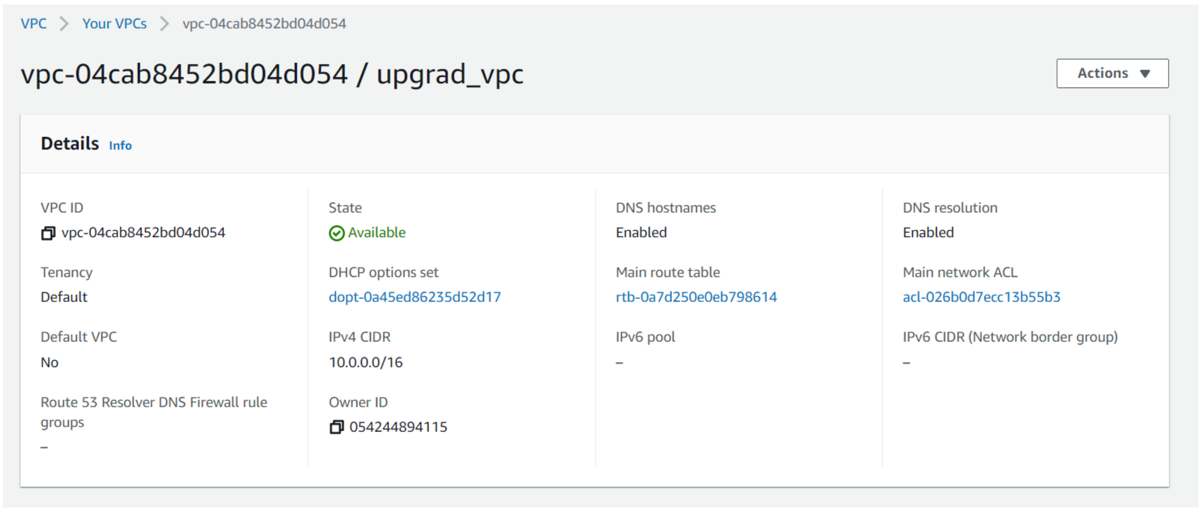
# Capstone Project - BookMyConsultation

Author: Anubhav Apurva
Email ID: anubhavh3@gmail.com
Batch: SD C25 Dec'20

## Table of content

# 1. Environment Setup

## 1.1. VPC Setup



VPC > Your VPCs > vpc-04cab8452bd04d054

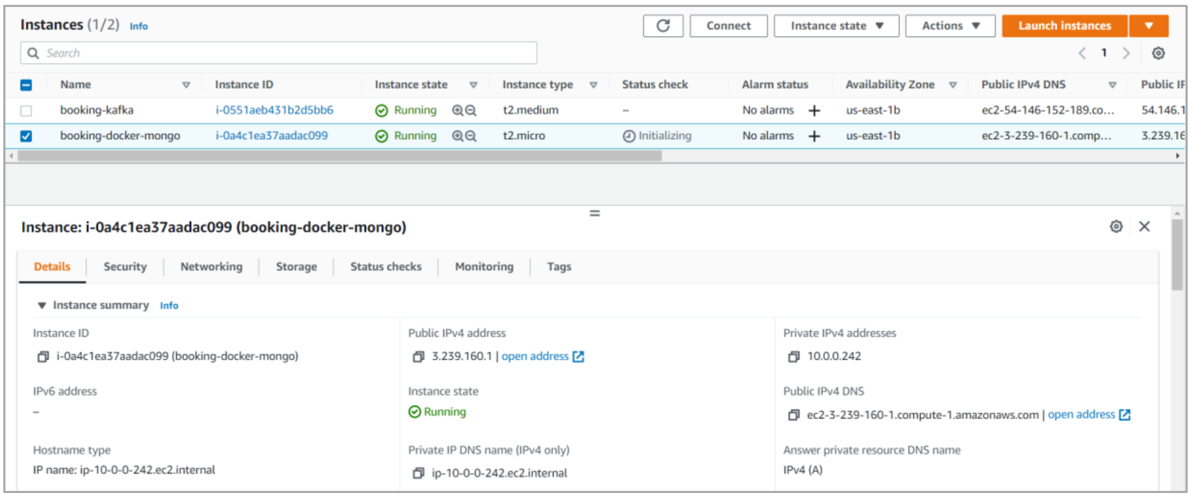
vpc-04cab8452bd04d054 / upgrad_vpc

Details Info

| | | | |
|---|---|---|---|
| VPC ID<br>vpc-04cab8452bd04d054 | State<br>✓ Available | DNS hostnames<br>Enabled | DNS resolution<br>Enabled |
| Tenancy<br>Default | DHCP options set<br>dopt-0a45ed86235d52d17 | Main route table<br>rtb-0a7d250e0eb798614 | Main network ACL<br>acl-026b0d7ecc13b55b3 |
| Default VPC<br>No | IPv4 CIDR<br>10.0.0.0/16 | IPv6 pool<br>– | IPv6 CIDR (Network border group)<br>– |
| Route 53 Resolver DNS Firewall rule groups<br>– | Owner ID<br>054244894115 | | |

## 1.2. EC2 Setup

In this project we need two EC2 instances. The first instance (booking-docker-mongo in this screenshot) will be used to run MongoDB and deploy the developed microservices. The second instance (booking-kafka in this screenshot) will be used to run Kafka.



Instances (1/2) Info

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 DNS | Public IF |
|---|---|---|---|---|---|---|---|---|
| booking-kafka | i-0551aeb431b2d5bb6 | ✓ Running | t2.medium | – | No alarms | us-east-1b | ec2-54-146-152-189.co... | 54.146.1 |
| booking-docker-mongo | i-0a4c1ea37aadac099 | ✓ Running | t2.micro | Initializing | No alarms | us-east-1b | ec2-3-239-160-1.comp... | 3.239.16 |

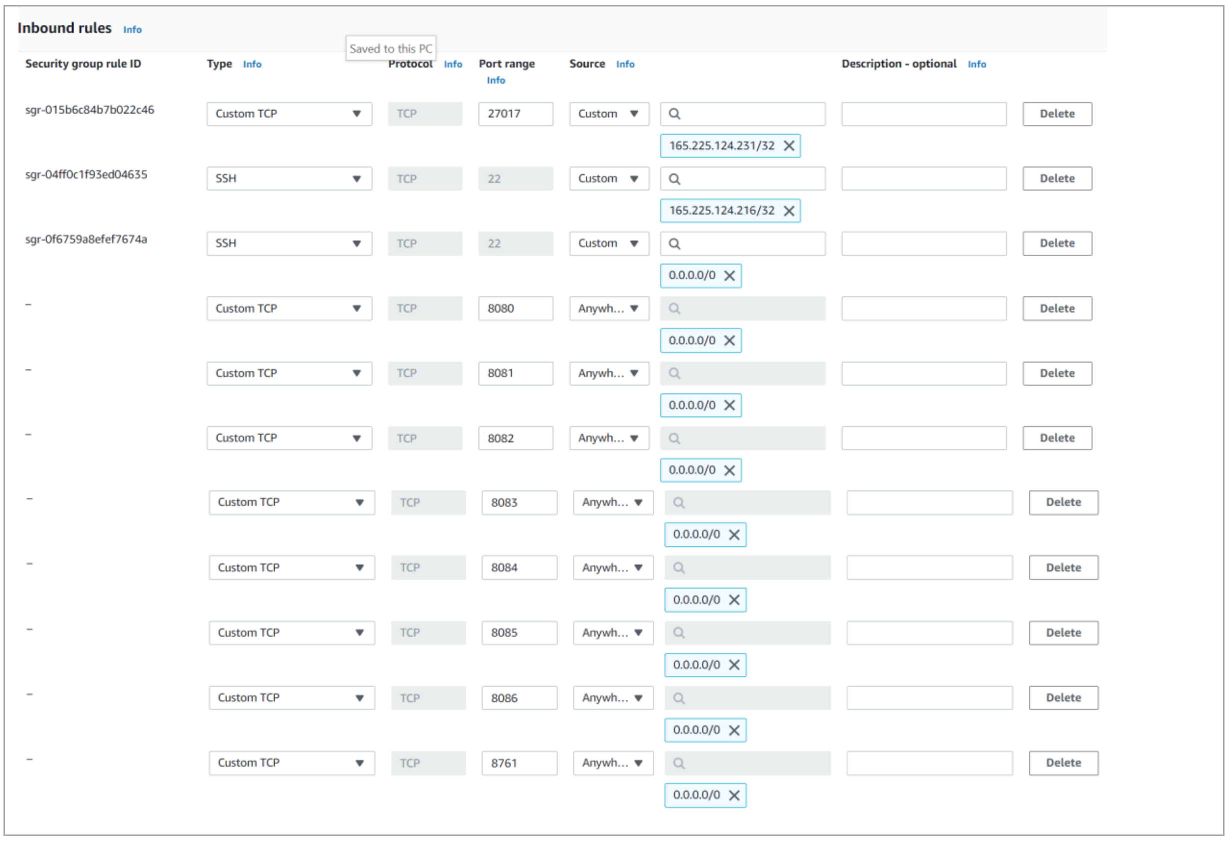
Instance: i-0a4c1ea37aadac099 (booking-docker-mongo)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary Info

| | | |
|---|---|---|
| Instance ID<br>i-0a4c1ea37aadac099 (booking-docker-mongo) | Public IPv4 address<br>3.239.160.1 \| open address | Private IPv4 addresses<br>10.0.0.242 |
| IPv6 address<br>– | Instance state<br>✓ Running | Public IPv4 DNS<br>ec2-3-239-160-1.compute-1.amazonaws.com \| open address |
| Hostname type<br>IP name: ip-10-0-0-242.ec2.internal | Private IP DNS name (IPv4 only)<br>ip-10-0-0-242.ec2.internal | Answer private resource DNS name<br>IPv4 (A) |

EC2 (first instance) Security Group Inbound Rules -

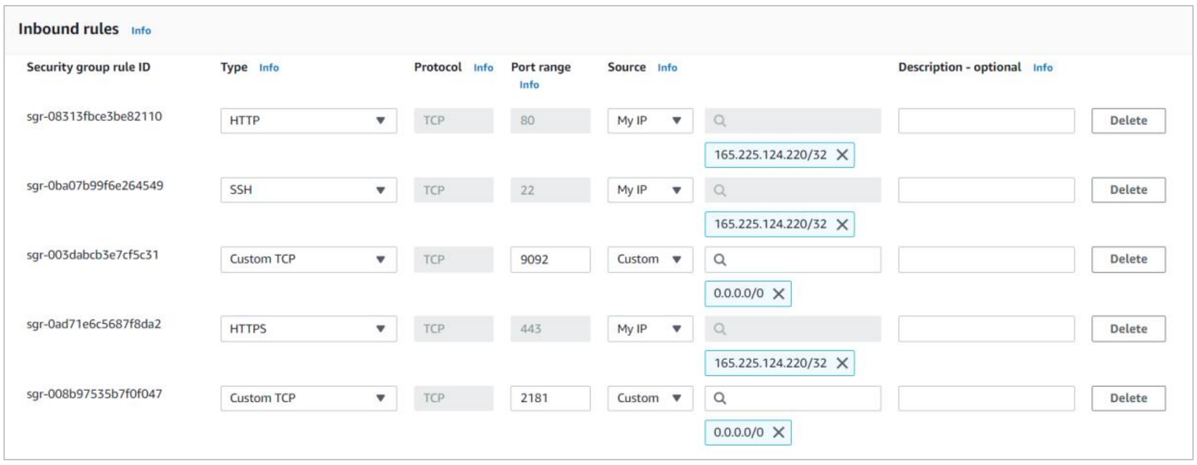


Inbound rules Info

Saved to this PC

| Security group rule ID | Type | Protocol | Port range | Source | | Description - optional | |
|---|---|---|---|---|---|---|---|
| sgr-015b6c84b7b022c46 | Custom TCP | TCP | 27017 | Custom | 165.225.124.231/32 | | Delete |
| sgr-04ff0c1f93ed04635 | SSH | TCP | 22 | Custom | 165.225.124.216/32 | | Delete |
| sgr-0f6759a8efef7674a | SSH | TCP | 22 | Custom | 0.0.0.0/0 | | Delete |
| – | Custom TCP | TCP | 8080 | Anywh... | 0.0.0.0/0 | | Delete |
| – | Custom TCP | TCP | 8081 | Anywh... | 0.0.0.0/0 | | Delete |
| – | Custom TCP | TCP | 8082 | Anywh... | 0.0.0.0/0 | | Delete |
| – | Custom TCP | TCP | 8083 | Anywh... | 0.0.0.0/0 | | Delete |
| – | Custom TCP | TCP | 8084 | Anywh... | 0.0.0.0/0 | | Delete |
| – | Custom TCP | TCP | 8085 | Anywh... | 0.0.0.0/0 | | Delete |
| – | Custom TCP | TCP | 8086 | Anywh... | 0.0.0.0/0 | | Delete |
| – | Custom TCP | TCP | 8761 | Anywh... | 0.0.0.0/0 | | Delete |

EC2 (second instance) Security Group Inbound Rules -

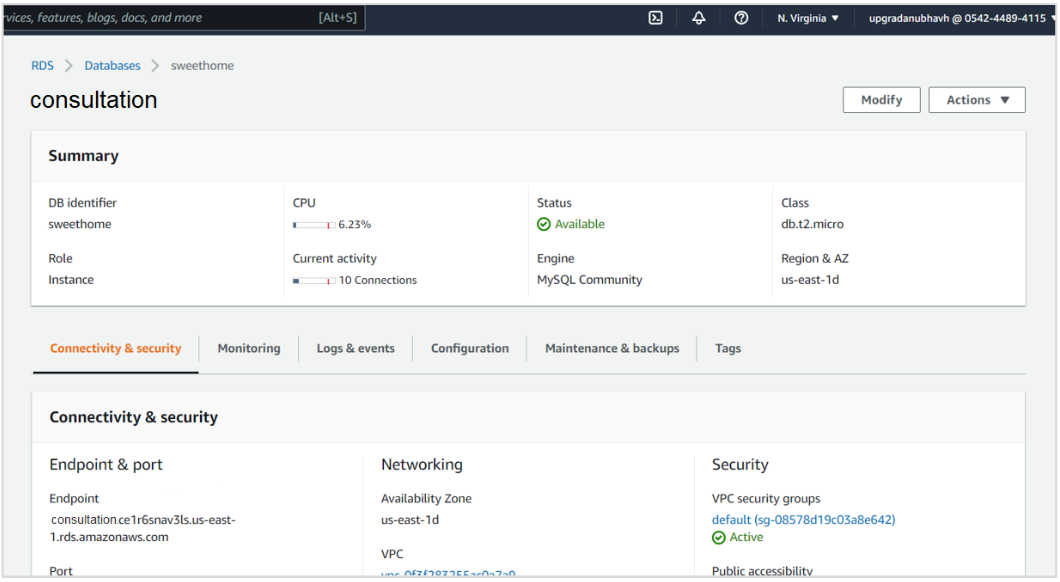


Inbound rules Info

| Security group rule ID | Type | Protocol | Port range | Source | | Description - optional | |
|---|---|---|---|---|---|---|---|
| sgr-08313fbce3be82110 | HTTP | TCP | 80 | My IP | 165.225.124.220/32 | | Delete |
| sgr-0ba07b99f6e264549 | SSH | TCP | 22 | My IP | 165.225.124.220/32 | | Delete |
| sgr-003dabcb3e7cf5c31 | Custom TCP | TCP | 9092 | Custom | 0.0.0.0/0 | | Delete |
| sgr-0ad71e6c5687f8da2 | HTTPS | TCP | 443 | My IP | 165.225.124.220/32 | | Delete |
| sgr-008b97535b7f0f047 | Custom TCP | TCP | 2181 | Custom | 0.0.0.0/0 | | Delete |

## 1.3. RDS Setup

Create a RDS MySQL instance and create a database (e.g. upgrad) in it.



RDS > Databases > sweethome

consultation

Summary

| | | | |
|---|---|---|---|
| DB identifier | CPU | Status | Class |
| sweethome | 6.23% | Available | db.t2.micro |
| Role | Current activity | Engine | Region & AZ |
| Instance | 10 Connections | MySQL Community | us-east-1d |

Connectivity & security

Endpoint & port

Endpoint
consultation.ce1r6snav3ls.us-east-1.rds.amazonaws.com

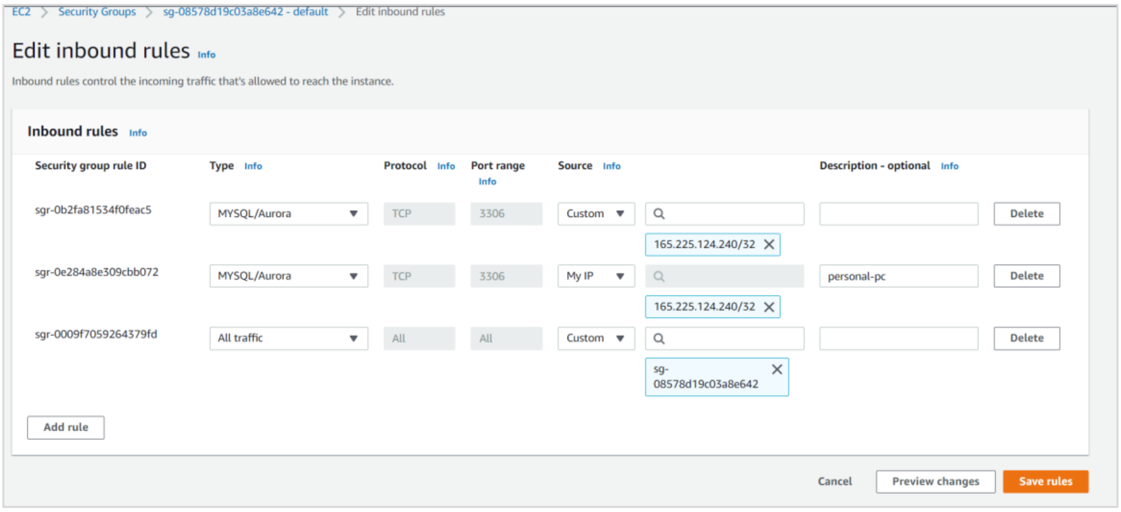
Networking

Availability Zone
us-east-1d

Security

VPC security groups
default (sg-08578d19c03a8e642)
Active

RDS Security Group Inbound Rules –



EC2 > Security Groups > sg-08578d19c03a8e642 - default > Edit inbound rules

Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules

| Security group rule ID | Type | Protocol | Port range | Source | | Description - optional |
|---|---|---|---|---|---|---|
| sgr-0b2fa81534f0feac5 | MYSQL/Aurora | TCP | 3306 | Custom | 165.225.124.240/32 | |
| sgr-0e284a8e309cbb072 | MYSQL/Aurora | TCP | 3306 | My IP | 165.225.124.240/32 | personal-pc |
| sgr-0009f7059264379fd | All traffic | All | All | Custom | sg-08578d19c03a8e642 | |

## 2. Steps to Deploy and Run (Without Docker)

- Start Kafka
  - Start Kafka server and zookeeper on the second EC2 instance

- Start MongoDBs
  - Start MongoDB either on the first EC2 instance or on the second EC2 instance.

- FTP BookMyConsultation.zip to the first EC2 instance

- Unzip Sweet-Home.zip

```
$ unzip BookMyConsultation.zip
```

- Configure ENV variables
  Set all the ENV variables in the env file in the root directory of the project. Export all the variables

```
$ . ./env
```

- cd to each microservice directory and bring up the service

```
$ cd BookMyConsultation/eureka
$ mvn spring-boot:run

$ cd BookMyConsultation/doctor-service
$ mvn spring-boot:run

$ cd BookMyConsultation/user-service
$ mvn spring-boot:run

$ cd BookMyConsultation/appointment-service
$ mvn spring-boot:run

$ cd BookMyConsultation/payment-service
$ mvn spring-boot:run

$ cd BookMyConsultation/rating-service
$ mvn spring-boot:run

$ cd BookMyConsultation/notification-service
$ mvn spring-boot:run

$ cd BookMyConsultation/bmc-gateway
$ mvn spring-boot:run
```

## 3. Steps to Deploy and Run (With Docker)

- Start Kafka
  - Start Kafka server and zookeeper on the second EC2 instance

- Start MongoDBs
  - Start MongoDB either on the first EC2 instance or on the second EC2 instance.

- FTP BookMyConsultation.zip to the first EC2 instance

- Unzip Sweet-Home.zip

```
$ unzip BookMyConsultation.zip
```

- Configure ENV variables
  Set all the ENV variables or edit them in docker-compose.yaml file in the root directory

- Create docker bridge network

```
$ sudo docker network create microservicesnet
```

- Create JAR of each service and build the docker image

```
$ cd BookMyConsultation/eureka
$ mvn clean install spring-boot:repackage -DskipTests
$ sudo docker build -t bookingaap/eurekasvc:1.0.0 .

$ cd BookMyConsultation/eureka
$ mvn clean install spring-boot:repackage -DskipTests
$ sudo docker build -t bookingaap/doctorsvc:1.0.0 .

And so on…
```

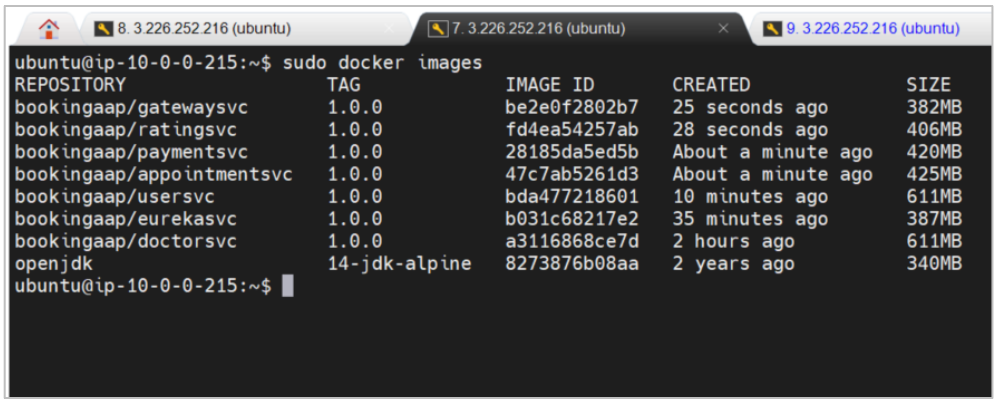
- Start services

```
$ sudo docker run -it --name=serviceregistry -d -p8761:8761 -e
EUREKA_HOST_NAME=54.87.134.192 --net=microservicesnet bookingaap/eurekasvc:1.0.0
```

Similarly, repeat above steps for each service.

Or use the docker-compose file to build images and deploy all the services at once

```
$ sudo docker-compose up -d
```


8. 3.226.252.216 (ubuntu)
7. 3.226.252.216 (ubuntu)
9. 3.226.252.216 (ubuntu)

```
ubuntu@ip-10-0-0-215:~$ sudo docker images
REPOSITORY                 TAG             IMAGE ID       CREATED              SIZE
bookingaap/gatewaysvc      1.0.0           be2e0f2802b7   25 seconds ago       382MB
bookingaap/ratingsvc       1.0.0           fd4ea54257ab   28 seconds ago       406MB
bookingaap/paymentsvc      1.0.0           28185da5ed5b   About a minute ago   420MB
bookingaap/appointmentsvc  1.0.0           47c7ab5261d3   About a minute ago   425MB
bookingaap/usersvc         1.0.0           bda477218601   10 minutes ago       611MB
bookingaap/eurekasvc       1.0.0           b031c68217e2   35 minutes ago       387MB
bookingaap/doctorsvc       1.0.0           a3116868ce7d   2 hours ago          611MB
openjdk                    14-jdk-alpine   8273876b08aa   2 years ago          340MB
ubuntu@ip-10-0-0-215:~$
```

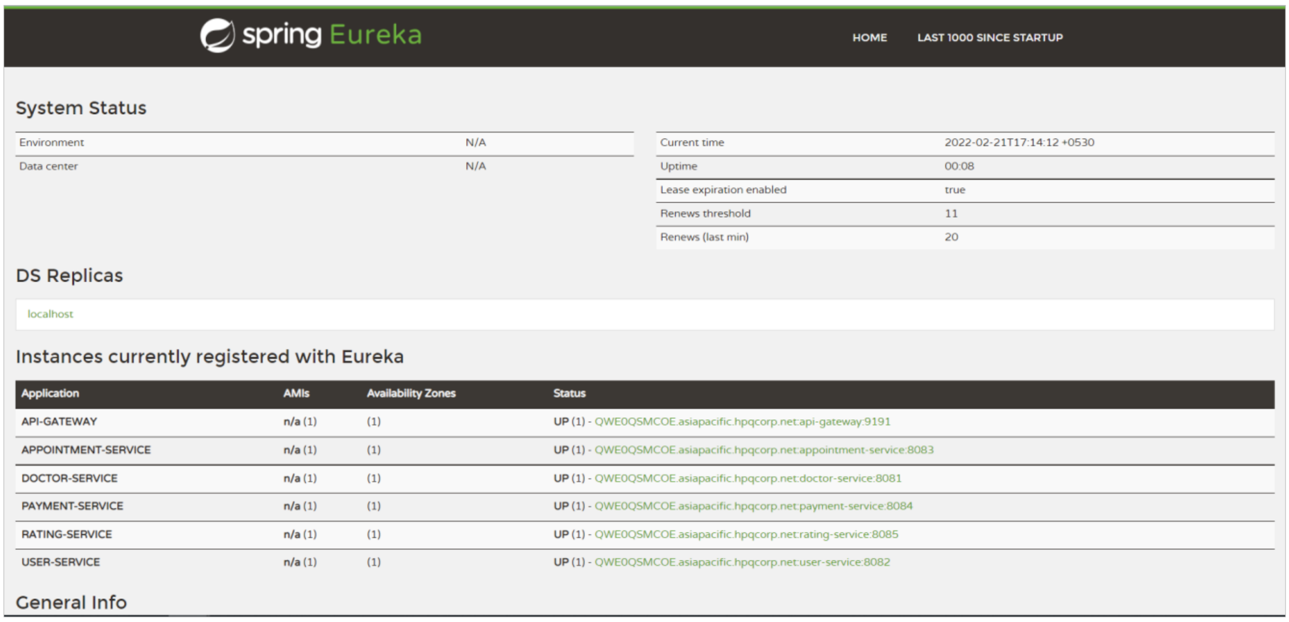
## 4. Eureka Server

Eureka Server is an application that holds the information about all client-service applications. It knows all the client applications running on each port and IP address. Eureka Server is also known as Discovery Server.

Every micro-service in this project including the API gateway registers themselves into the Eureka server.

The Eureka Server is started on port 8761.

On the browser if we go to http://localhost:8761/ or http://<ec2-host-ip>:8671/ we can see the services are up –



System Status

| | |
|---|---|
| Environment | N/A |
| Data center | N/A |

| | |
|---|---|
| Current time | 2022-02-21T17:14:12 +0530 |
| Uptime | 00:08 |
| Lease expiration enabled | true |
| Renews threshold | 11 |
| Renews (last min) | 20 |

DS Replicas

localhost

Instances currently registered with Eureka

| Application | AMIs | Availability Zones | Status |
|---|---|---|---|
| API-GATEWAY | n/a (1) | (1) | UP (1) - QWE0QSMCOE.asiapacific.hpqcorp.net:api-gateway:9191 |
| APPOINTMENT-SERVICE | n/a (1) | (1) | UP (1) - QWE0QSMCOE.asiapacific.hpqcorp.net:appointment-service:8083 |
| DOCTOR-SERVICE | n/a (1) | (1) | UP (1) - QWE0QSMCOE.asiapacific.hpqcorp.net:doctor-service:8081 |
| PAYMENT-SERVICE | n/a (1) | (1) | UP (1) - QWE0QSMCOE.asiapacific.hpqcorp.net:payment-service:8084 |
| RATING-SERVICE | n/a (1) | (1) | UP (1) - QWE0QSMCOE.asiapacific.hpqcorp.net:rating-service:8085 |
| USER-SERVICE | n/a (1) | (1) | UP (1) - QWE0QSMCOE.asiapacific.hpqcorp.net:user-service:8082 |

General Info

## 5. Doctor-Onboarding Service

### 5.1. Endpoint-1: Collect doctor information

POST localhost:8081/doctors

Content-Type application/json

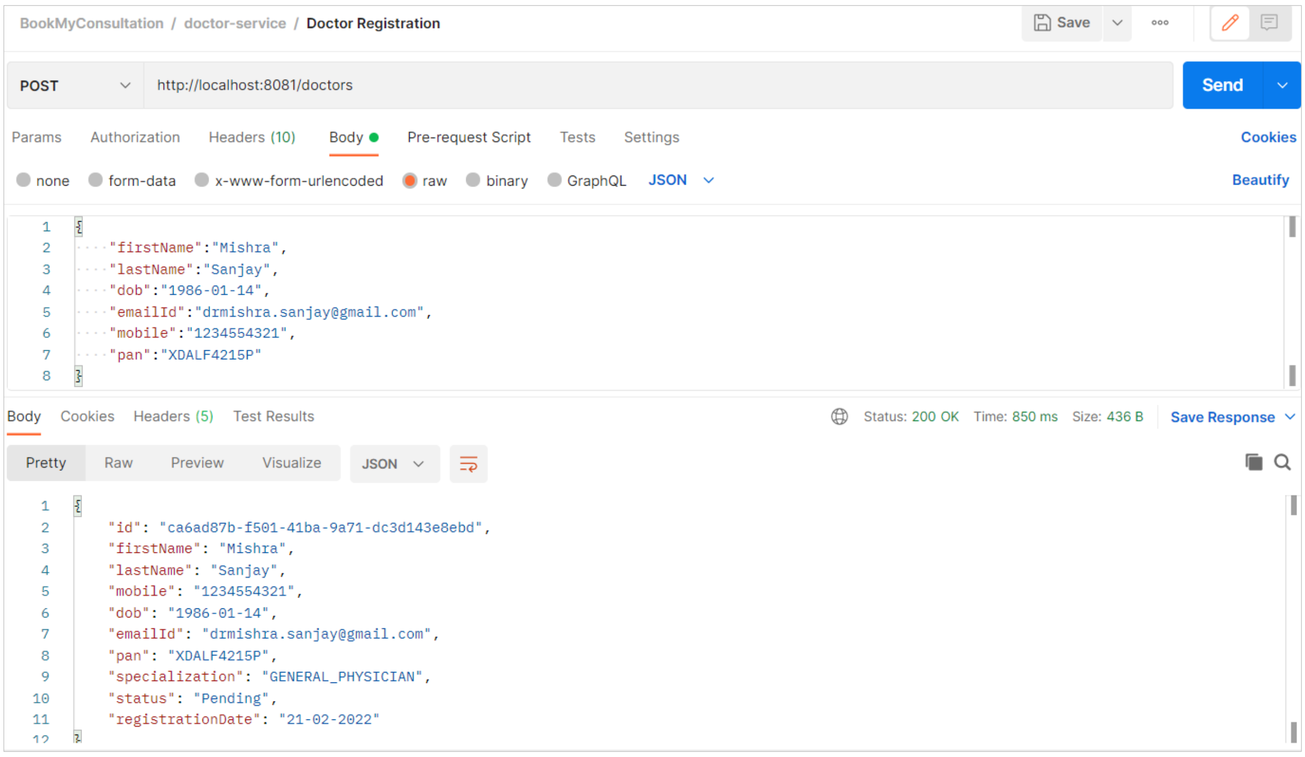
Request Body Ex –

```
{
    "firstName":"Mishra",
    "lastName":"Sanjay",
    "dob":"1986-01-14",
    "emailId":"drmishra.sanjay@gmail.com",
    "mobile":"1234554321",
    "pan":"XDALF4215P"
}
```

Response Body Ex –

```
{
    "id": "ca6ad87b-f501-41ba-9a71-dc3d143e8ebd",
    "firstName": "Mishra",
    "lastName": "Sanjay",
    "mobile": "1234554321",
    "dob": "1986-01-14",
    "emailId": "drmishra.sanjay@gmail.com",
    "pan": "XDALF4215P",
    "specialization": "GENERAL_PHYSICIAN",
    "status": "Pending",
    "registrationDate": "21-02-2022"
}
```

Calling the endpoint from POSTMAN -



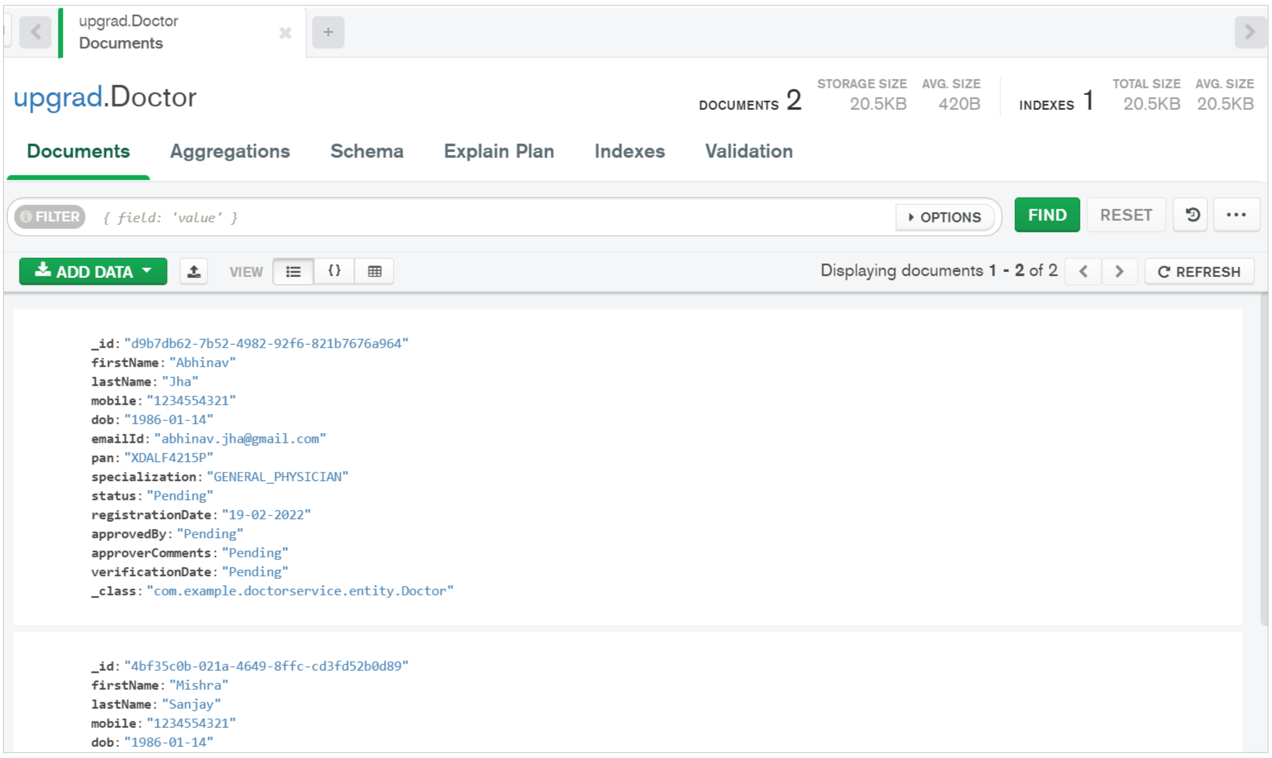
http://localhost:8081/doctors

```
{
    "firstName":"Mishra",
    "lastName":"Sanjay",
    "dob":"1986-01-14",
    "emailId":"drmishra.sanjay@gmail.com",
    "mobile":"1234554321",
    "pan":"XDALF4215P"
}
```

Status: 200 OK   Time: 850 ms   Size: 436 B

```
{
    "id": "ca6ad87b-f501-41ba-9a71-dc3d143e8ebd",
    "firstName": "Mishra",
    "lastName": "Sanjay",
    "mobile": "1234554321",
    "dob": "1986-01-14",
    "emailId": "drmishra.sanjay@gmail.com",
    "pan": "XDALF4215P",
    "specialization": "GENERAL_PHYSICIAN",
    "status": "Pending",
    "registrationDate": "21-02-2022"
}
```

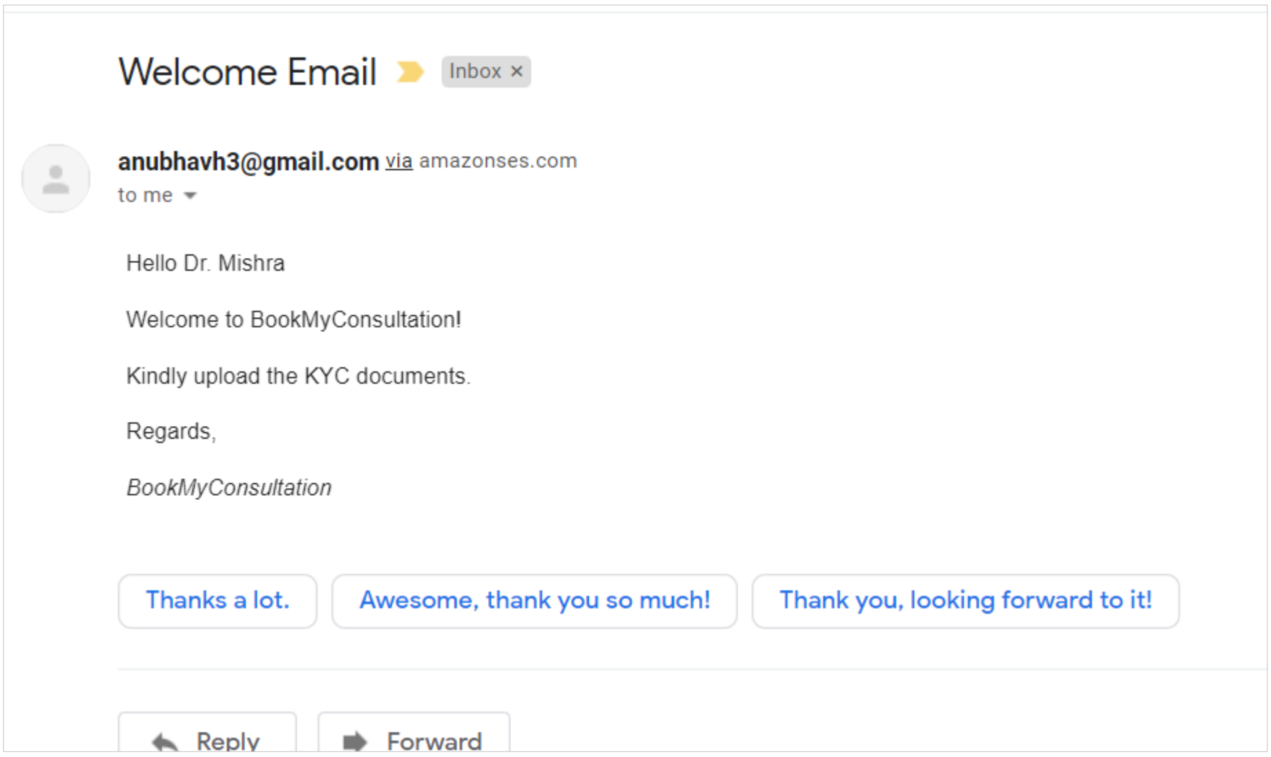
Mongo Collection 'Doctor' -



```
_id: "d9b7db62-7b52-4982-92f6-821b7676a964"
firstName: "Abhinav"
lastName: "Jha"
mobile: "1234554321"
dob: "1986-01-14"
emailId: "abhinav.jha@gmail.com"
pan: "XDALF4215P"
specialization: "GENERAL_PHYSICIAN"
status: "Pending"
registrationDate: "19-02-2022"
approvedBy: "Pending"
approverComments: "Pending"
verificationDate: "Pending"
_class: "com.example.doctorservice.entity.Doctor"

_id: "4bf35c0b-021a-4649-8ffc-cd3fd52b0d89"
firstName: "Mishra"
lastName: "Sanjay"
mobile: "1234554321"
dob: "1986-01-14"
```

Email received by the new doctor-


Welcome Email
Inbox ×
Thanks a lot.
Awesome, thank you so much!
Thank you, looking forward to it!
Reply
Forward

**anubhavh3@gmail.com** via amazonses.com
to me

Hello Dr. Mishra

Welcome to BookMyConsultation!

Kindly upload the KYC documents.

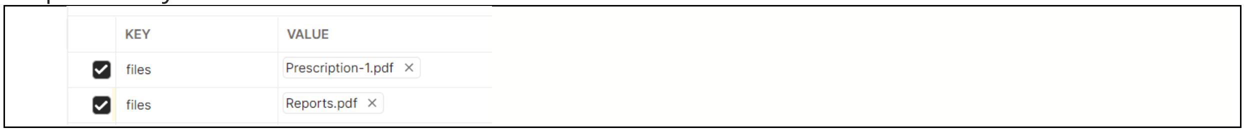
Regards,

*BookMyConsultation*

## 5.2. Endpoint-2: Upload doctor documents

POST localhost:8081/doctors/{doctorId}/document

Request Body Ex –

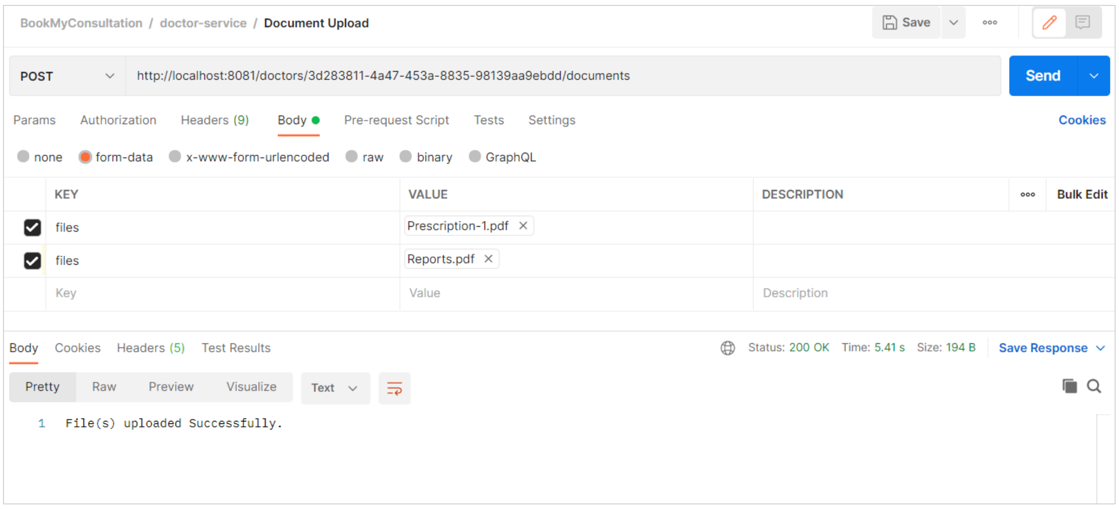


| | KEY | VALUE |
|---|---|---|
| ✓ | files | Prescription-1.pdf |
| ✓ | files | Reports.pdf |

Response Body Ex –

```
File(s) uploaded Successfully.
```

Calling the endpoint from POSTMAN -



BookMyConsultation / doctor-service / Document Upload

POST http://localhost:8081/doctors/3d283811-4a47-453a-8835-98139aa9ebdd/documents

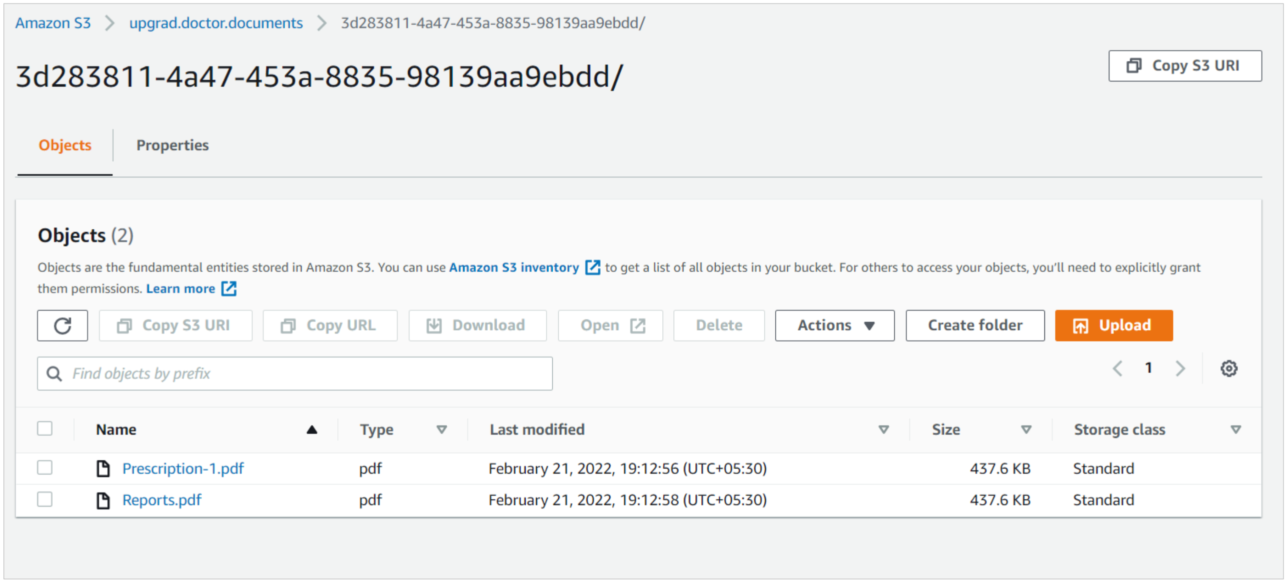
Body: form-data

| | KEY | VALUE | DESCRIPTION |
|---|---|---|---|
| ✓ | files | Prescription-1.pdf | |
| ✓ | files | Reports.pdf | |

Status: 200 OK Time: 5.41 s Size: 194 B

```
File(s) uploaded Successfully.
```

S3 bucket updated with the uploaded documents -



Amazon S3 > upgrad.doctor.documents > 3d283811-4a47-453a-8835-98139aa9ebdd/

3d283811-4a47-453a-8835-98139aa9ebdd/

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

| Name | Type | Last modified | Size | Storage class |
|---|---|---|---|---|
| Prescription-1.pdf | pdf | February 21, 2022, 19:12:56 (UTC+05:30) | 437.6 KB | Standard |
| Reports.pdf | pdf | February 21, 2022, 19:12:58 (UTC+05:30) | 437.6 KB | Standard |

## 5.3. Endpoint-3: Approve doctor registration

**PUT** localhost:8081/doctors/{doctorId}/approve

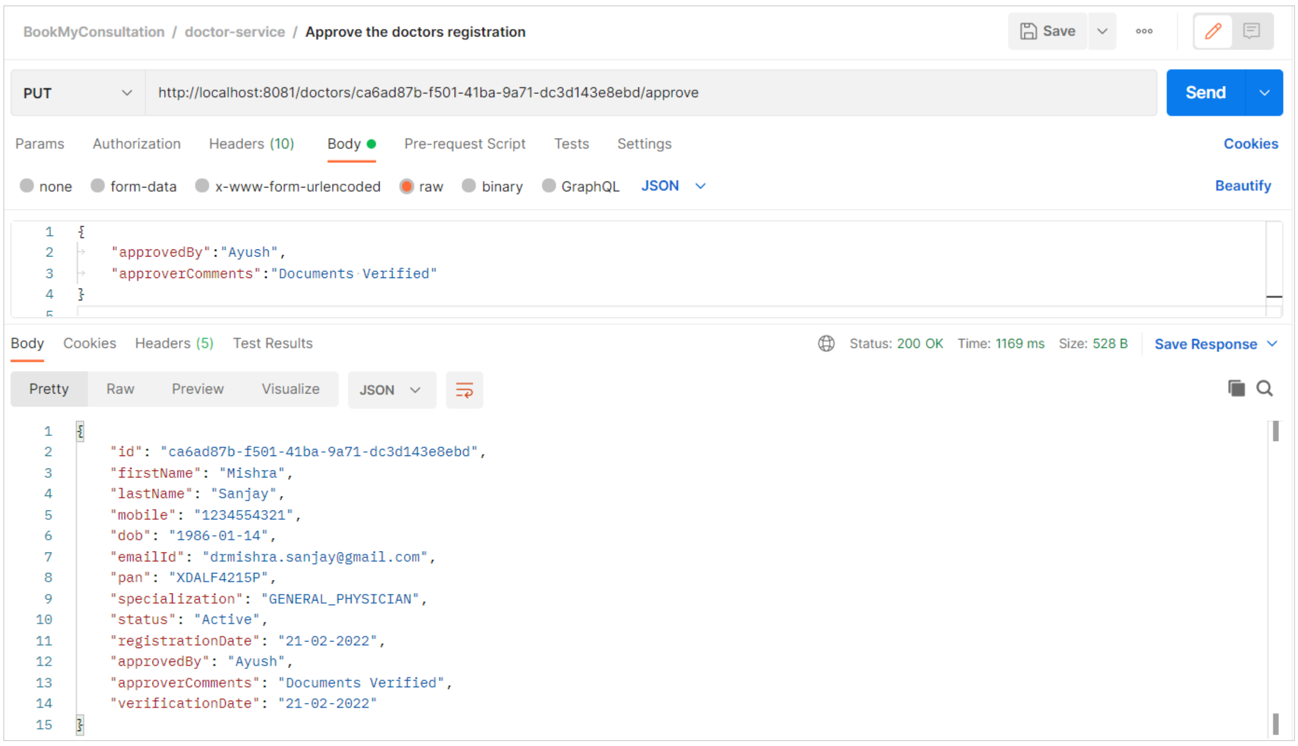
Request Body Ex –

```
{
    "approvedBy":"Ayush",
    "approverComments":"Documents Verified"
}
```

Response Body Ex –

```
{
    "id": "ca6ad87b-f501-41ba-9a71-dc3d143e8ebd",
    "firstName": "Mishra",
    "lastName": "Sanjay",
    "mobile": "1234554321",
    "dob": "1986-01-14",
    "emailId": "drmishra.sanjay@gmail.com",
    "pan": "XDALF4215P",
    "specialization": "GENERAL_PHYSICIAN",
    "status": "Active",
    "registrationDate": "21-02-2022",
    "approvedBy": "Ayush",
    "approverComments": "Documents Verified",
    "verificationDate": "21-02-2022"
}
```

Calling the endpoint from POSTMAN –



http://localhost:8081/doctors/ca6ad87b-f501-41ba-9a71-dc3d143e8ebd/approve

```
{
    "approvedBy":"Ayush",
    "approverComments":"Documents Verified"
}
```

Status: 200 OK Time: 1169 ms Size: 528 B

```
{
    "id": "ca6ad87b-f501-41ba-9a71-dc3d143e8ebd",
    "firstName": "Mishra",
    "lastName": "Sanjay",
    "mobile": "1234554321",
    "dob": "1986-01-14",
    "emailId": "drmishra.sanjay@gmail.com",
    "pan": "XDALF4215P",
    "specialization": "GENERAL_PHYSICIAN",
    "status": "Active",
    "registrationDate": "21-02-2022",
    "approvedBy": "Ayush",
    "approverComments": "Documents Verified",
    "verificationDate": "21-02-2022"
}
```

## 5.4. Endpoint-4: Reject doctor registration

**PUT** localhost:8081/doctors/{doctorId}/reject

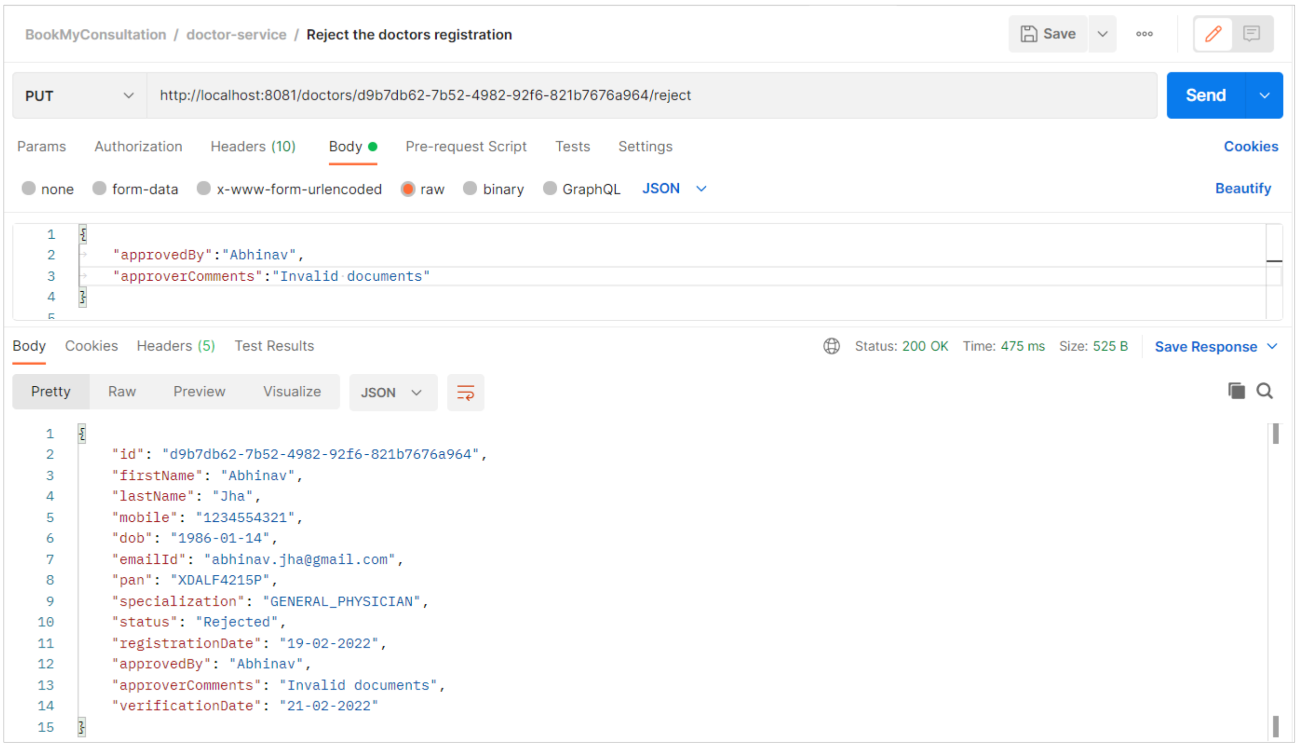
Request Body Ex –

```
{
    "approvedBy":"Abhi",
    "approverComments":"Invalid documents"
}
```

Response Body Ex –

```
{
    "id": "d9b7db62-7b52-4982-92f6-821b7676a964",
    "firstName": "Abhinav",
    "lastName": "Jha",
    "mobile": "1234554321",
    "dob": "1986-01-14",
    "emailId": "abhinav.jha@gmail.com",
    "pan": "XDALF4215P",
    "specialization": "GENERAL_PHYSICIAN",
    "status": "Rejected",
    "registrationDate": "19-02-2022",
    "approvedBy": "Abhi",
    "approverComments": "Invalid documents",
    "verificationDate": "21-02-2022"
}
```

Calling the endpoint from POSTMAN –



BookMyConsultation / doctor-service / Reject the doctors registration

PUT http://localhost:8081/doctors/d9b7db62-7b52-4982-92f6-821b7676a964/reject

```
{
    "approvedBy":"Abhinav",
    "approverComments":"Invalid documents"
}
```

Status: 200 OK Time: 475 ms Size: 525 B

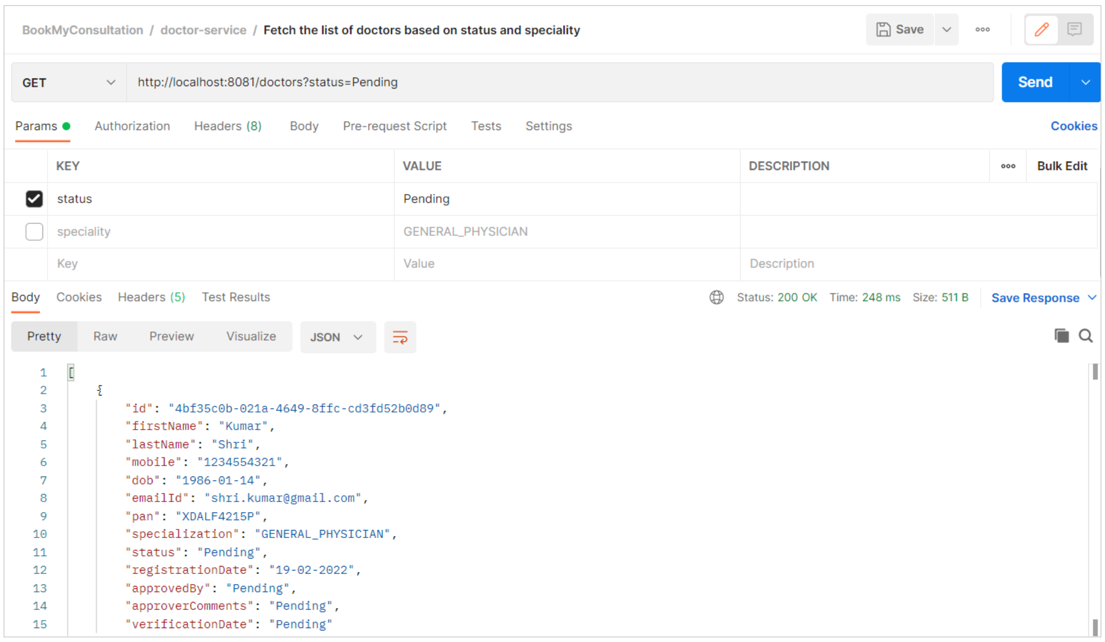
```
{
    "id": "d9b7db62-7b52-4982-92f6-821b7676a964",
    "firstName": "Abhinav",
    "lastName": "Jha",
    "mobile": "1234554321",
    "dob": "1986-01-14",
    "emailId": "abhinav.jha@gmail.com",
    "pan": "XDALF4215P",
    "specialization": "GENERAL_PHYSICIAN",
    "status": "Rejected",
    "registrationDate": "19-02-2022",
    "approvedBy": "Abhinav",
    "approverComments": "Invalid documents",
    "verificationDate": "21-02-2022"
}
```

## 5.5. Endpoint-5: Return list of doctors

### 5.5.1. Scenario-1: The status is pending

**GET** localhost:8081/doctors?status=Pending

Calling the endpoint from POSTMAN –



GET http://localhost:8081/doctors?status=Pending

| | KEY | VALUE | DESCRIPTION |
|---|---|---|---|
| ✓ | status | Pending | |
| | speciality | GENERAL_PHYSICIAN | |
| | Key | Value | Description |

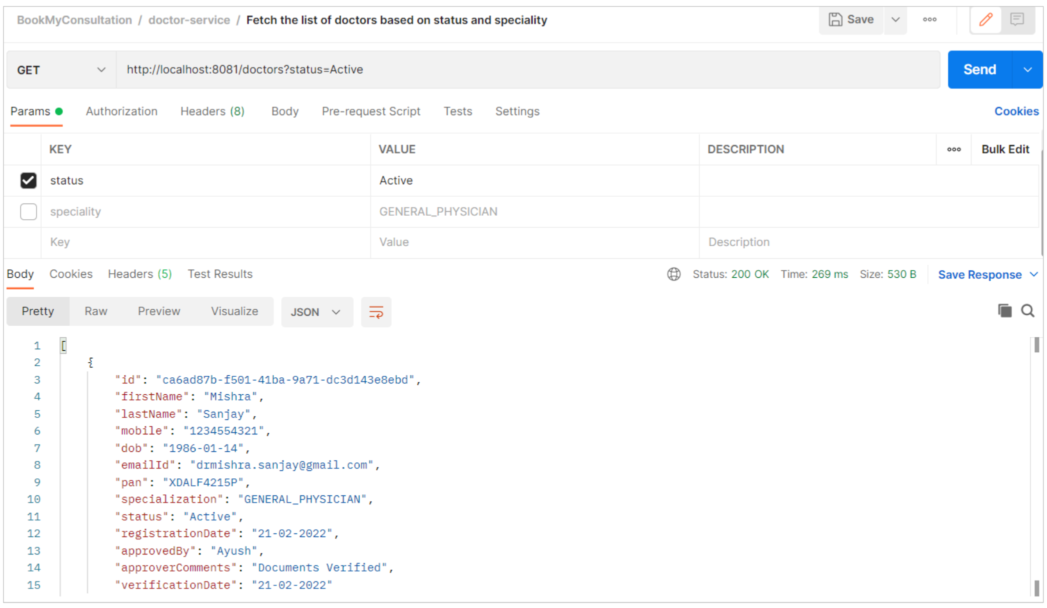
Status: 200 OK Time: 248 ms Size: 511 B

```
[
    {
        "id": "4bf35c0b-021a-4649-8ffc-cd3fd52b0d89",
        "firstName": "Kumar",
        "lastName": "Shri",
        "mobile": "1234554321",
        "dob": "1986-01-14",
        "emailId": "shri.kumar@gmail.com",
        "pan": "XDALF4215P",
        "specialization": "GENERAL_PHYSICIAN",
        "status": "Pending",
        "registrationDate": "19-02-2022",
        "approvedBy": "Pending",
        "approverComments": "Pending",
        "verificationDate": "Pending"
```

### 5.5.2. Scenario-2: The status is Active

**GET** localhost:8081/doctors?status=Active

Calling the endpoint from POSTMAN –



GET http://localhost:8081/doctors?status=Active

| | KEY | VALUE | DESCRIPTION |
|---|---|---|---|
| ✓ | status | Active | |
| | speciality | GENERAL_PHYSICIAN | |
| | Key | Value | Description |

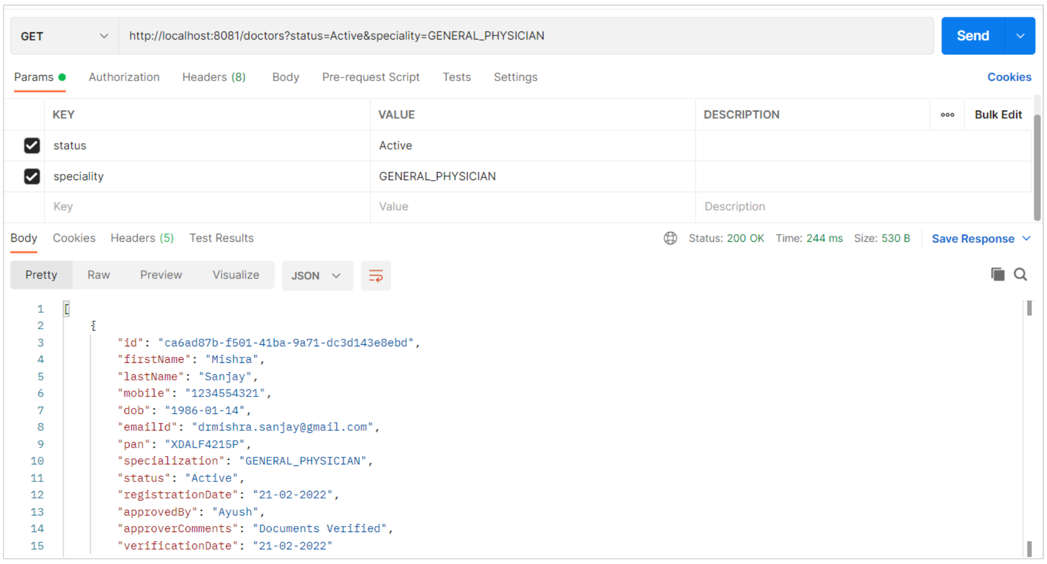
Status: 200 OK Time: 269 ms Size: 530 B

```
[
    {
        "id": "ca6ad87b-f501-41ba-9a71-dc3d143e8ebd",
        "firstName": "Mishra",
        "lastName": "Sanjay",
        "mobile": "1234554321",
        "dob": "1986-01-14",
        "emailId": "drmishra.sanjay@gmail.com",
        "pan": "XDALF4215P",
        "specialization": "GENERAL_PHYSICIAN",
        "status": "Active",
        "registrationDate": "21-02-2022",
        "approvedBy": "Ayush",
        "approverComments": "Documents Verified",
        "verificationDate": "21-02-2022"
```

### 5.5.3. Scenario-2: Based on speciality

**GET** localhost:8081/status=Active&speciality=GENERAL_PHYSICIAN

Calling the endpoint from POSTMAN –



GET http://localhost:8081/doctors?status=Active&speciality=GENERAL_PHYSICIAN

| KEY | VALUE | DESCRIPTION |
|---|---|---|
| status | Active | |
| speciality | GENERAL_PHYSICIAN | |

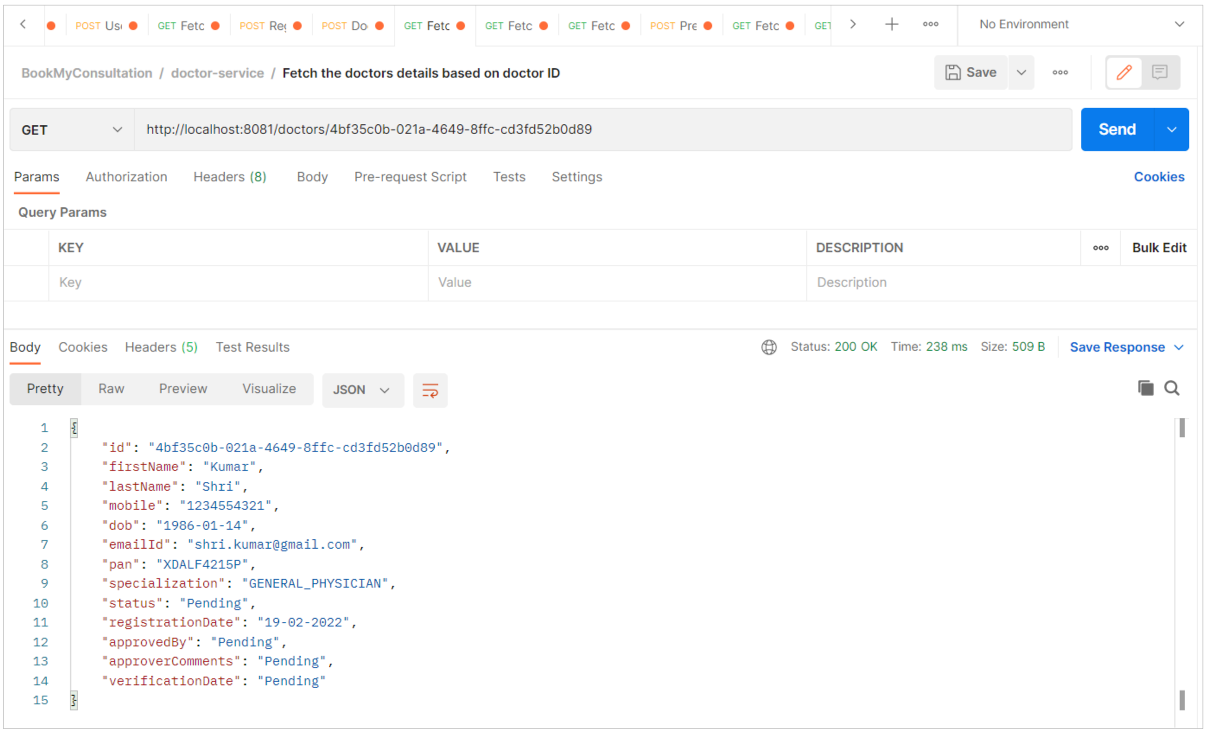
Status: 200 OK Time: 244 ms Size: 530 B

```
[
    {
        "id": "ca6ad87b-f501-41ba-9a71-dc3d143e8ebd",
        "firstName": "Mishra",
        "lastName": "Sanjay",
        "mobile": "1234554321",
        "dob": "1986-01-14",
        "emailId": "drmishra.sanjay@gmail.com",
        "pan": "XDALF4215P",
        "specialization": "GENERAL_PHYSICIAN",
        "status": "Active",
        "registrationDate": "21-02-2022",
        "approvedBy": "Ayush",
        "approverComments": "Documents Verified",
        "verificationDate": "21-02-2022"
```

## 5.6. Endpoint-6: Return doctors based on doctor-ID

**GET** localhost:8081/ /doctors/{doctorId}

Calling the endpoint from POSTMAN –



BookMyConsultation / doctor-service / Fetch the doctors details based on doctor ID

GET http://localhost:8081/doctors/4bf35c0b-021a-4649-8ffc-cd3fd52b0d89

Status: 200 OK Time: 238 ms Size: 509 B

```
{
    "id": "4bf35c0b-021a-4649-8ffc-cd3fd52b0d89",
    "firstName": "Kumar",
    "lastName": "Shri",
    "mobile": "1234554321",
    "dob": "1986-01-14",
    "emailId": "shri.kumar@gmail.com",
    "pan": "XDALF4215P",
    "specialization": "GENERAL_PHYSICIAN",
    "status": "Pending",
    "registrationDate": "19-02-2022",
    "approvedBy": "Pending",
    "approverComments": "Pending",
    "verificationDate": "Pending"
}
```

## 6. User-Onboarding Service

### 6.1. Endpoint-1: Collect user information

POST localhost:8082/users

Content-Type application/json

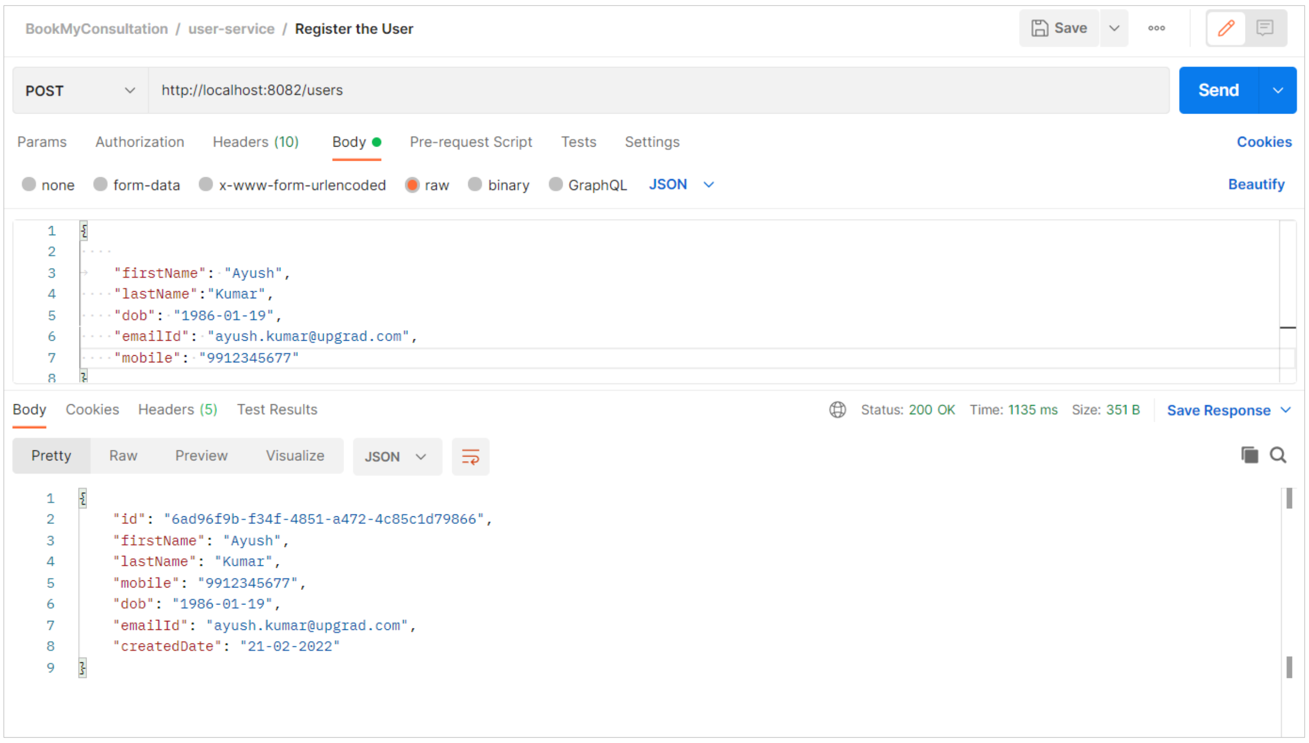
Request Body Ex –

```
{
     "firstName": "Ayush",
     "lastName":"Kumar",
     "dob": "1986-01-19",
     "emailId": "ayush.kumar@upgrad.com",
     "mobile": "9912345677"
}
```

Response Body Ex –

```
{
     "id": "6ad96f9b-f34f-4851-a472-4c85c1d79866",
     "firstName": "Ayush",
     "lastName": "Kumar",
     "mobile": "9912345677",
     "dob": "1986-01-19",
     "emailId": "ayush.kumar@upgrad.com",
     "createdDate": "21-02-2022"
}
```

Calling the endpoint from POSTMAN -



BookMyConsultation / user-service / Register the User

POST http://localhost:8082/users

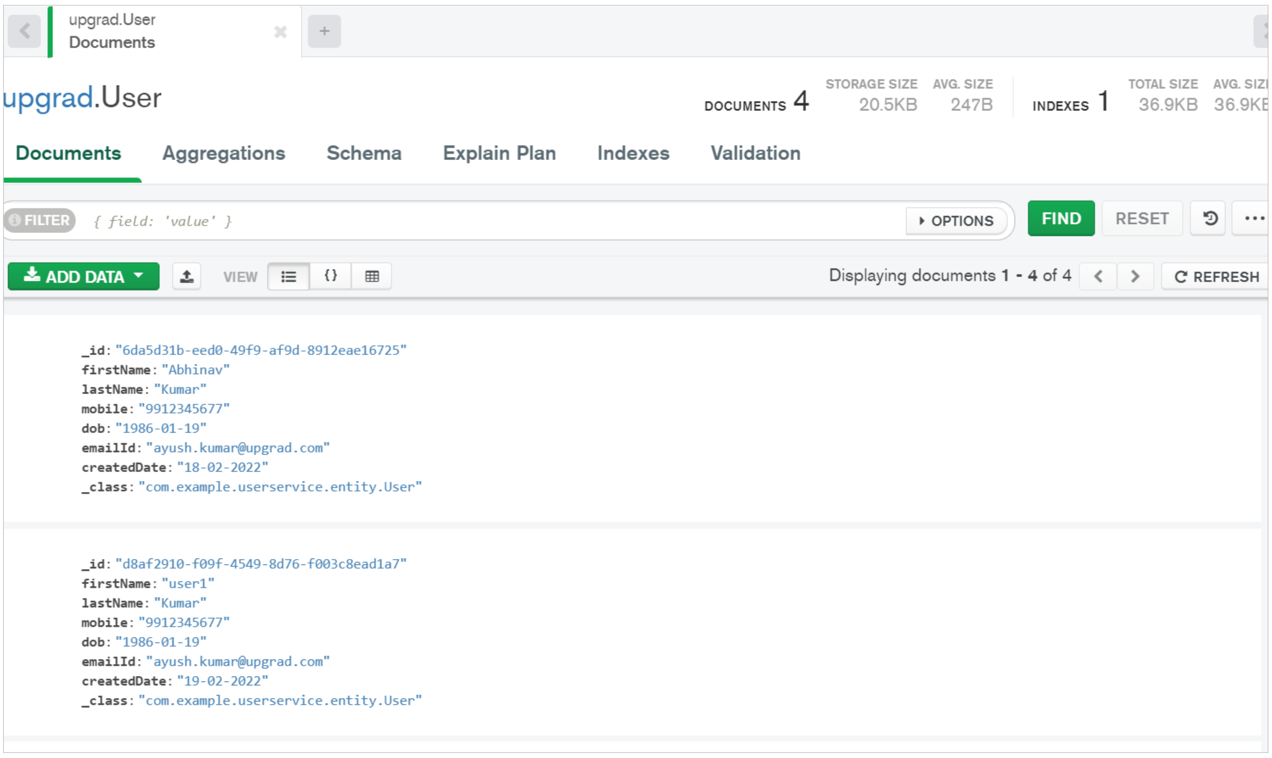
```
{

    "firstName": "Ayush",
    "lastName":"Kumar",
    "dob": "1986-01-19",
    "emailId": "ayush.kumar@upgrad.com",
    "mobile": "9912345677"
}
```

Status: 200 OK Time: 1135 ms Size: 351 B

```
{
    "id": "6ad96f9b-f34f-4851-a472-4c85c1d79866",
    "firstName": "Ayush",
    "lastName": "Kumar",
    "mobile": "9912345677",
    "dob": "1986-01-19",
    "emailId": "ayush.kumar@upgrad.com",
    "createdDate": "21-02-2022"
}
```

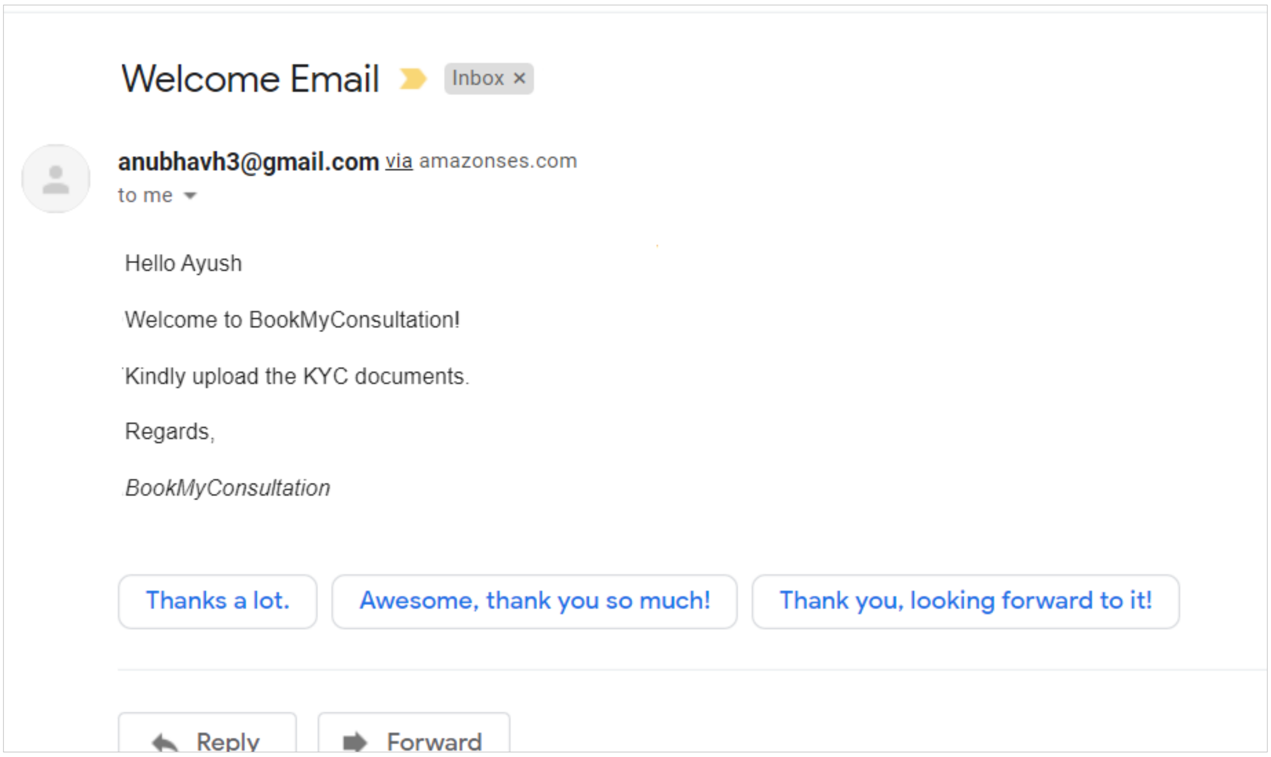
Mongo Collection 'User' -



```
_id: "6da5d31b-eed0-49f9-af9d-8912eae16725"
firstName: "Abhinav"
lastName: "Kumar"
mobile: "9912345677"
dob: "1986-01-19"
emailId: "ayush.kumar@upgrad.com"
createdDate: "18-02-2022"
_class: "com.example.userservice.entity.User"

_id: "d8af2910-f09f-4549-8d76-f003c8ead1a7"
firstName: "user1"
lastName: "Kumar"
mobile: "9912345677"
dob: "1986-01-19"
emailId: "ayush.kumar@upgrad.com"
createdDate: "19-02-2022"
_class: "com.example.userservice.entity.User"
```

Email received by the new user-


Welcome Email
Inbox ×
Thanks a lot.
Awesome, thank you so much!
Thank you, looking forward to it!
Reply
Forward

**anubhavh3@gmail.com** via amazonses.com
to me

Hello Ayush

Welcome to BookMyConsultation!

Kindly upload the KYC documents.

Regards,

*BookMyConsultation*

## 6.2. Endpoint-2: Fetch user information

**POST** localhost:8082/users/{userID}

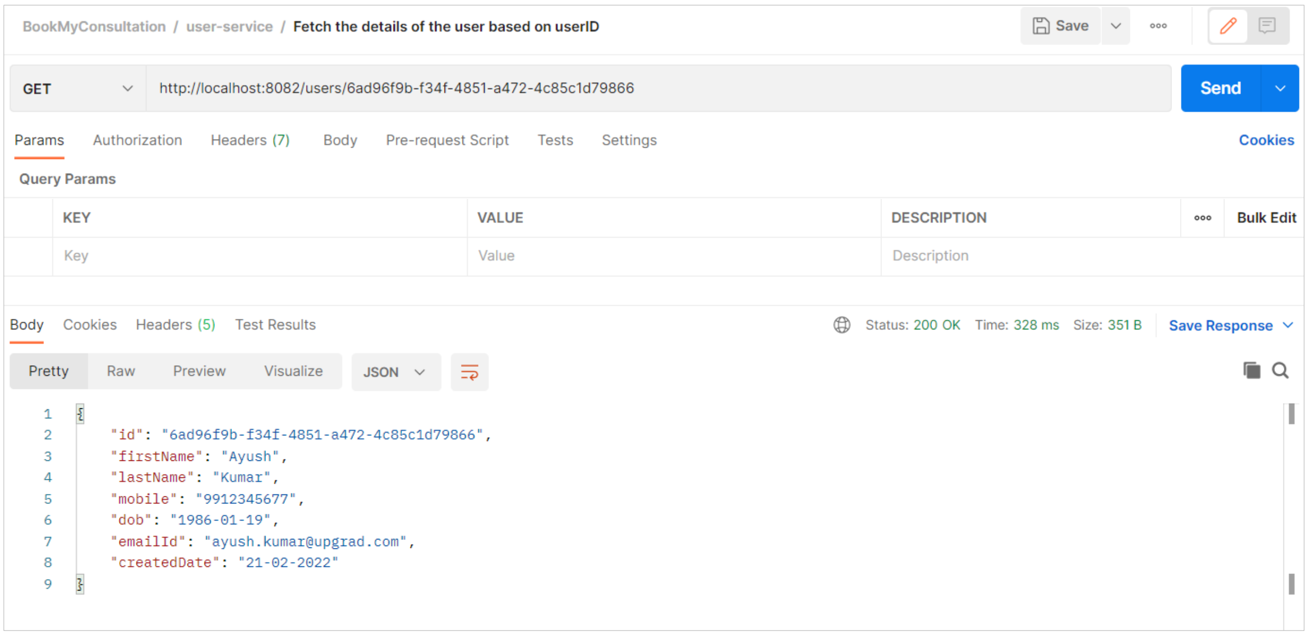
Request Ex –

```
http://localhost:8082/users/6da5d31b-eed0-49f9-af9d-8912eae16725
```

Response Body Ex –

```
{
    "id": "6da5d31b-eed0-49f9-af9d-8912eae16725",
    "firstName": "Abhinav",
    "lastName": "Kumar",
    "mobile": "9912345677",
    "dob": "1986-01-19",
    "emailId": "ayush.kumar@upgrad.com",
    "createdDate": "18-02-2022"
}
```

Calling the endpoint from POSTMAN -



GET http://localhost:8082/users/6ad96f9b-f34f-4851-a472-4c85c1d79866

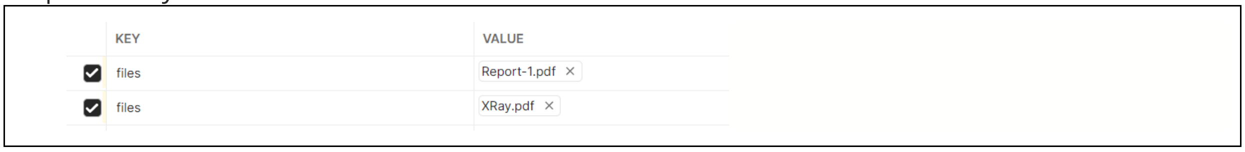
Status: 200 OK Time: 328 ms Size: 351 B

```
{
    "id": "6ad96f9b-f34f-4851-a472-4c85c1d79866",
    "firstName": "Ayush",
    "lastName": "Kumar",
    "mobile": "9912345677",
    "dob": "1986-01-19",
    "emailId": "ayush.kumar@upgrad.com",
    "createdDate": "21-02-2022"
}
```

## 6.3. Endpoint-3: Upload user documents

POST localhost:8082/users/{userId}/document

Request Body Ex –

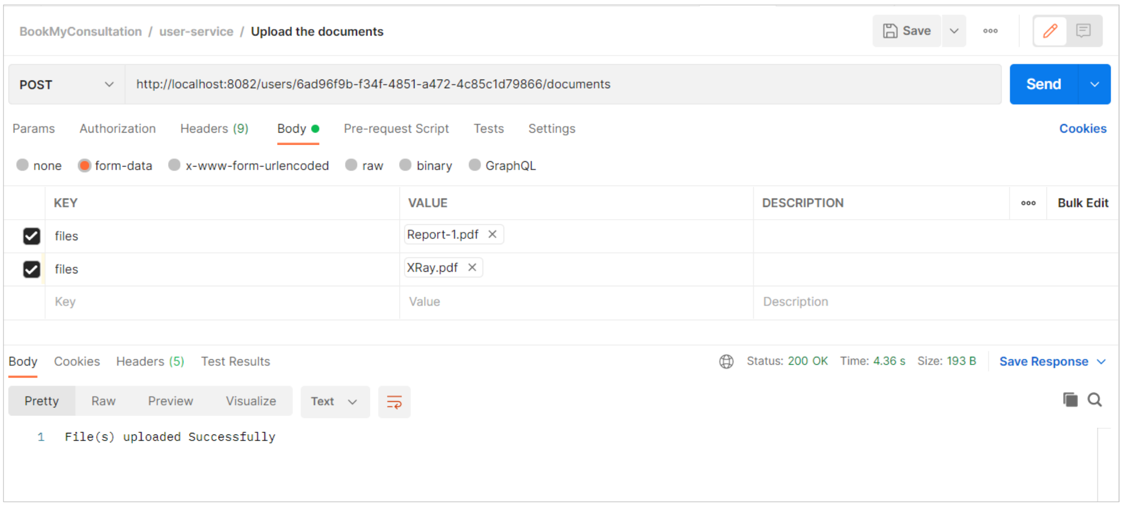


| KEY | VALUE |
|---|---|
| ✓ files | Report-1.pdf |
| ✓ files | XRay.pdf |

Response Body Ex –

```
    File(s) uploaded Successfully.
```

Calling the endpoint from POSTMAN -



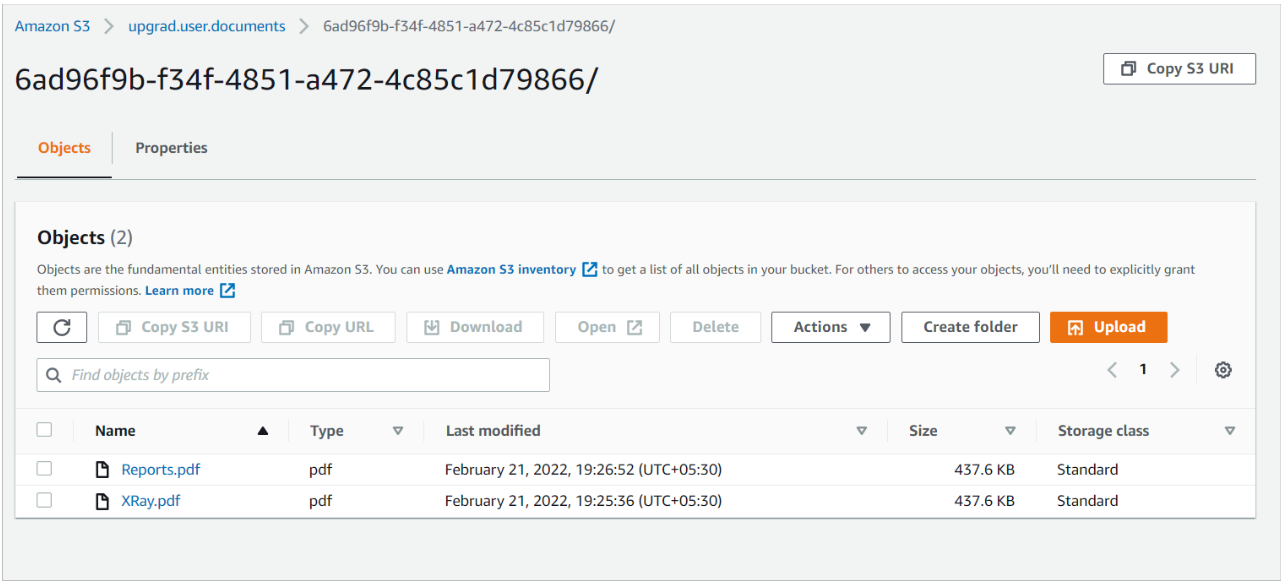
http://localhost:8082/users/6ad96f9b-f34f-4851-a472-4c85c1d79866/documents

| KEY | VALUE | DESCRIPTION |
|---|---|---|
| ✓ files | Report-1.pdf | |
| ✓ files | XRay.pdf | |
| Key | Value | Description |

Status: 200 OK Time: 4.36 s Size: 193 B

```
File(s) uploaded Successfully
```

S3 bucket updated with the uploaded documents -



6ad96f9b-f34f-4851-a472-4c85c1d79866/

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

| Name | Type | Last modified | Size | Storage class |
|---|---|---|---|---|
| Reports.pdf | pdf | February 21, 2022, 19:26:52 (UTC+05:30) | 437.6 KB | Standard |
| XRay.pdf | pdf | February 21, 2022, 19:25:36 (UTC+05:30) | 437.6 KB | Standard |

# 7. Appointment Service

## 7.1. Endpoint-1: Update availability of the doctors.

**POST** localhost:8083/doctor/{doctorId}/availability

Content-Type application/json

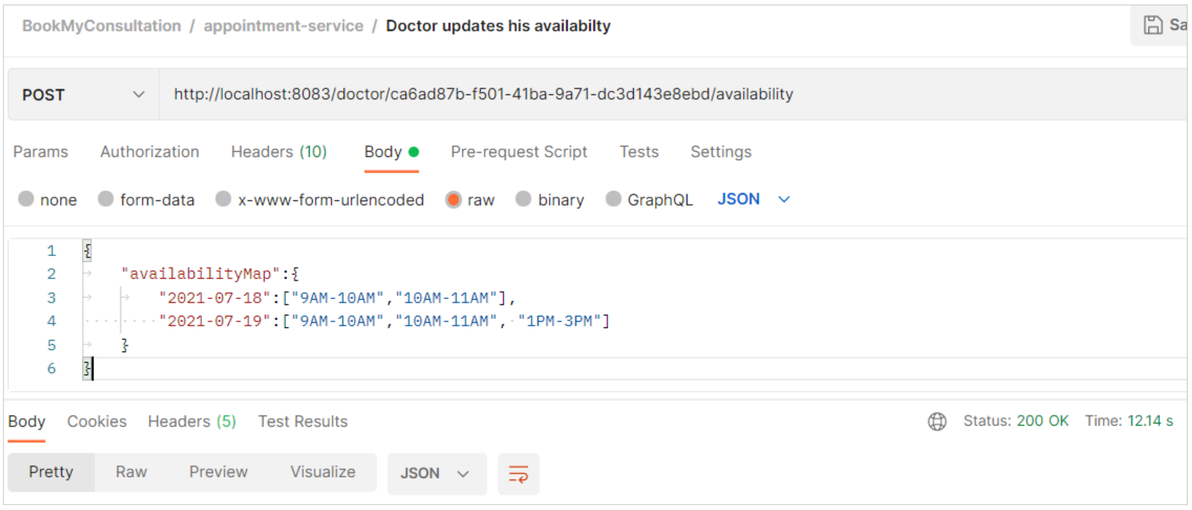
Request Body Ex –

```
{
    "availabilityMap":{
        "2021-07-18":["9AM-10AM","10AM-11AM"],
        "2021-07-19":["9AM-10AM","10AM-11AM", "1PM-3PM"]
    }
}
```

Response -

```
HTTP Status 200 OK
```

Calling the endpoint from POSTMAN -

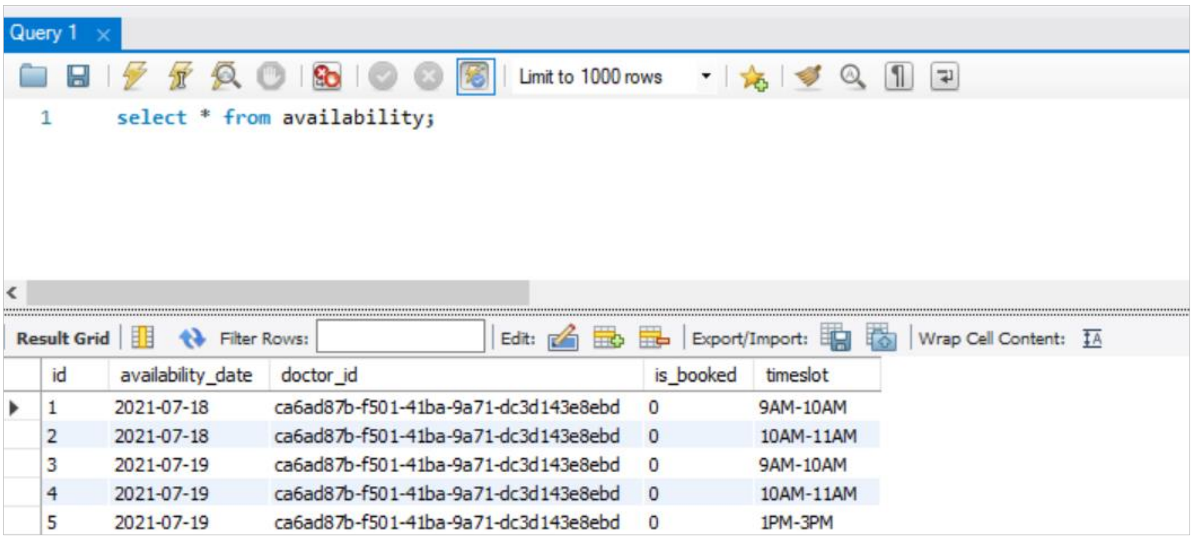


POST http://localhost:8083/doctor/ca6ad87b-f501-41ba-9a71-dc3d143e8ebd/availability

```
{
    "availabilityMap":{
        "2021-07-18":["9AM-10AM","10AM-11AM"],
        "2021-07-19":["9AM-10AM","10AM-11AM", "1PM-3PM"]
    }
}
```

Status: 200 OK Time: 12.14 s

Records stored in RDS –



```
select * from availability;
```

| id | availability_date | doctor_id | is_booked | timeslot |
|---|---|---|---|---|
| 1 | 2021-07-18 | ca6ad87b-f501-41ba-9a71-dc3d143e8ebd | 0 | 9AM-10AM |
| 2 | 2021-07-18 | ca6ad87b-f501-41ba-9a71-dc3d143e8ebd | 0 | 10AM-11AM |
| 3 | 2021-07-19 | ca6ad87b-f501-41ba-9a71-dc3d143e8ebd | 0 | 9AM-10AM |
| 4 | 2021-07-19 | ca6ad87b-f501-41ba-9a71-dc3d143e8ebd | 0 | 10AM-11AM |
| 5 | 2021-07-19 | ca6ad87b-f501-41ba-9a71-dc3d143e8ebd | 0 | 1PM-3PM |

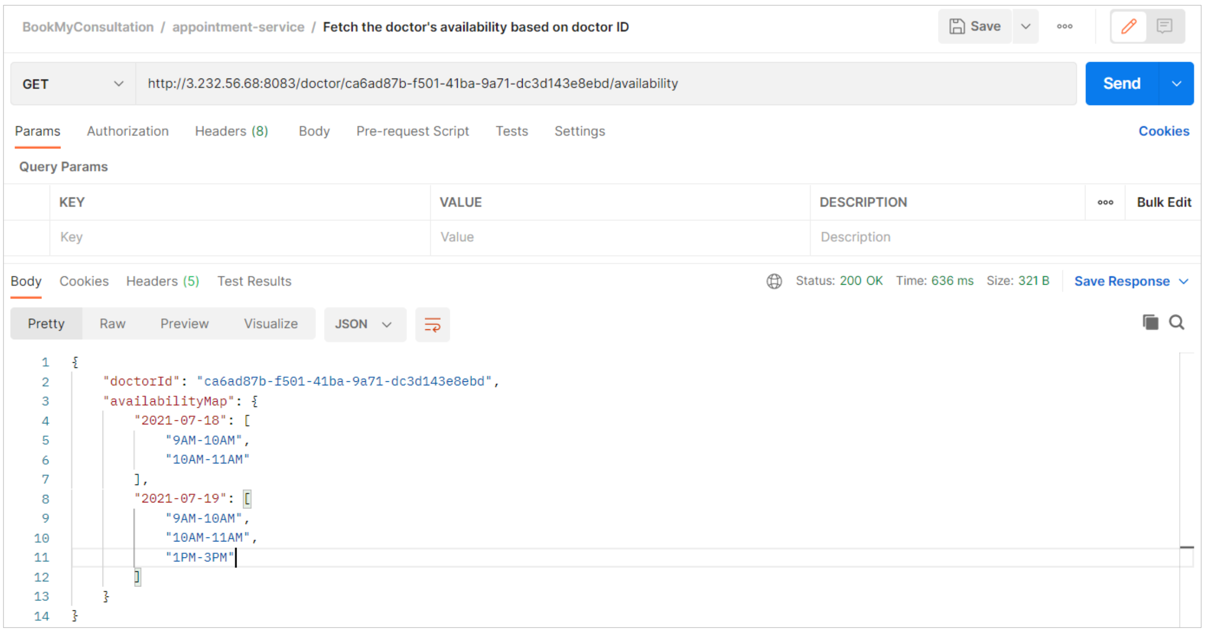
## 7.2. Endpoint-2: Fetch doctor's availability.

GET localhost:8083/doctor/{doctorId}/availability

Response Ex -

```
{
    "doctorId": "ca6ad87b-f501-41ba-9a71-dc3d143e8ebd",
    "availabilityMap": {
        "2021-07-18": [
            "9AM-10AM",
            "10AM-11AM"
        ],
        "2021-07-19": [
            "9AM-10AM",
            "10AM-11AM",
            "1PM-3PM"
        ]
    }
}
```

Calling the endpoint from POSTMAN -



http://3.232.56.68:8083/doctor/ca6ad87b-f501-41ba-9a71-dc3d143e8ebd/availability

Status: 200 OK Time: 636 ms Size: 321 B

```
{
    "doctorId": "ca6ad87b-f501-41ba-9a71-dc3d143e8ebd",
    "availabilityMap": {
        "2021-07-18": [
            "9AM-10AM",
            "10AM-11AM"
        ],
        "2021-07-19": [
            "9AM-10AM",
            "10AM-11AM",
            "1PM-3PM"
        ]
    }
}
```

## 7.3. Endpoint-3: Book Appointment

POST localhost:8083/appointments

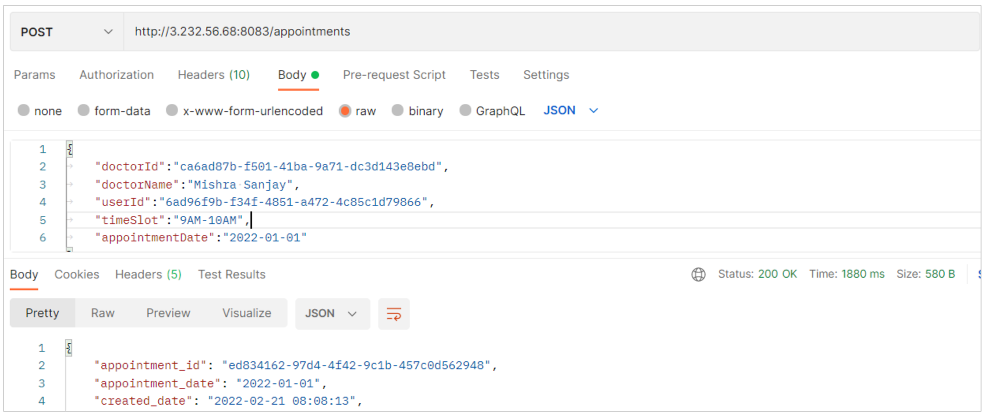
Request Body Ex –

```
{
    "doctorId":"4bf35c0b-021a-4649-8ffc-cd3fd52b0d89",
    "doctorName":"Kumar Shri",
    "userId":"6ad96f9b-f34f-4851-a472-4c85c1d79866",
    "timeSlot":"01PM-02PM",
    "appointmentDate":"2022-04-02"
}
```

Response Ex -

```
{
    "appointment_id": "977e6645-bd76-4a15-87d0-f5dc1fd76122",
    "appointment_date": "2022-04-02",
    "created_date": "2022-02-21 08:23:14",
    "doctor_id": "4bf35c0b-021a-4649-8ffc-cd3fd52b0d89",
    "prior_medical_history": null,
    "status": "PAYMENT_PENDING",
    "symptoms": null,
    "timeslot": "01PM-02PM",
    "userid": "6ad96f9b-f34f-4851-a472-4c85c1d79866",
    "user_email_id": "ayush.kumar@upgrad.com",
    "user_name": "Ayush Kumar",
    "doctor_name": "Kumar Shri"
}
```

Calling the endpoint from POSTMAN -



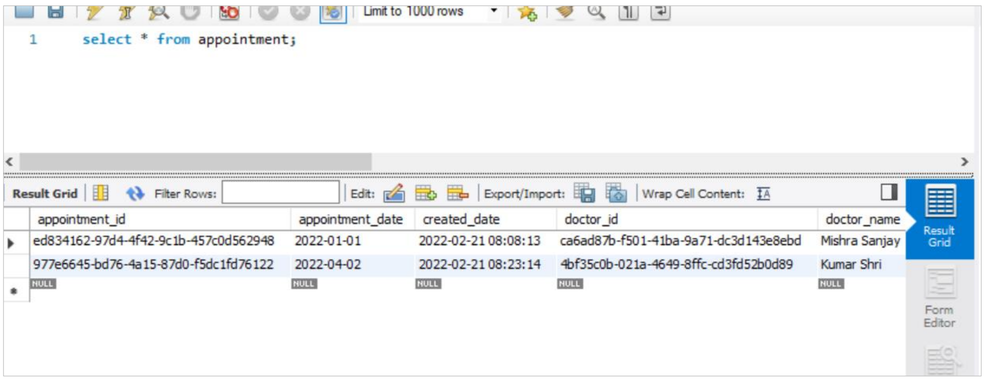
POST http://3.232.56.68:8083/appointments

```
{
  "doctorId":"ca6ad87b-f501-41ba-9a71-dc3d143e8ebd",
  "doctorName":"Mishra Sanjay",
  "userId":"6ad96f9b-f34f-4851-a472-4c85c1d79866",
  "timeSlot":"9AM-10AM",
  "appointmentDate":"2022-01-01"
```

Status: 200 OK Time: 1880 ms Size: 580 B

```
{
  "appointment_id": "ed834162-97d4-4f42-9c1b-457c0d562948",
  "appointment_date": "2022-01-01",
  "created_date": "2022-02-21 08:08:13",
```

Records stored in RDS –



```
select * from appointment;
```

| | appointment_id | appointment_date | created_date | doctor_id | doctor_name |
|---|---|---|---|---|---|
| ▶ | ed834162-97d4-4f42-9c1b-457c0d562948 | 2022-01-01 | 2022-02-21 08:08:13 | ca6ad87b-f501-41ba-9a71-dc3d143e8ebd | Mishra Sanjay |
| | 977e6645-bd76-4a15-87d0-f5dc1fd76122 | 2022-04-02 | 2022-02-21 08:23:14 | 4bf35c0b-021a-4649-8ffc-cd3fd52b0d89 | Kumar Shri |
| * | NULL | NULL | NULL | NULL | NULL |

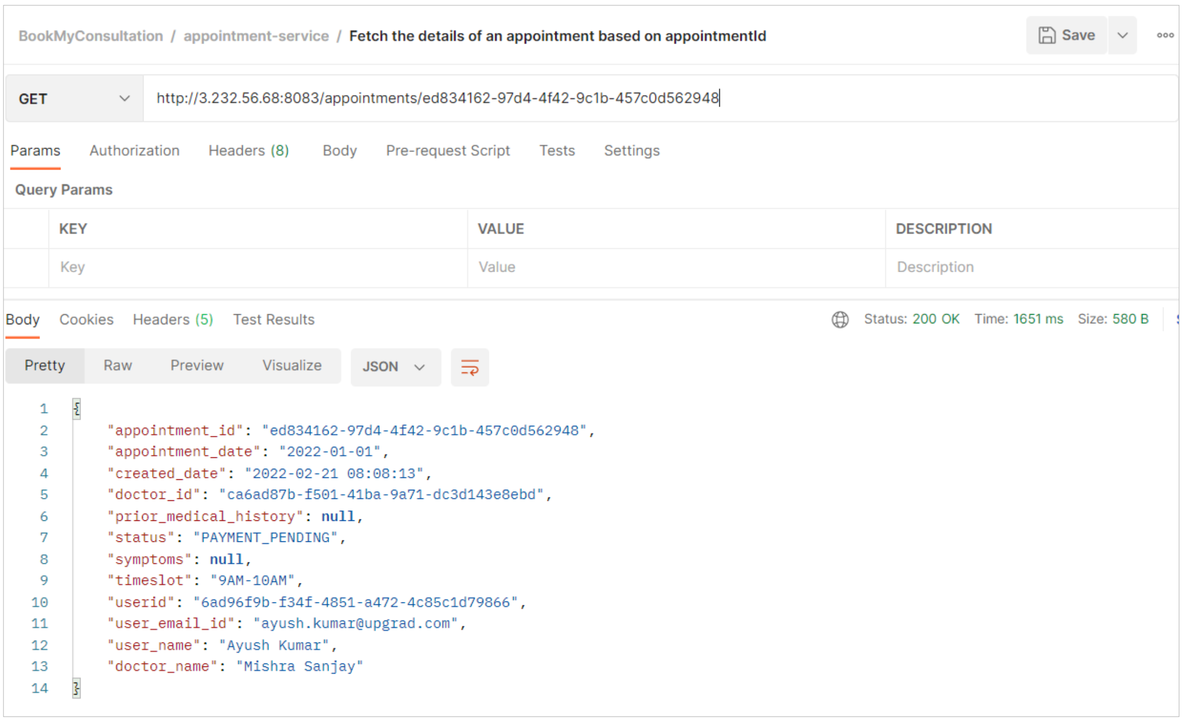
## 7.4. Endpoint-4: Fetch Appointment details

GET localhost:8083 /appointments/{appointmentId}

Response Ex -

```
{
    "appointment_id": "ed834162-97d4-4f42-9c1b-457c0d562948",
    "appointment_date": "2022-01-01",
    "created_date": "2022-02-21 08:08:13",
    "doctor_id": "ca6ad87b-f501-41ba-9a71-dc3d143e8ebd",
    "prior_medical_history": null,
    "status": "PAYMENT_PENDING",
    "symptoms": null,
    "timeslot": "9AM-10AM",
    "userid": "6ad96f9b-f34f-4851-a472-4c85c1d79866",
    "user_email_id": "ayush.kumar@upgrad.com",
    "user_name": "Ayush Kumar",
    "doctor_name": "Mishra Sanjay"
}
```

Calling the endpoint from POSTMAN -



http://3.232.56.68:8083/appointments/ed834162-97d4-4f42-9c1b-457c0d562948

Status: 200 OK Time: 1651 ms Size: 580 B

```
{
    "appointment_id": "ed834162-97d4-4f42-9c1b-457c0d562948",
    "appointment_date": "2022-01-01",
    "created_date": "2022-02-21 08:08:13",
    "doctor_id": "ca6ad87b-f501-41ba-9a71-dc3d143e8ebd",
    "prior_medical_history": null,
    "status": "PAYMENT_PENDING",
    "symptoms": null,
    "timeslot": "9AM-10AM",
    "userid": "6ad96f9b-f34f-4851-a472-4c85c1d79866",
    "user_email_id": "ayush.kumar@upgrad.com",
    "user_name": "Ayush Kumar",
    "doctor_name": "Mishra Sanjay"
}
```

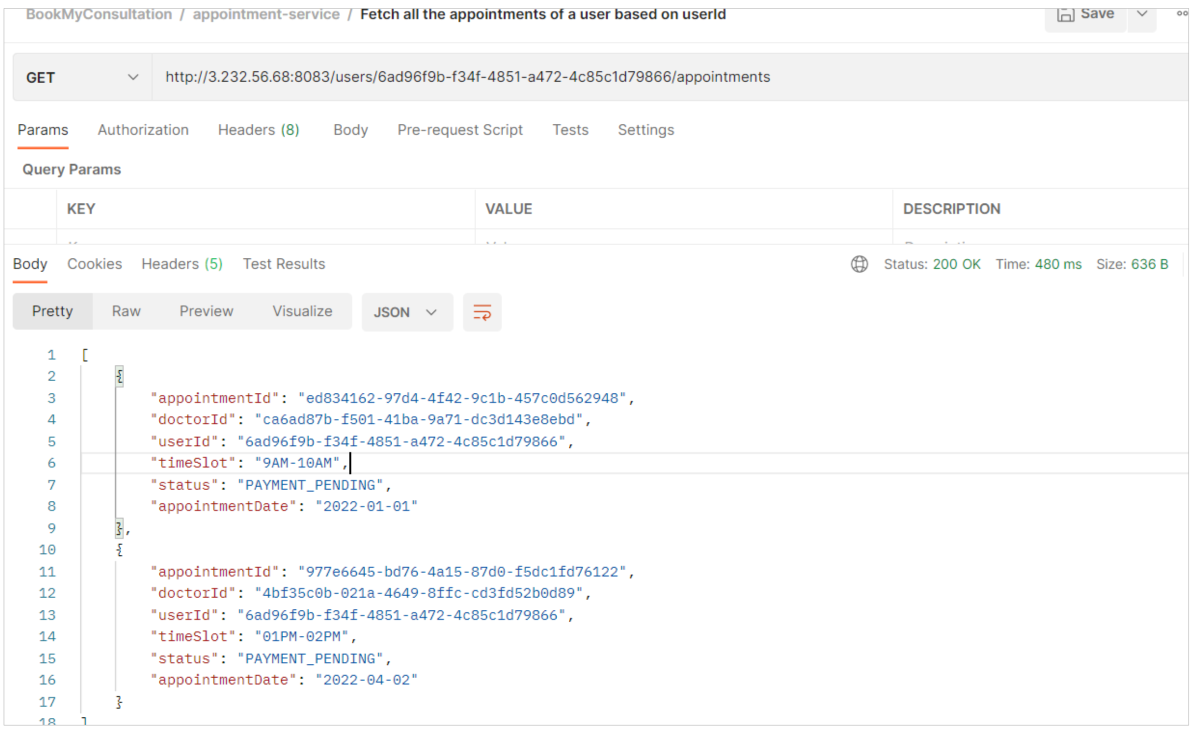
## 7.5. Endpoint-5: Fetch Appointments by userId

**GET** localhost:8083 /users/{userId}/appointments

Response Ex -

```
[
    {
        "appointmentId": "ed834162-97d4-4f42-9c1b-457c0d562948",
        "doctorId": "ca6ad87b-f501-41ba-9a71-dc3d143e8ebd",
        "userId": "6ad96f9b-f34f-4851-a472-4c85c1d79866",
        "timeSlot": "9AM-10AM",
        "status": "PAYMENT_PENDING",
        "appointmentDate": "2022-01-01"
    },
    {
        "appointmentId": "977e6645-bd76-4a15-87d0-f5dc1fd76122",
        "doctorId": "4bf35c0b-021a-4649-8ffc-cd3fd52b0d89",
        "userId": "6ad96f9b-f34f-4851-a472-4c85c1d79866",
        "timeSlot": "01PM-02PM",
        "status": "PAYMENT_PENDING",
        "appointmentDate": "2022-04-02"
    }
]
```

Calling the endpoint from POSTMAN -


BookMyConsultation / appointment-service / Fetch all the appointments of a user based on userId

GET http://3.232.56.68:8083/users/6ad96f9b-f34f-4851-a472-4c85c1d79866/appointments

Status: 200 OK Time: 480 ms Size: 636 B

```
[
    {
        "appointmentId": "ed834162-97d4-4f42-9c1b-457c0d562948",
        "doctorId": "ca6ad87b-f501-41ba-9a71-dc3d143e8ebd",
        "userId": "6ad96f9b-f34f-4851-a472-4c85c1d79866",
        "timeSlot": "9AM-10AM",
        "status": "PAYMENT_PENDING",
        "appointmentDate": "2022-01-01"
    },
    {
        "appointmentId": "977e6645-bd76-4a15-87d0-f5dc1fd76122",
        "doctorId": "4bf35c0b-021a-4649-8ffc-cd3fd52b0d89",
        "userId": "6ad96f9b-f34f-4851-a472-4c85c1d79866",
        "timeSlot": "01PM-02PM",
        "status": "PAYMENT_PENDING",
        "appointmentDate": "2022-04-02"
    }
]
```

## 7.6. Endpoint-6: Send prescription

**GET** localhost:8083 /users/{userId}/appointments

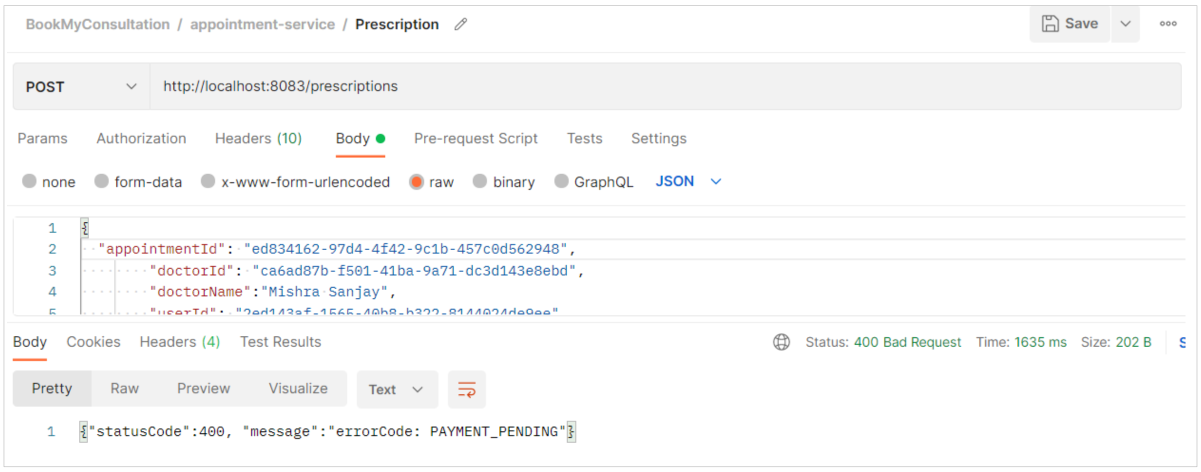
Request Body Ex -

```
{
  "appointmentId": "ed834162-97d4-4f42-9c1b-457c0d562948",
        "doctorId": "ca6ad87b-f501-41ba-9a71-dc3d143e8ebd",
        "doctorName":"Mishra Sanjay",
        "userId": "2ed143af-1565-40b8-b322-8144024de9ee",
        "diagnosis":" Teeth Cavity",
        "medicineList":[
          {
            "name":"Calpol",
            "type":"Tablet",
            "dosage":"1 week",
            "duration":"1 week",
            "frequency":"3 times a day",
            "remarks":"after food"
          },
          {
            "name":"PainKill",
            "type":"Syrup",
            "dosage":"1 week",
            "duration":"1 week",
            "frequency":"3 times a day",
            "remarks":"after food"
          }
          ]
}
```

Response Ex -

```
{
    "appointmentId": "ed834162-97d4-4f42-9c1b-457c0d562948",
    "doctorId": "ca6ad87b-f501-41ba-9a71-dc3d143e8ebd",
    "doctorName": "Mishra Sanjay",
    "userId": "2ed143af-1565-40b8-b322-8144024de9ee",
    "diagnosis": " Teeth Cavity",
    "medicineList": [
        {
            "name": "Calpol",
            "type": "Tablet",
            "dosage": "1 week",
            "duration": "1 week",
            "frequency": "3 times a day",
            "remarks": "after food"
        },
        {
            "name": "PainKill",
            "type": "Syrup",
            "dosage": "1 week",
            "duration": "1 week",
            "frequency": "3 times a day",
            "remarks": "after food"
        }
    ]
}
```

Calling the endpoint from POSTMAN when PAYMENT is pending -



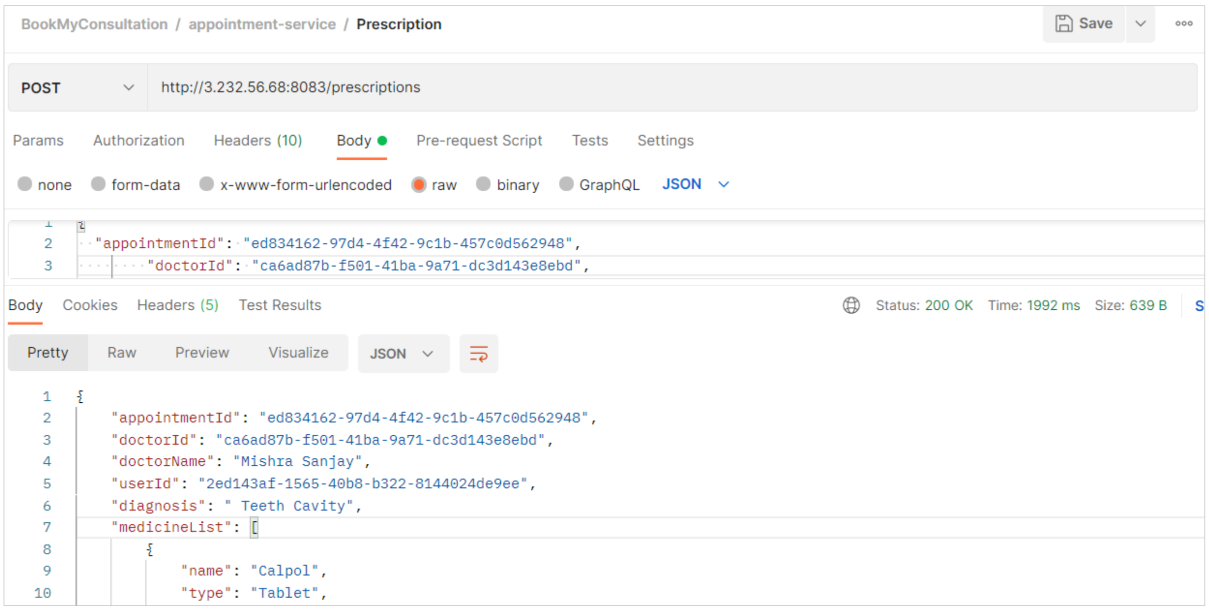
http://localhost:8083/prescriptions

```
{
  "appointmentId": "ed834162-97d4-4f42-9c1b-457c0d562948",
        "doctorId": "ca6ad87b-f501-41ba-9a71-dc3d143e8ebd",
        "doctorName":"Mishra Sanjay",
```

Status: 400 Bad Request Time: 1635 ms Size: 202 B

```
{"statusCode":400, "message":"errorCode: PAYMENT_PENDING"}
```

Calling the endpoint from POSTMAN when PAYMENT is confirmed –



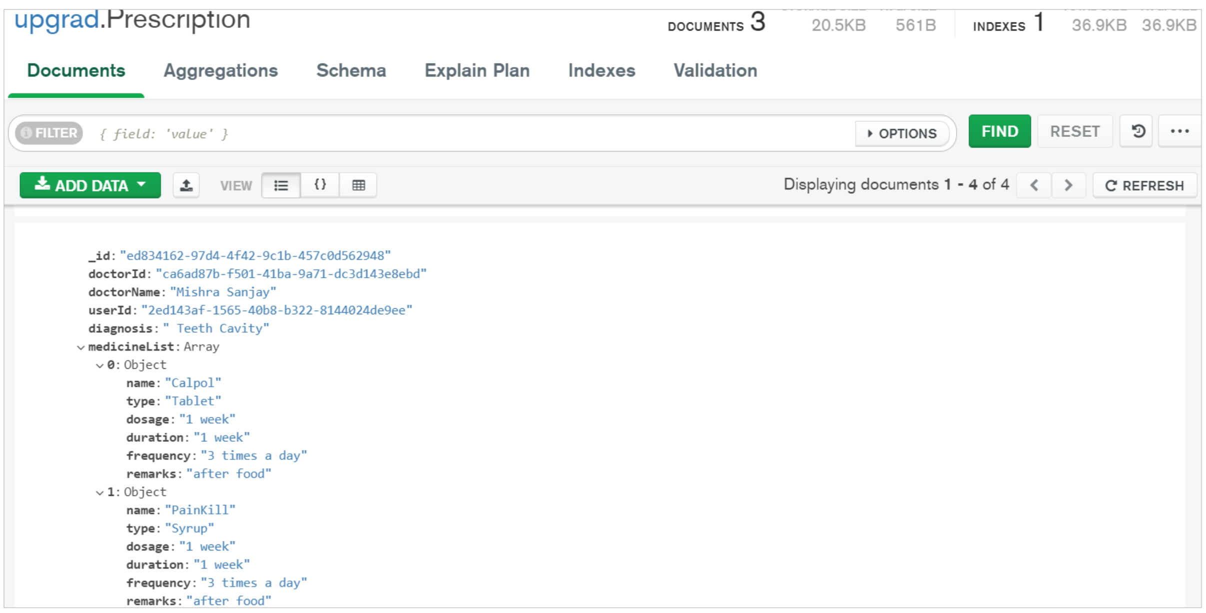
http://3.232.56.68:8083/prescriptions

```
  "appointmentId": "ed834162-97d4-4f42-9c1b-457c0d562948",
        "doctorId": "ca6ad87b-f501-41ba-9a71-dc3d143e8ebd",
```

Status: 200 OK Time: 1992 ms Size: 639 B

```
{
    "appointmentId": "ed834162-97d4-4f42-9c1b-457c0d562948",
    "doctorId": "ca6ad87b-f501-41ba-9a71-dc3d143e8ebd",
    "doctorName": "Mishra Sanjay",
    "userId": "2ed143af-1565-40b8-b322-8144024de9ee",
    "diagnosis": " Teeth Cavity",
    "medicineList": [
        {
            "name": "Calpol",
            "type": "Tablet",
```

Mongo Collection 'Prescription' –



```
_id: "ed834162-97d4-4f42-9c1b-457c0d562948"
doctorId: "ca6ad87b-f501-41ba-9a71-dc3d143e8ebd"
doctorName: "Mishra Sanjay"
userId: "2ed143af-1565-40b8-b322-8144024de9ee"
diagnosis: " Teeth Cavity"
medicineList: Array
  0: Object
     name: "Calpol"
     type: "Tablet"
     dosage: "1 week"
     duration: "1 week"
     frequency: "3 times a day"
     remarks: "after food"
  1: Object
     name: "PainKill"
     type: "Syrup"
     dosage: "1 week"
     duration: "1 week"
     frequency: "3 times a day"
     remarks: "after food"
```

# 8. Payment Service

## 8.1. Endpoint-1: Make payment for appointment.

**POST** localhost:8084/payments?appointmentId=<appointment-id>

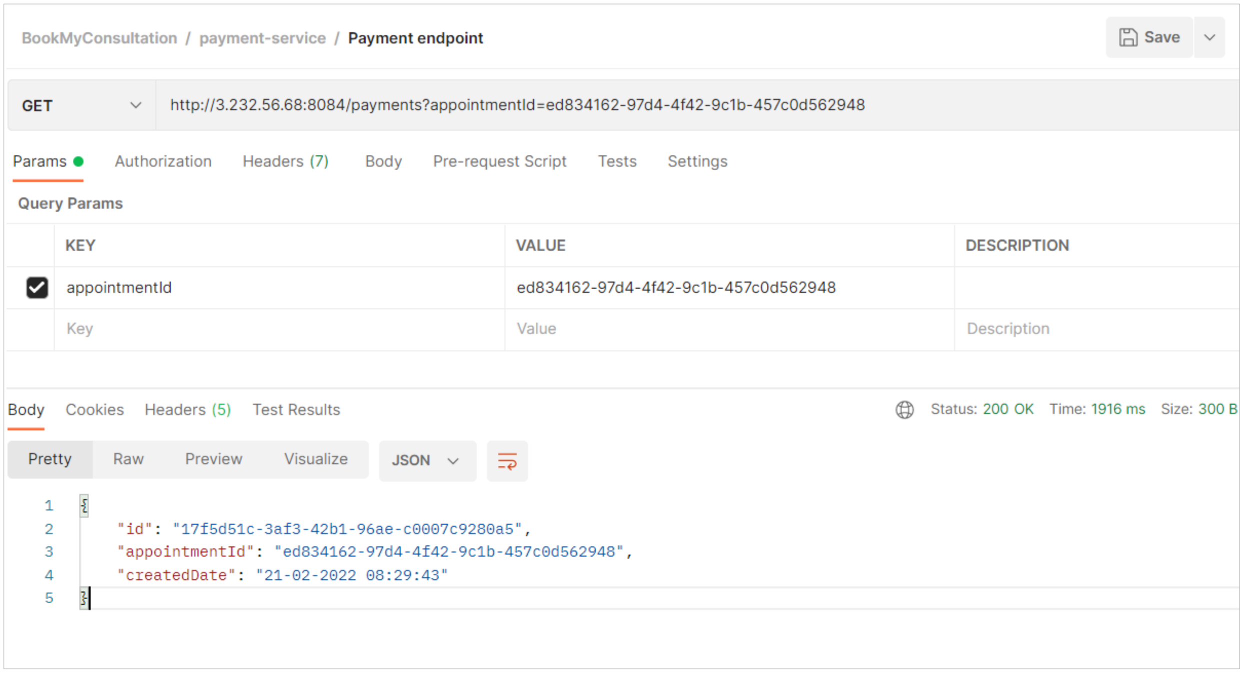
Request Body Ex –

```
http://3.232.56.68:8084/payments?appointmentId=ed834162-97d4-4f42-9c1b-457c0d562948
```

Response Ex -

```
{
    "id": "17f5d51c-3af3-42b1-96ae-c0007c9280a5",
    "appointmentId": "ed834162-97d4-4f42-9c1b-457c0d562948",
    "createdDate": "21-02-2022 08:29:43"
}
```

Calling the endpoint from POSTMAN –



BookMyConsultation / payment-service / Payment endpoint

GET http://3.232.56.68:8084/payments?appointmentId=ed834162-97d4-4f42-9c1b-457c0d562948

Query Params

| | KEY | VALUE | DESCRIPTION |
|---|---|---|---|
| ✓ | appointmentId | ed834162-97d4-4f42-9c1b-457c0d562948 | |

Status: 200 OK Time: 1916 ms Size: 300 B

```
{
    "id": "17f5d51c-3af3-42b1-96ae-c0007c9280a5",
    "appointmentId": "ed834162-97d4-4f42-9c1b-457c0d562948",
    "createdDate": "21-02-2022 08:29:43"
}
```

## 9. Rating Service

### 9.1. Endpoint-1: Rate Doctors.

POST localhost:8085/ratings

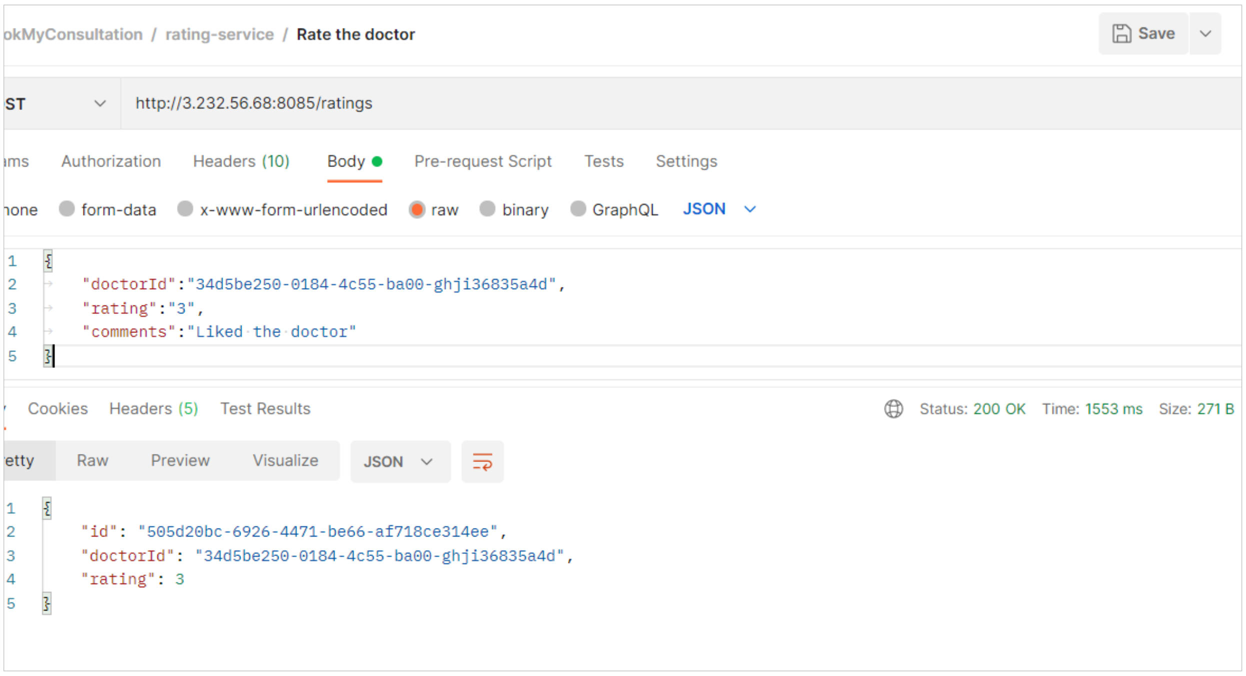
Request Body Ex –

```
{
    "doctorId":"34d5be250-0184-4c55-ba00-ghji36835a4d",
    "rating":"3",
    "comments":"Liked the doctor"
}
```

Response Ex -

```
{
    "id": "505d20bc-6926-4471-be66-af718ce314ee",
    "doctorId": "34d5be250-0184-4c55-ba00-ghji36835a4d",
    "rating": 3
}
```

Calling the endpoint from POSTMAN –



http://3.232.56.68:8085/ratings

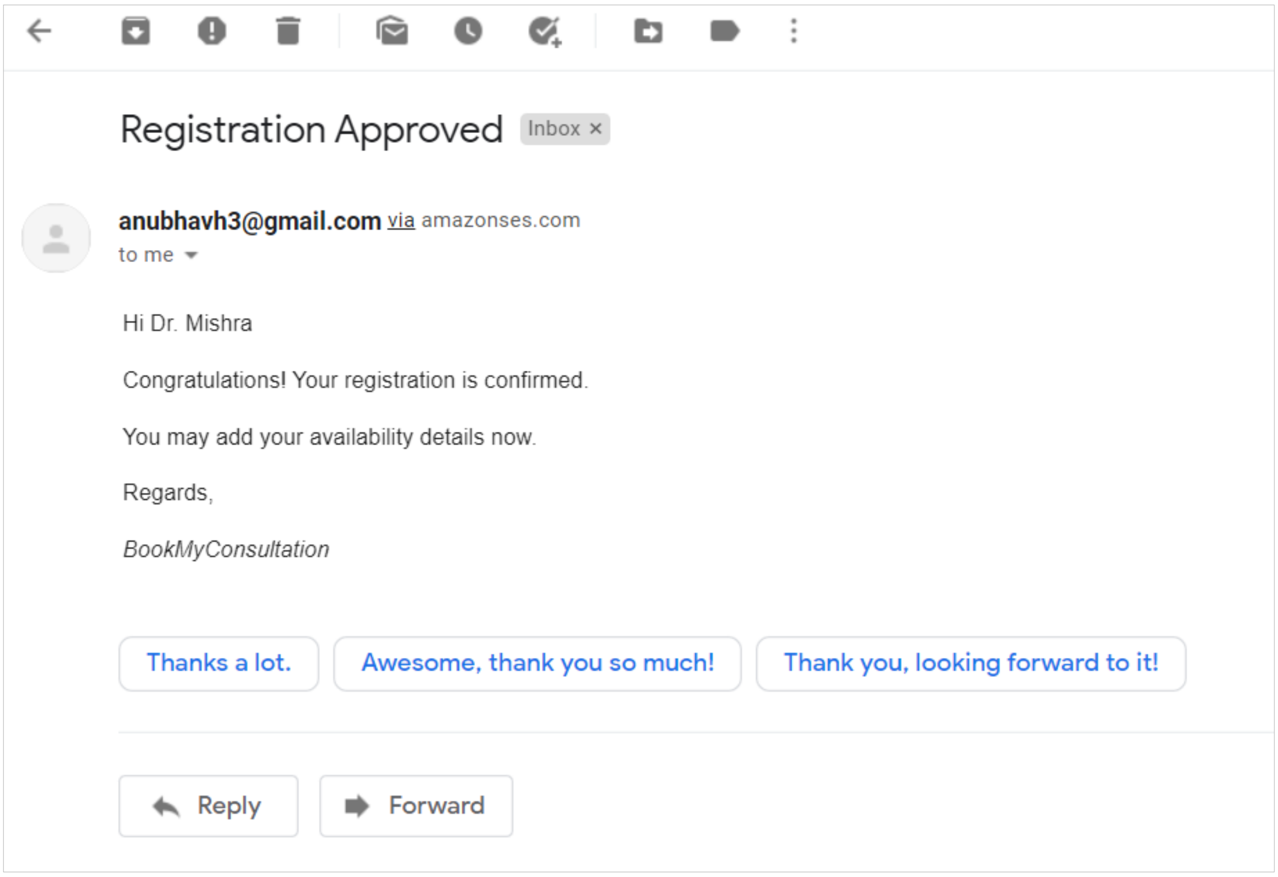
```
{
    "doctorId":"34d5be250-0184-4c55-ba00-ghji36835a4d",
    "rating":"3",
    "comments":"Liked the doctor"
}
```

Status: 200 OK Time: 1553 ms Size: 271 B

```
{
    "id": "505d20bc-6926-4471-be66-af718ce314ee",
    "doctorId": "34d5be250-0184-4c55-ba00-ghji36835a4d",
    "rating": 3
}
```

## 10. Notification Service

### 10.1. Doctor Registration Approval


Registration Approved Inbox ×

anubhavh3@gmail.com via amazonses.com
to me

Hi Dr. Mishra

Congratulations! Your registration is confirmed.

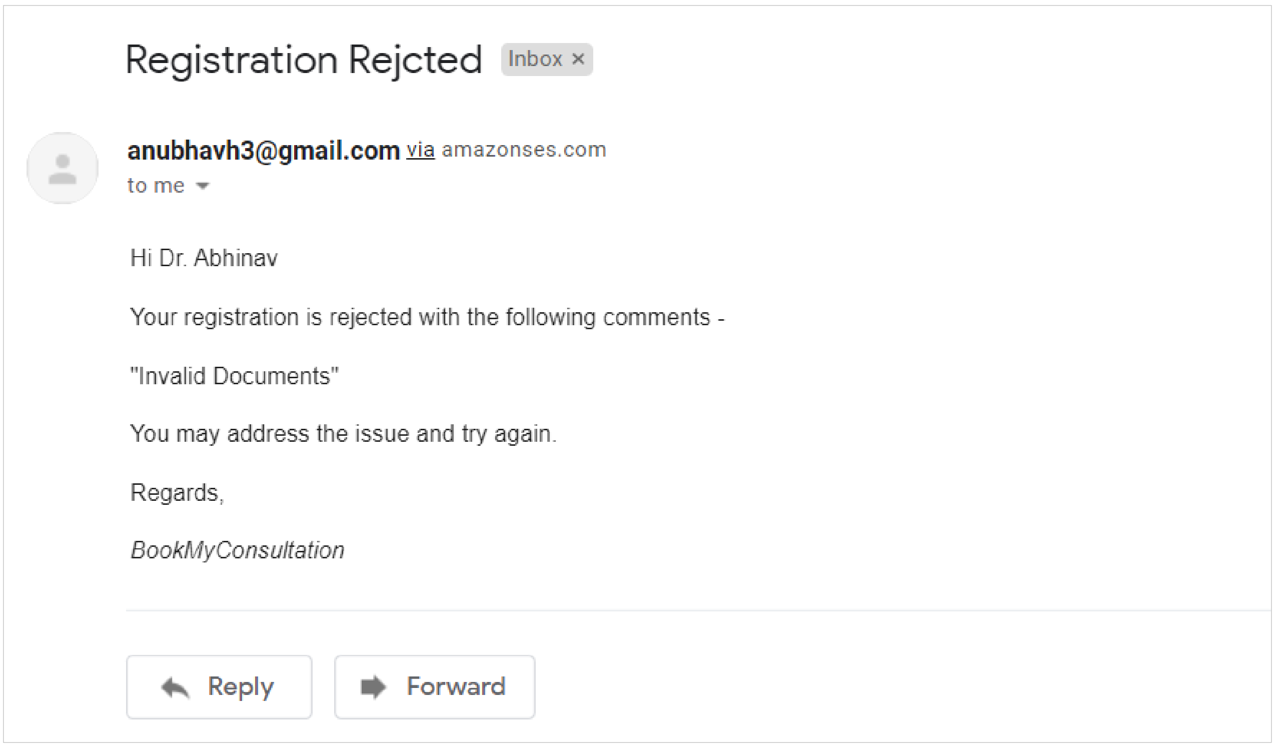
You may add your availability details now.

Regards,

*BookMyConsultation*

Thanks a lot.
Awesome, thank you so much!
Thank you, looking forward to it!
Reply
Forward

### 10.2. Doctor Registration Rejection


Registration Rejcted Inbox ×

anubhavh3@gmail.com via amazonses.com
to me

Hi Dr. Abhinav

Your registration is rejected with the following comments -

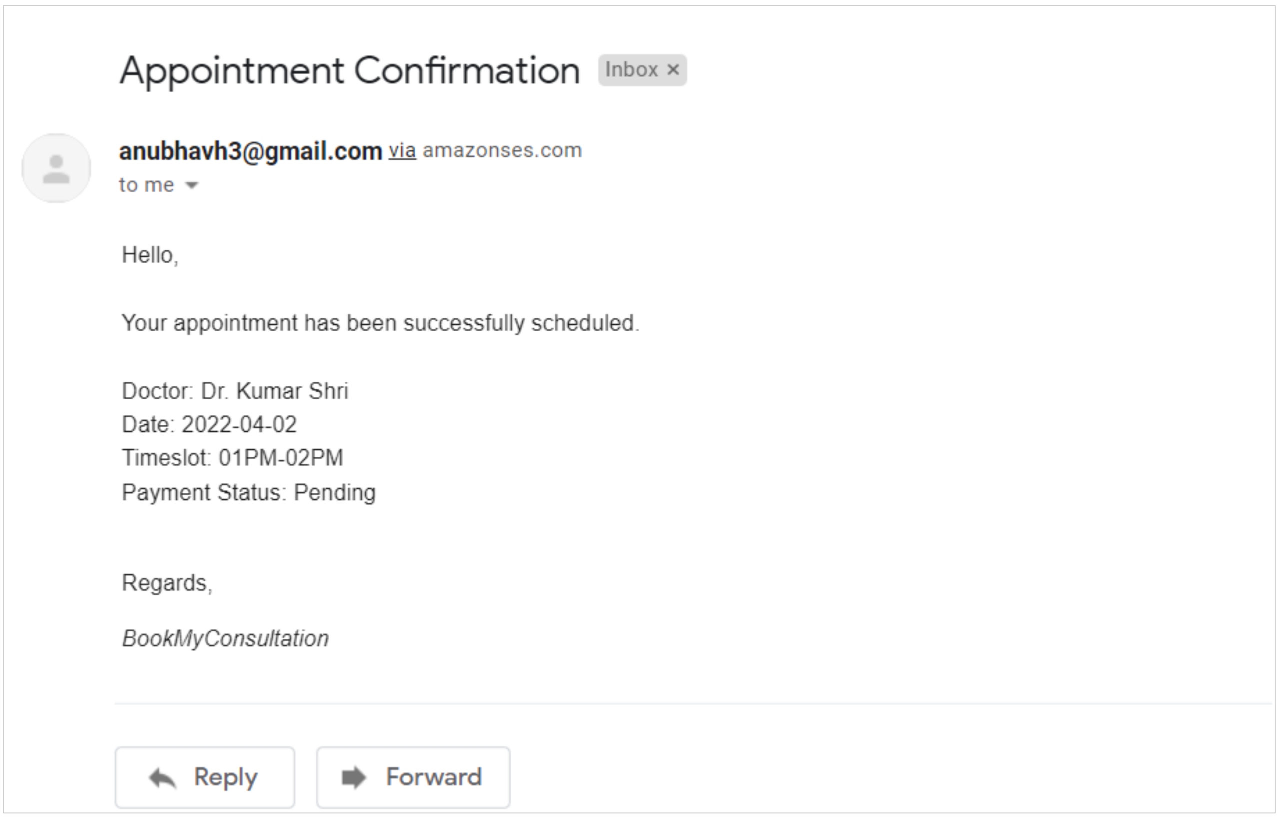
"Invalid Documents"

You may address the issue and try again.

Regards,

*BookMyConsultation*

## 10.3. Appointment Confirmation



Appointment Confirmation

anubhavh3@gmail.com via amazonses.com
to me

Hello,

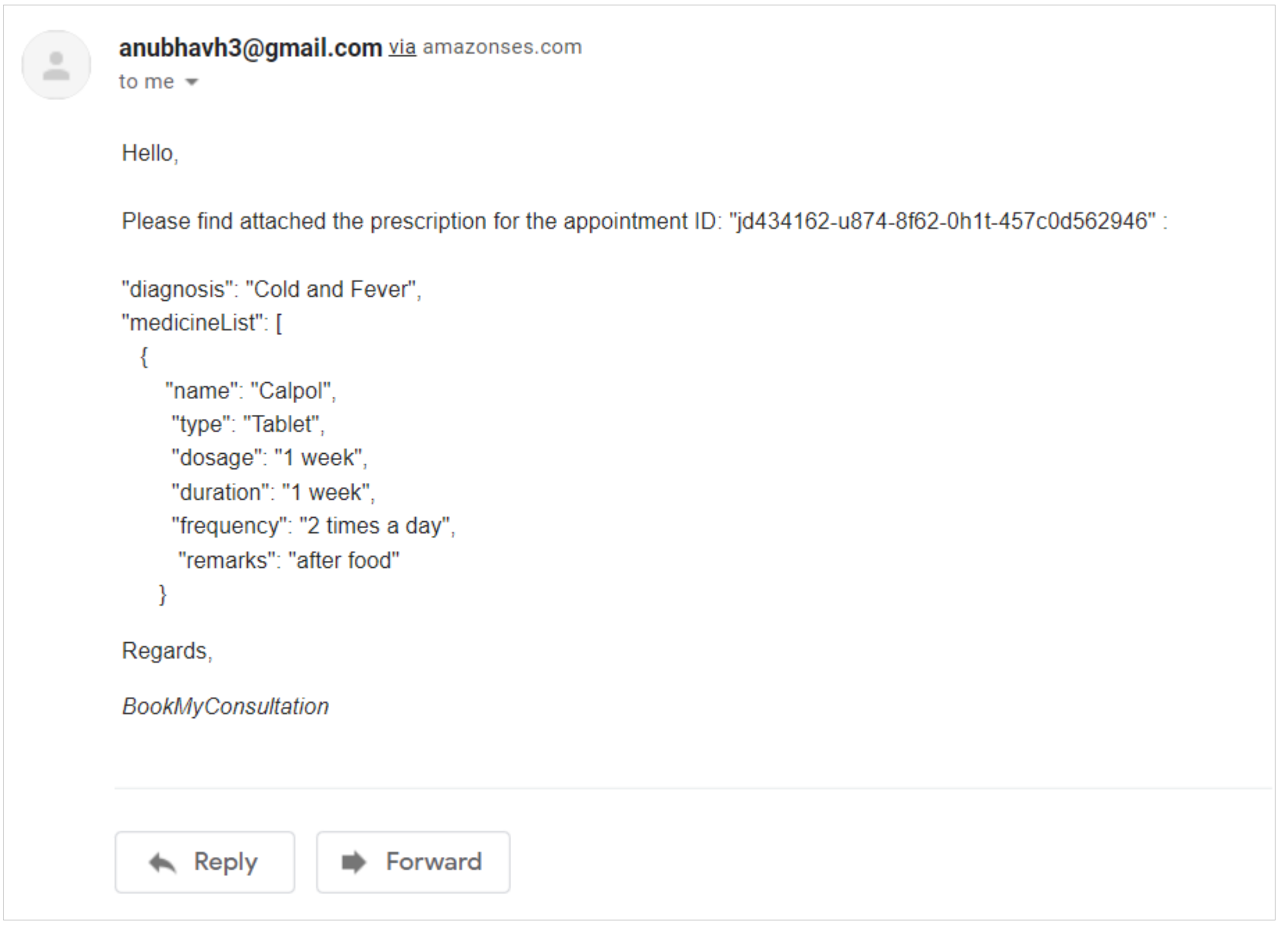
Your appointment has been successfully scheduled.

Doctor: Dr. Kumar Shri
Date: 2022-04-02
Timeslot: 01PM-02PM
Payment Status: Pending

Regards,

*BookMyConsultation*

## 10.4. Prescription



anubhavh3@gmail.com via amazonses.com
to me

Hello,

Please find attached the prescription for the appointment ID: "jd434162-u874-8f62-0h1t-457c0d562946" :

```
"diagnosis": "Cold and Fever",
"medicineList": [
  {
     "name": "Calpol",
      "type": "Tablet",
      "dosage": "1 week",
      "duration": "1 week",
      "frequency": "2 times a day",
       "remarks": "after food"
    }
```

Regards,

*BookMyConsultation*

## 11. API Gateway

An API Gateway has been implemented to act as reverse proxy. Instead of calling the individual API's, one can call the API Gateway like following-

```
http://<EC2host|localhost>:8080/doctorsvc/ . . .

http://<EC2host|localhost>:8080/usersvc/ . . .

http://<EC2host|localhost>:8080/appointmentsvc/ . . .

http://<EC2host|localhost>:8080/paymentsvc/ . . .

http://<EC2host|localhost>:8080/ratingsvc/ . . .
```

## 12. Security

Token generation and validation has not been implemented in this submission.

## 13. Future Enhancements

- Add error and exception handling at all the possible points
- Implement a Configuration Server

*** END ***