**Online Doctor Appointment Booking System**

**List of APIs developed for the Project:**

1. To fetch Patient Details based on Mobile Number, so that if no record found for the given number they can register as a new patient.

* Verb: Get
* Parameters passed in request URL: Mobile number.
* Authentication used: No
* API Endpoint : http://localhost:8200/GetPatientByMobileno/8754972498
* Sample Response

{

    "patientId": 252,

    "patientName": "Yasvand Cumaar",

    "gender": "Male",

    "dateofbirth": "1996-09-19",

    "mobileno": "8754972498",

    "mailid": "cumaaryasvand@gmail.com",

    "createdDate": "2023-05-12",

    "createddate": "2023-05-12"

}

1. To fetch Patient Details based on Mail Id, so that if no record found for the given mail id they can register as a new patient.

* Verb: Get
* Parameters passed in request URL: Mail Id.
* Authentication used: No
* API Endpoint : http://localhost:8200/GetPatientByMailId/cumaaryasvand@gmail.com
* Sample Response

{

    "patientId": 252,

    "patientName": "Yasvand Cumaar",

    "gender": "Male",

    "dateofbirth": "1996-09-19",

    "mobileno": "8754972498",

    "mailid": "cumaaryasvand@gmail.com",

    "createdDate": "2023-05-12",

    "createddate": "2023-05-12"

}

1. Registering as a new Patient

* Verb: Post
* Authentication used: No
* API Endpoint : http://localhost:8200/PostPatient
* Sample Request

{

    "patientName":"Sathiyanarayanan",

    "gender":"Male",

    "dateofbirth":"1966-04-16",

    "mobileno":"9486623723",

    "mailid":"test@gmail.com",

    "createdDate":"2023-05-12"

}

1. Deleting a registered Patient

* Verb: Delete
* Authentication used: Yes
* API Endpoint : <http://localhost:8200/DeletePatient/PatientId>.

1. Get a list of all registered Patient

* Verb: Get
* Authentication used: Yes
* API Endpoint : http://localhost:8200/ListPatients

[

    {

        "patientId": 302,

        "patientName": "Sathiyanarayanan",

        "gender": "Male",

        "dateofbirth": "1966-04-16",

        "mobileno": "9486623723",

        "mailid": "test@gmail.com",

        "createdDate": "2023-05-12",

        "createddate": "2023-05-12"

    },

    {

        "patientId": 252,

        "patientName": "Yasvand Cumaar",

        "gender": "Male",

        "dateofbirth": "1996-09-19",

        "mobileno": "8754972498",

        "mailid": "cumaaryasvand@gmail.com",

        "createdDate": "2023-05-12",

        "createddate": "2023-05-12"

    },

    {

        "patientId": 255,

        "patientName": "Sornalatha",

        "gender": "Female",

        "dateofbirth": "1972-09-22",

        "mobileno": "7708512756",

        "mailid": "sornalathasathiyanarayanan@gmail.com",

        "createdDate": "2023-05-12",

        "createddate": "2023-05-12"

    }

]

1. To get the list of Specialities

* Verb: Get
* Parameters passed in request URL: None
* Authentication used: Yes.
* API Endpoint : http://localhost:8200/ListSpeciality
* Sample Response

[

{

"specialityCode": "215",

"specialityName": "ENT"

},

{

"specialityCode": "251",

"specialityName": "CARDIOLOGY"

},

{

"specialityCode": "151",

"specialityName": "Oncology"

},

{

"specialityCode": "130",

"specialityName": "Dermatology"

},

{

"specialityCode": "110",

"specialityName": "General Physician"

},

{

"specialityCode": "120",

"specialityName": "Urology"

},

{

"specialityCode": "156",

"specialityName": "Internal Medicine"

},

{

"specialityCode": "166",

"specialityName": "Critical Care"

},

{

"specialityCode": "100",

"specialityName": "General Surgery"

}

]

1. To get, list of Practitioner names and their details based on the chosen Speciality.

* Verb: Get
* Parameters passed in request URL: SpecialityCode
* Authentication used: No.
* API Endpoint : http://localhost:8200/FindPractitionersBySpecialityCode/215
* Sample Response

[

    {

        "practitionerId": "714518",

        "practitionerName": "Dr. Ganapathy H",

        "specialityCode": "215",

        "yearsofexperience": "10",

        "registrationNumber": "617645",

        "qualification": "MBBS",

        "languagesknown": "Tamil, English",

        "consultationFees": "600",

        "consultationTimings": "06.00 PM To 08.30 PM"

    },

    {

        "practitionerId": "714522",

        "practitionerName": "Dr. Nithya Narayanan",

        "specialityCode": "215",

        "yearsofexperience": "4",

        "registrationNumber": "61765",

        "qualification": "MBBS",

        "languagesknown": "Tamil, English, Hindi",

        "consultationFees": "300",

        "consultationTimings": "04.00 PM To 07.30 PM"

    },

    {

        "practitionerId": "714641",

        "practitionerName": "Dr. Prabhakaran M",

        "specialityCode": "215",

        "yearsofexperience": "6",

        "registrationNumber": "6765",

        "qualification": "MBBS",

        "languagesknown": "Tamil, English, Hindi",

        "consultationFees": "400",

        "consultationTimings": "10.00 AM To 12.30 PM"

    }

]

1. To get available slots for the chosen Practitioner and required date in which appointment needs to be booked.

* Verb: Get
* Parameters passed in request URL: doctorId, slotDate, slotAvailability=True
* Authentication used: Yes.
* API Endpoint : http://localhost:8200/GetSlots?doctorId=700576&slotDate=09-05-2023&slotAvailability=True
* Sample Response

[

{

"slotDate": "09-05-2023",

"doctorId": "700576",

"slotstarttime": "12.00",

"slotendtime": "13.00",

"slotAvailability": "True",

"slotid": 105

},

{

"slotDate": "09-05-2023",

"doctorId": "700576",

"slotstarttime": "08.00",

"slotendtime": "09.00",

"slotAvailability": "True",

"slotid": 106

}

]

1. Booking an appointment.

* Verb: Post
* Authentication used: Yes.
* API Endpoint : http://localhost:8200/CreateAppointment
* Sample Request

{

    "patientId": "252",

    "doctorId": "700576",

    "slotId": "104",

    "status": "A",

    "createdDate":"08-05-2023",

    "apptStartTime":"11.00",

    "apptEndTime":"12.00",

    "apptDate":"09-05-2023"

}

1. To get list of all appointments based on Patient Id.

* Verb: Get
* Parameters passed in request URL: Patient Id
* Authentication used: Yes.
* API Endpoint : http://localhost:8200/GetAppointmentsByPatientId/652
* Sample Response

[

    {

        "appointmentId": 352,

        "patientId": "252",

        "doctorId": "700576",

        "slotId": "103",

        "status": "D",

        "createdDate": "2023-05-12",

        "apptStartTime": "10.00",

        "apptEndTime": "11.00",

        "apptDate": "2023-05-09",

        "updatedDate": **null**

    },

    {

        "appointmentId": 452,

        "patientId": "252",

        "doctorId": "700576",

        "slotId": "104",

        "status": "A",

        "createdDate": "08-05-2023",

        "apptStartTime": "11.00",

        "apptEndTime": "12.00",

        "apptDate": "09-05-2023",

        "updatedDate": **null**

    }

]

1. Cancelling an appointment

* Verb: Put
* Authentication used: Yes.
* API Endpoint : http://localhost:8200/CancelAppointment
* Sample Request

{

    "appointmentId": 452,

    "patientId": "252",

    "doctorId": "700576",

    "slotId": "103",

    "status": "D",

    "createdDate": "05-05-2023",

    "apptStartTime": "11.00",

    "apptEndTime": "12.00",

    "apptDate": "09-05-2023",

    "updatedDate": "06-05-2023"

}

**Pre-requisite**

1. Database tables

\*Table creation – “**SPECIALITYMASTER**” table needs to be created manually and data needs to be inserted through Insert scripts.

\*Data Insertion:

\*In “**PRACTITIONERSLOTS**” table data alone need to inserted through scripts. That table hold all the slot details for a Practitioner.

\*In “**PRACTITIONER**” table data alone need to be inserted through scripts. That table hold all the basic details of a Practitioner.

1. Database sequences

\* “**PRACTSLOTS\_SEQ**” created to insert the sequence id at the time of inserting slot details in “PRACTITIONERSLOTS” table.

1. Database triggers

\*”**UPDATE\_SLOT\_STATUS**”, “**UPDATE\_SLOT\_STATUSS**” are the two database triggers created for updating the slot availability in “**PRACTITIONERSLOTS**” table. When any appointment is booked then that particular **Slot\_Availablity** will be updated as “False” and if we cancel the appointment then **Slot\_Availability** will be updated as “True” so that others can use that slot for booking.

“**deletedpatient\_trigger**” trigger created to update the deleted patients in deletedpatients table before deleting in patient table.

Attached file contain the DB scripts for creation of above-mentioned things.



**Following are the functionalities not supported in this project**.

\*We can’t be able to update any Patient Personal Details.

\*After booking an appointment we don’t have option to reschedule an appointment. If required, we need to cancel and then book.