Implementation Task:
Testing for each Testcase Identifier:

DoctorByIdHappyTest:

Operation

GetDoctor

```
query GetDoctor($doctorId: ID!){
  doctor(id: $doctorId) {
    id
    name
    clinic_name
    speciality
  }
}

query GetDoctorFreeSlots($doctorFreeslotsId: ID!){
  doctor_freeslots(id: $doctorFreeslotsId) {
```

Variables | Headers

JSON

```
{
  "doctorId": "6"
}
```

Response

STATUS 200 | 11.0ms | 25B

```
{
  "data": {
    "doctor": null
  }
}
```

DoctorByIdErrorTest:

Doctors

Operation

GetDoctor

```
query GetDoctor($doctorId: ID!){
  doctor(id: $doctorId) {
    id
    name
    clinic_name
    speciality
  }
}

query GetDoctorFreeSlots($doctorFreeslotsId: ID!){
  doctor_freeslots(id: $doctorFreeslotsId) {
```

Variables | Headers

JSON

```
{
  "doctorId": "1"
}
```

Response

STATUS 200

```
{
  "data": {
    "doctor": {
      "id": "1",
      "name": "Mathila",
      "clinic_name": "apollo",
      "speciality": "general physician"
    }
  }
}
```

DoctorFreeSlotsHappyTest:

Doctors

Operation

GetDoctorFreeSlots

```
23    }
24  }
25
26  query GetDoctorFreeSlots($doctorFreeslotsId: ID!){
27    doctor_freeslots(id: $doctorFreeslotsId) {
28      timeslots_available {
29        timeslot
30      }
31    }
32  }
33
34
35  mutation create_doctor($createinput: CreateDoctor){
36    create_doctor(input: $createinput) {
```

Variables | Headers

JSON

```
1  {
2    "doctorFreeslotsId": "1"
3  }
4
```

Response

STATUS 200 | 10.0ms | 327B


```
{
  "data": {
    "doctor_freeslots": {
      "timeslots_available": [
        {
          "timeslot": "9"
        },
        {
          "timeslot": "9:30"
        },
        {
          "timeslot": "11"
        },
        {
          "timeslot": "11:30"
        },
        {
          "timeslot": "12"
        },
        {
          "timeslot": "12:30"
        },
        {
          "timeslot": "13"
        },
        {
          "timeslot": "13:30"
        },
        {
```


DoctorFreeSlotsErrorTest:

Doctors

Operation

GetDoctorFreeSlots

```
23    }
24  }
25
26  query GetDoctorFreeSlots($doctorFreeslotsId: ID!){
27    doctor_freeslots(id: $doctorFreeslotsId) {
28      timeslots_available {
29        timeslot
30      }
31    }
32  }
33
34
35  mutation create_doctor($createinput: CreateDoctor){
36    create_doctor(input: $createinput) {
```

Variables | Headers

JSON

```
1  {
2    "doctorFreeslotsId": "8"
3  }
4
```

Response

STATUS 200 | 9.00ms | 35B


```
{
  "data": {
    "doctor_freeslots": null
  }
}
```

ScheduleAppointmentHTest:

Doctors

Operation

schedule_appointment

```
mutation schedule_appointment($scheduleAppointmentId!, $newapp: ScheduleAppointment){
  schedule_appointment(id: $scheduleAppointmentId, input: $newapp) {
    id
    name
    timeslots_booked {
      patient_name
      timeslot
    }
    speciality
  }
}
```

Variables | Headers

JSON

```
{
  "scheduleAppointmentId": "1",
  "newapp":{"patient_name":"newp","timeslot":"11"}
}
```

Response

STATUS 200 | 41.0ms | 241B


```
{
  "data": {
    "schedule_appointment": {
      "id": "1",
      "name": "Mathila",
      "timeslots_booked": [
        {
          "patient_name": "Lalitha",
          "timeslot": "10"
        },
        {
          "patient_name": "sai",
          "timeslot": "10:30"
        },
        {
          "patient_name": "newp",
          "timeslot": "11"
        }
      ],
      "speciality": "general physician"
    }
  }
}
```


ScheduleAppointmentETest:

Doctors

Submit Operation (⌘ + Enter)

Operation

schedule_appointment

```
mutation schedule_appointment($scheduleAppointmentId!, $newapp: ScheduleAppointment){
  schedule_appointment(id: $scheduleAppointmentId, input: $newapp) {
    id
    name
    timeslots_booked {
      patient_name
      timeslot
    }
    speciality
  }
}
```

Variables | Headers

JSON

```
{
  "scheduleAppointmentId": "10",
  "newapp":{"patient_name":"newp","timeslot":"11"}
}
```

Response

STATUS 200 | 12.0ms | 39B


```
{
  "data": {
    "schedule_appointment": null
  }
}
```

CancelAppointmentHTest:

Doctors

Operation

cancel_appointment

```
mutation cancel_appointment($cancelAppointmentId: I...
$patientName: String!){
  cancel_appointment(id: $cancelAppointmentId,
patient_name: $patientName) {
    id
    name
    clinic_name
    timeslots_booked {
      patient_name
      timeslot
    }
  }
}
```

Variables Headers

JSON

```
{
"cancelAppointmentId": "1",
  "patientName": "sai"
}
```

Response

STATUS 200 | 12.0ms | 187B


```
{
  "data": {
    "cancel_appointment": {
      "id": "1",
      "name": "Mathila",
      "clinic_name": "apollo",
      "timeslots_booked": [
        {
          "patient_name": "Lalitha",
          "timeslot": "10"
        },
        {
          "patient_name": "newp",
          "timeslot": "11"
        }
      ]
    }
  }
}
```


CancelAppointmentETest:

Doctors

Operation

cancel_appointment

```
mutation cancel_appointment($cancelAppointmentId: I...
$patientName: String!){
  cancel_appointment(id: $cancelAppointmentId,
patient_name: $patientName) {
    id
    name
    clinic_name
    timeslots_booked {
      patient_name
      timeslot
    }
  }
}
```

Variables Headers

JSON

```
{
"cancelAppointmentId": "1",
  "patientName": "noName"
}
```

Response

STATUS 200 | 10.0ms | 37B


```
{
  "data": {
    "cancel_appointment": null
  }
}
```

UpdateAppointmentHTest:

Doctors

Submit Operation (⌘ + Enter)

Operation

update

```
mutation update($updateAppointmentOldName2: String, $updateAppointmentNewName2: String){
  update_appointment(oldName: $updateAppointmentOldName2,newName: $updateAppointmentNewName2) {
    clinic_name
    name
    timeslots_booked {
      patient_name
      timeslot
    }
  }
}
```

Variables Headers

JSON

```
{
"updateAppointmentOldName2":"Lalitha",
"updateAppointmentNewName2":"newName"
}
```

Response

STATUS 200 | 9.00ms | 178B

```
{
  "data": {
    "update_appointment": {
      "clinic_name": "apollo",
      "name": "Mathila",
      "timeslots_booked": [
        {
          "patient_name": "newName",
          "timeslot": "10"
        },
        {
          "patient_name": "newp",
          "timeslot": "11"
        }
      ]
    }
  }
}
```

UpdateAppointmentETest:

Doctors

Submit Operation (⌘ + Enter)

Operation

update

```
mutation update($updateAppointmentOldName2: String, $updateAppointmentNewName2: String){
  update_appointment(oldName: $updateAppointmentOldName2,newName: $updateAppointmentNewName2) {
    clinic_name
    name
    timeslots_booked {
      patient_name
      timeslot
    }
  }
}
```

Variables Headers

JSON

```
{
"updateAppointmentOldName2":"hi",
"updateAppointmentNewName2":"newName"
}
```

Response

STATUS 200 | 6.00ms | 37B

```
{
  "data": {
    "update_appointment": null
  }
}
```

Extra Queries and Mutations Tests:

Operation

Doctors

```
query Doctors {
  doctors {
    id
    name
    clinic_name
    speciality
    timeslots_available {
      timeslot
    }
    timeslots_booked {
      patient_name
      timeslot
    }
  }
}

query GetDoctor($doctorId: ID!){
  doctor(id: $doctorId) {
    name
    clinic_name
    speciality
  }
}

query GetDoctorFreeSlots
```

Variables Headers

Response

STATUS 200 | 36.0ms | 706B


```
{
  "data": {
    "doctors": [
      {
        "id": "1",
        "name": "Mathila",
        "clinic_name": "apollo",
        "speciality": "general physician",
        "timeslots_available": [
          {
            "timeslot": "9"
          },
          {
            "timeslot": "9:30"
          },
          {
            "timeslot": "11"
          },
          {
            "timeslot": "11:30"
          },
          {
            "timeslot": "12"
          },
          {
            "timeslot": "12:30"
          },
          {
            "timeslot": "13"
```


Operation

GetDoctor

```
query GetDoctor($doctorId: ID!){
  doctor(id: $doctorId) {
    name
    clinic_name
    speciality
  }
}
```

Variables Headers

JSON

```
{
  "doctorId": "1"
}
```

Response

STATUS 200 | 7.00ms | 95B


```
{
  "data": {
    "doctor": {
      "name": "Mathila",
      "clinic_name": "apollo",
      "speciality": "general physician"
    }
  }
}
```

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.