speciality: String!
    timeslots_available: [Free]
    timeslots_booked: [Booked]
}
```

Describe two GraphQL best practices that you have incorporated in your API design. GraphQL variables were used to provide arguments, naming all queries and querying the data that is only needed are few of the GraphQL best practices that are incorporated in this API design.