Hospital Management System

By Naga Satya Sai Pavirala

Tools Used:

1.Spring boot framework for easy development

2.Maven build tool for building the project

3.Github for version control system of code created

4.Swagger UI for graphical user interface

5.Postman tool to handle get and post requests

6.Mongo db(no sql) database to store data

7.S3 Studio for interacting with database in graphical user interface

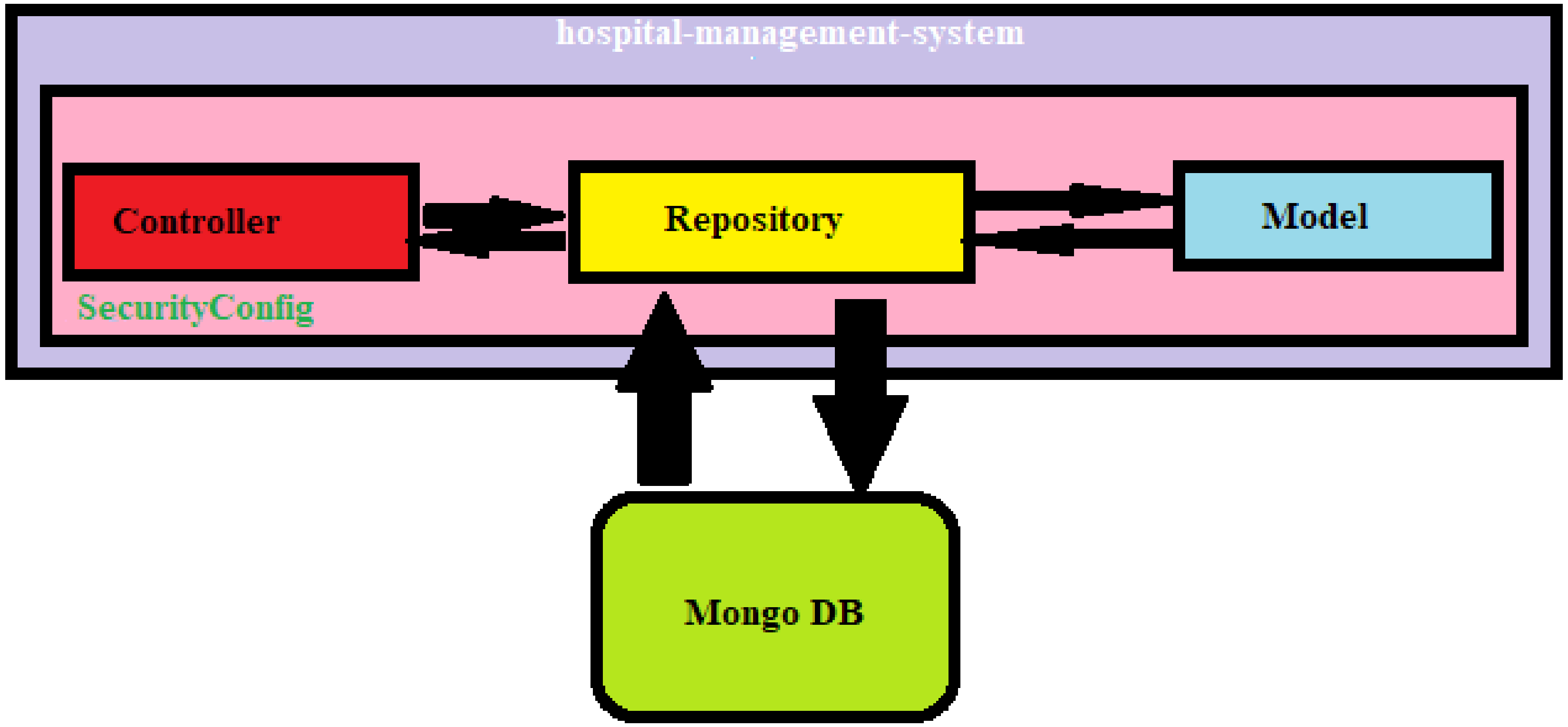
7.Junit for testing the code

8.Docker Desktop for deploying the images into container and for running the images

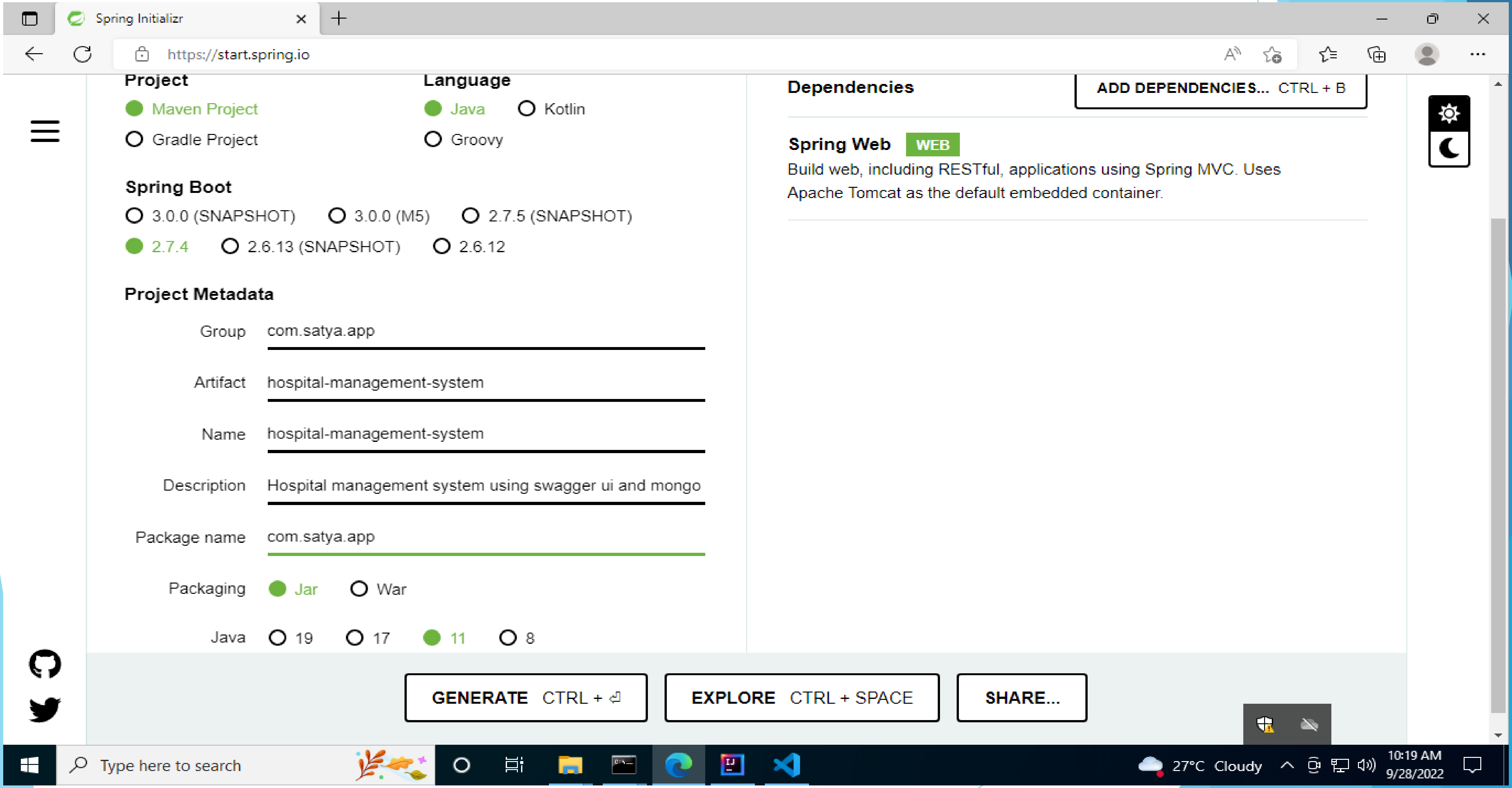
List of dependencies added in POM.xml:

1. spring-boot-starter-web
2. spring-boot-starter-test
3. spring-boot-starter-security
4. spring-security-test
5. springdoc-openapi-ui
6. commons-io
7. Junit
8. spring-boot-starter-data-mongodb

Architecture:



Creation of Project:



Spring Initializr

https://start.spring.io

Project

● Maven Project

○ Gradle Project

Language

● Java ○ Kotlin

○ Groovy

Spring Boot

○ 3.0.0 (SNAPSHOT) ○ 3.0.0 (M5) ○ 2.7.5 (SNAPSHOT)

● 2.7.4 ○ 2.6.13 (SNAPSHOT) ○ 2.6.12

Project Metadata

Group	com.satya.app
Artifact	hospital-management-system
Name	hospital-management-system
Description	Hospital management system using swagger ui and mongo
Package name	com.satya.app
Packaging	● Jar ○ War
Java	○ 19 ○ 17 ● 11 ○ 8

Dependencies

Spring Web WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

MVC Files used in system:

1. Controllers:
   - a. Doctor
   - b. Patient
   - c. Prescription
2. Data model
   - a. Appointment
   - b. Prescription
3. Repository
   - a. Appointment repository
   - b. Prescription repository

Data Model Attributes:

a) Appointment Data Model:

```
String appointmentId;
String patientName;
String doctorName;
String date;
Prescription prescription;
```

Data Model Attributes:

b)Prescription Data Model :

```
String prescriptionId;
String appointmentId;
String description;
String patientName;
String doctorName;
```

Doctor Controller:

“/save” - This will send POST requests along with Appointment object in JSON format as body.

Ex - “http://localhost:8081/doctor/save” .

```
{
   "appointmentId":"appointment1",
   "patientName":"patient1",
   "doctorName":"doctor1",
   "date":"29/09/2022",
   "prescription":{
    "prescriptionId":"prescription1",
    "appointmentId":"appointment1",
    "description":"description1",
    "patientName":"patient1",
    "doctorName":"doctor1"
   }
}
```

“/doctorappointment” - This will receive GET requests along with request parameters.

Ex-”http://localhost:8081/doctor/doctorappointment?doctorNa me=doctor1”.

Patient Controller:

“/save” – This will send POST requests along with Appointment object in JSON format as body.

Ex – “http://localhost:8081/patient/save” .

```
{
   "appointmentId":"appointment1",
   "patientName":"patient1",
   "doctorName":"doctor1",
   "date":"29/09/2022",
   "prescription":{
    "prescriptionId":"prescription1",
    "appointmentId":"appointment1",
    "description":"description1",
    "patientName":"patient1",
    "doctorName":"doctor1"
   }
}
```

“/myappointment” – This will receive GET requests along with request parameters.

Ex-”http://localhost:8081/patient/patientappointment?patie ntName=patient1”.

PrescriptionController:

“/saveprescription” – This will send POST requests along with Prescription object in JSON format as body.

Ex – “http://localhost:8081/saveprescription” .

```
{
    "prescriptionId":"prescription1",
    "appointmentId":"appointment1",
    "description":"prescription1",
    "patientName":"patient1",
    "doctorName":"doctor1"
}
```

“/viewprescription” - which will receive GET requests along with request parameters.

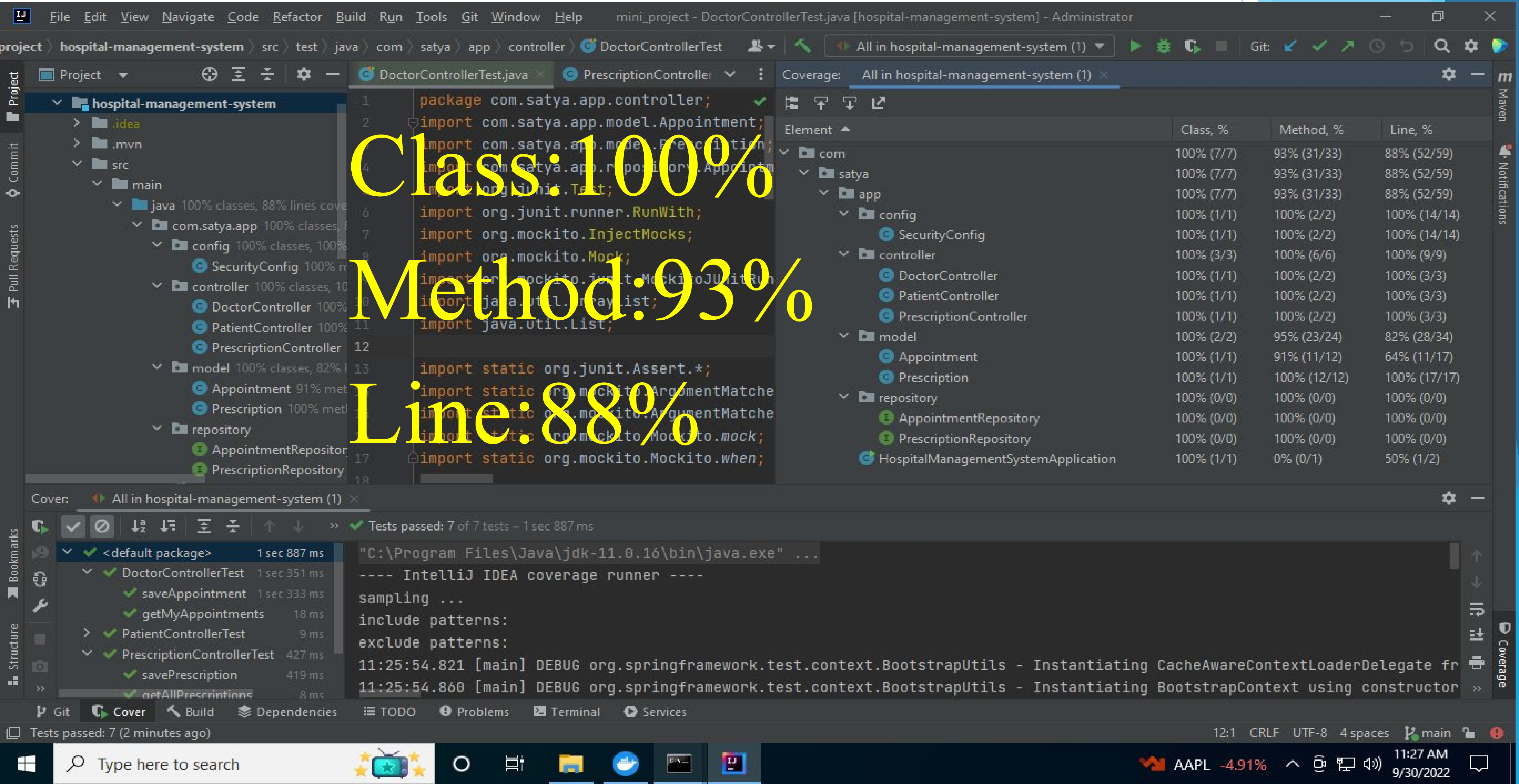
Ex-”http://localhost:8081/saveprescription?patientName= patient1”.

Security access role based:

Security is maintained by using security config file and access to controllers were allowed by using the following usernames and passwords.

“ /swagger-ui/index.html”	-	Username:admin01 Password:password
“ /doctor/doctorappointment”	-	Username:doctor01 Password:password
“/doctor/save“	-	Username:doctor01 Password:password
“/patient/myappointment“	-	Username:patient01 Password:password
“/patient/save”	-	Username:patient01 Password:password
“/viewprescrption“	-	Username:admin01 Password:password
“/saveprescrption”	-	Username:admin01 Password:password

Code Coverage:


Class:100%

Method:93%

Line:88%

```
package com.satya.app.controller;
import com.satya.app.model.Appointment;
import com.satya.app.model.Prescription;
import com.satya.app.repository.Appointm
import org.junit.Test;
import org.junit.runner.RunWith;
import org.mockito.InjectMocks;
import org.mockito.Mock;
import org.mockito.junit.MockitoJUnitRun
import java.util.ArrayList;
import java.util.List;

import static org.junit.Assert.*;
import static org.mockito.ArgumentMatche
import static org.mockito.ArgumentMatche
import static org.mockito.Mockito.mock;
import static org.mockito.Mockito.when;
```

Coverage: All in hospital-management-system (1)

Element	Class, %	Method, %	Line, %
com	100% (7/7)	93% (31/33)	88% (52/59)
satya	100% (7/7)	93% (31/33)	88% (52/59)
app	100% (7/7)	93% (31/33)	88% (52/59)
config	100% (1/1)	100% (2/2)	100% (14/14)
SecurityConfig	100% (1/1)	100% (2/2)	100% (14/14)
controller	100% (3/3)	100% (6/6)	100% (9/9)
DoctorController	100% (1/1)	100% (2/2)	100% (3/3)
PatientController	100% (1/1)	100% (2/2)	100% (3/3)
PrescriptionController	100% (1/1)	100% (2/2)	100% (3/3)
model	100% (2/2)	95% (23/24)	82% (28/34)
Appointment	100% (1/1)	91% (11/12)	64% (11/17)
Prescription	100% (1/1)	100% (12/12)	100% (17/17)
repository	100% (0/0)	100% (0/0)	100% (0/0)
AppointmentRepository	100% (0/0)	100% (0/0)	100% (0/0)
PrescriptionRepository	100% (0/0)	100% (0/0)	100% (0/0)
HospitalManagementSystemApplication	100% (1/1)	0% (0/1)	50% (1/2)

Cover: All in hospital-management-system (1)

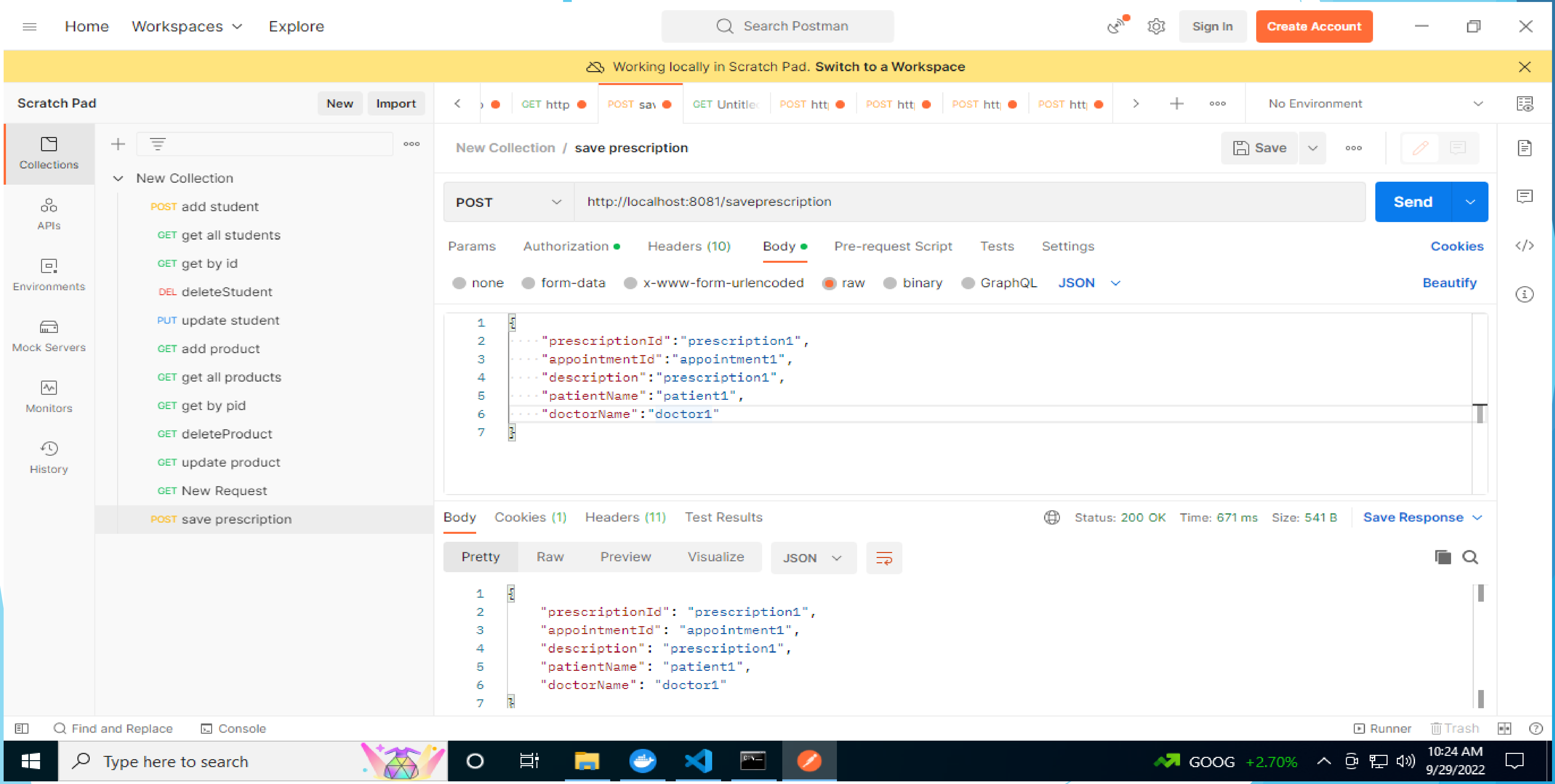
Tests passed: 7 of 7 tests – 1 sec 887 ms

```
<default package>              1 sec 887 ms
  DoctorControllerTest         1 sec 351 ms
    saveAppointment            1 sec 333 ms
    getMyAppointments          18 ms
  PatientControllerTest        9 ms
  PrescriptionControllerTest   427 ms
    savePrescription           419 ms
    getAllPrescriptions        8 ms
```

```
"C:\Program Files\Java\jdk-11.0.16\bin\java.exe" ...
---- IntelliJ IDEA coverage runner ----
sampling ...
include patterns:
exclude patterns:
11:25:54.821 [main] DEBUG org.springframework.test.context.BootstrapUtils - Instantiating CacheAwareContextLoaderDelegate fr
11:25:54.860 [main] DEBUG org.springframework.test.context.BootstrapUtils - Instantiating BootstrapContext using constructor
```

mini_project - DoctorControllerTest.java [hospital-management-system] - Administrator
File Edit View Navigate Code Refactor Build Run Tools Git Window Help
project › hospital-management-system › src › test › java › com › satya › app › controller › DoctorControllerTest
Project
hospital-management-system
.idea
.mvn
src
main
java 100% classes, 88% lines cove
com.satya.app 100% classes,
config 100% classes, 100%
SecurityConfig 100% m
controller 100% classes, 10
DoctorController 100%
PatientController 100%
PrescriptionController
model 100% classes, 82% l
Appointment 91% met
Prescription 100% metl
repository
AppointmentRepositor
PrescriptionRepository
DoctorControllerTest.java
PrescriptionController
Git Cover Build Dependencies TODO Problems Terminal Services
Tests passed: 7 (2 minutes ago)
12:1 CRLF UTF-8 4 spaces main
Type here to search
AAPL -4.91%
11:27 AM
9/30/2022

Postman POST request:



New Collection
- POST add student
- GET get all students
- GET get by id
- DEL deleteStudent
- PUT update student
- GET add product
- GET get all products
- GET get by pid
- GET deleteProduct
- GET update product
- GET New Request
- POST save prescription

New Collection / save prescription

POST http://localhost:8081/saveprescription

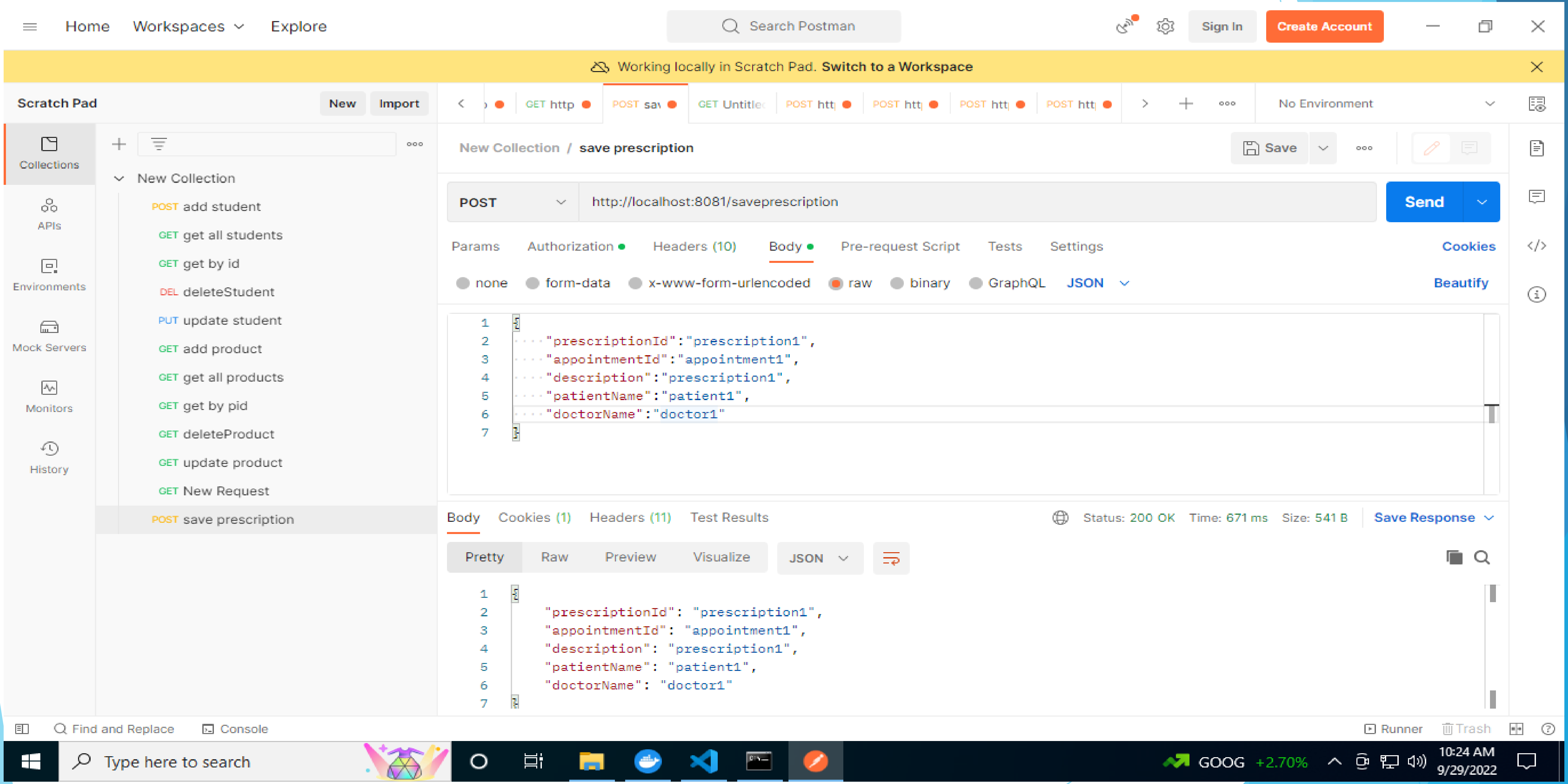
```
{
    "prescriptionId":"prescription1",
    "appointmentId":"appointment1",
    "description":"prescription1",
    "patientName":"patient1",
    "doctorName":"doctor1"
}
```

Body Cookies (1) Headers (11) Test Results   Status: 200 OK Time: 671 ms Size: 541 B

```
{
    "prescriptionId": "prescription1",
    "appointmentId": "appointment1",
    "description": "prescription1",
    "patientName": "patient1",
    "doctorName": "doctor1"
}
```

GOOG +2.70% 10:24 AM 9/29/2022

Postman GET request:



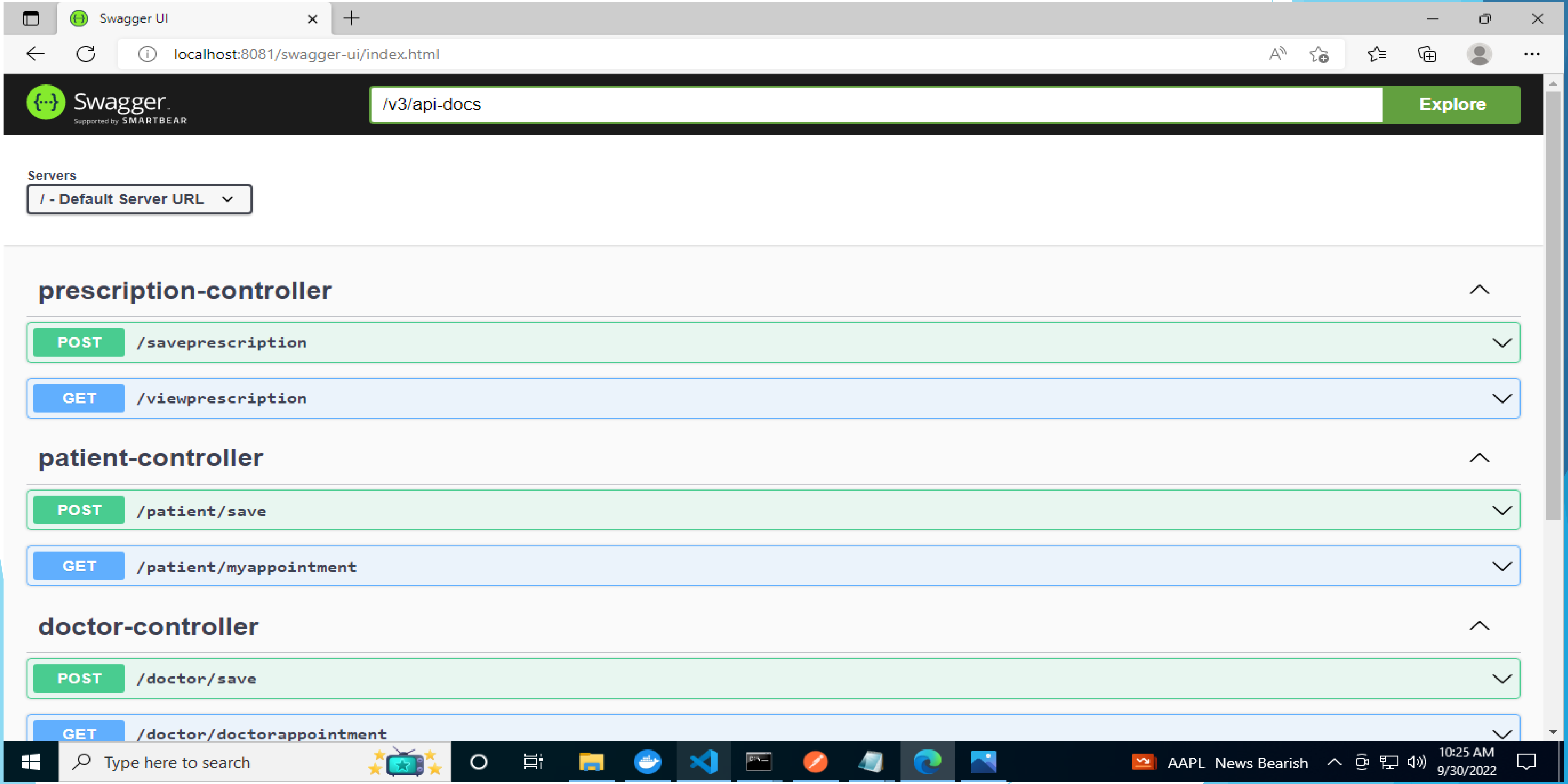
POST http://localhost:8081/saveprescription

```
{
    "prescriptionId":"prescription1",
    "appointmentId":"appointment1",
    "description":"prescription1",
    "patientName":"patient1",
    "doctorName":"doctor1"
}
```

Status: 200 OK Time: 671 ms Size: 541 B

```
{
    "prescriptionId": "prescription1",
    "appointmentId": "appointment1",
    "description": "prescription1",
    "patientName": "patient1",
    "doctorName": "doctor1"
}
```

Swagger UI



Swagger UI

localhost:8081/swagger-ui/index.html

/v3/api-docs

Explore

Servers

/ - Default Server URL

prescription-controller

POST /saveprescription

GET /viewprescription

patient-controller

POST /patient/save

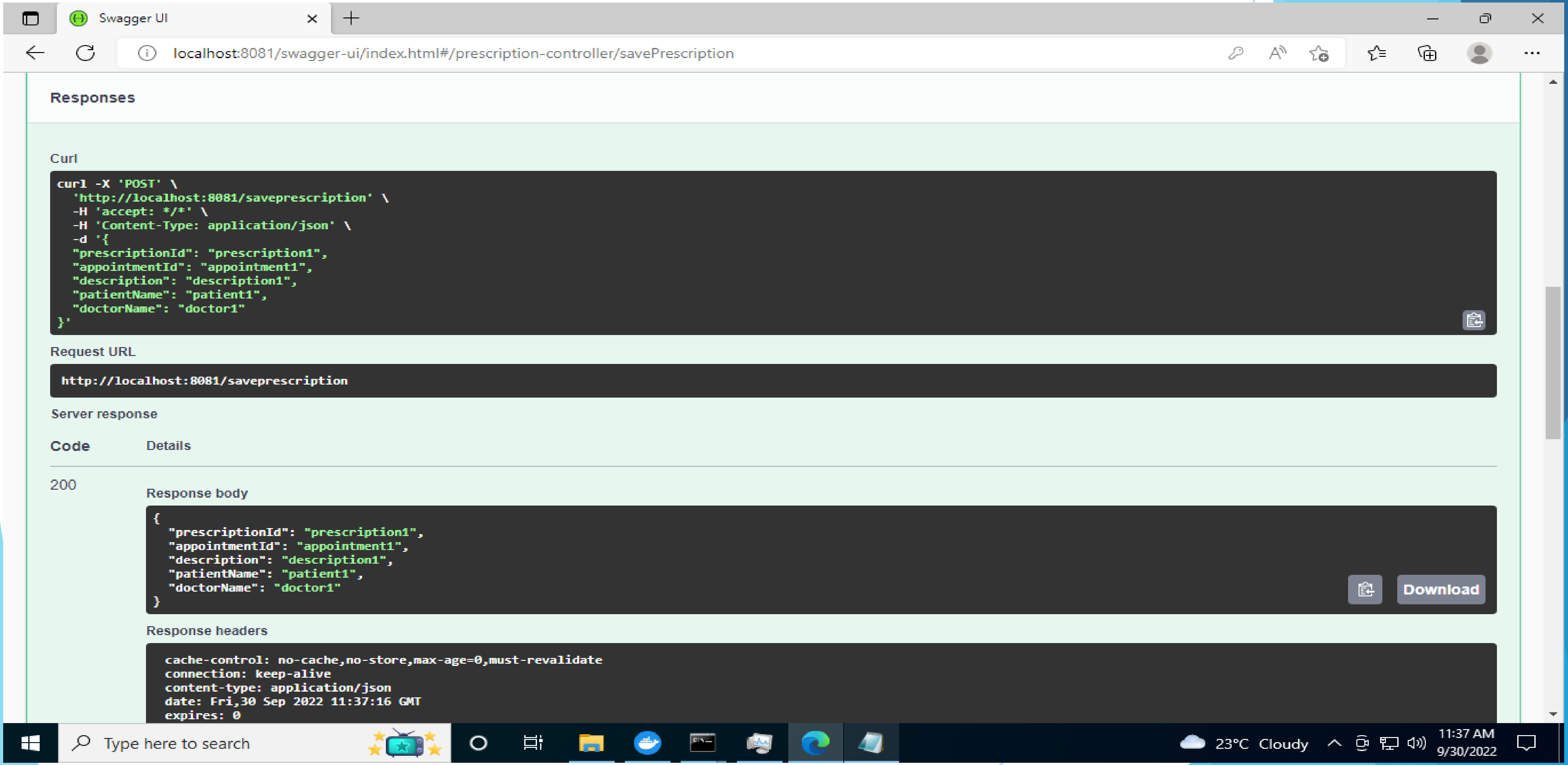
GET /patient/myappointment

doctor-controller

POST /doctor/save

GET /doctor/doctorappointment

Swagger UI POST request:



Swagger UI

localhost:8081/swagger-ui/index.html#/prescription-controller/savePrescription

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8081/saveprescription' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "prescriptionId": "prescription1",
  "appointmentId": "appointment1",
  "description": "description1",
  "patientName": "patient1",
  "doctorName": "doctor1"
}'
```

Request URL

```
http://localhost:8081/saveprescription
```

Server response

Code Details

200

Response body

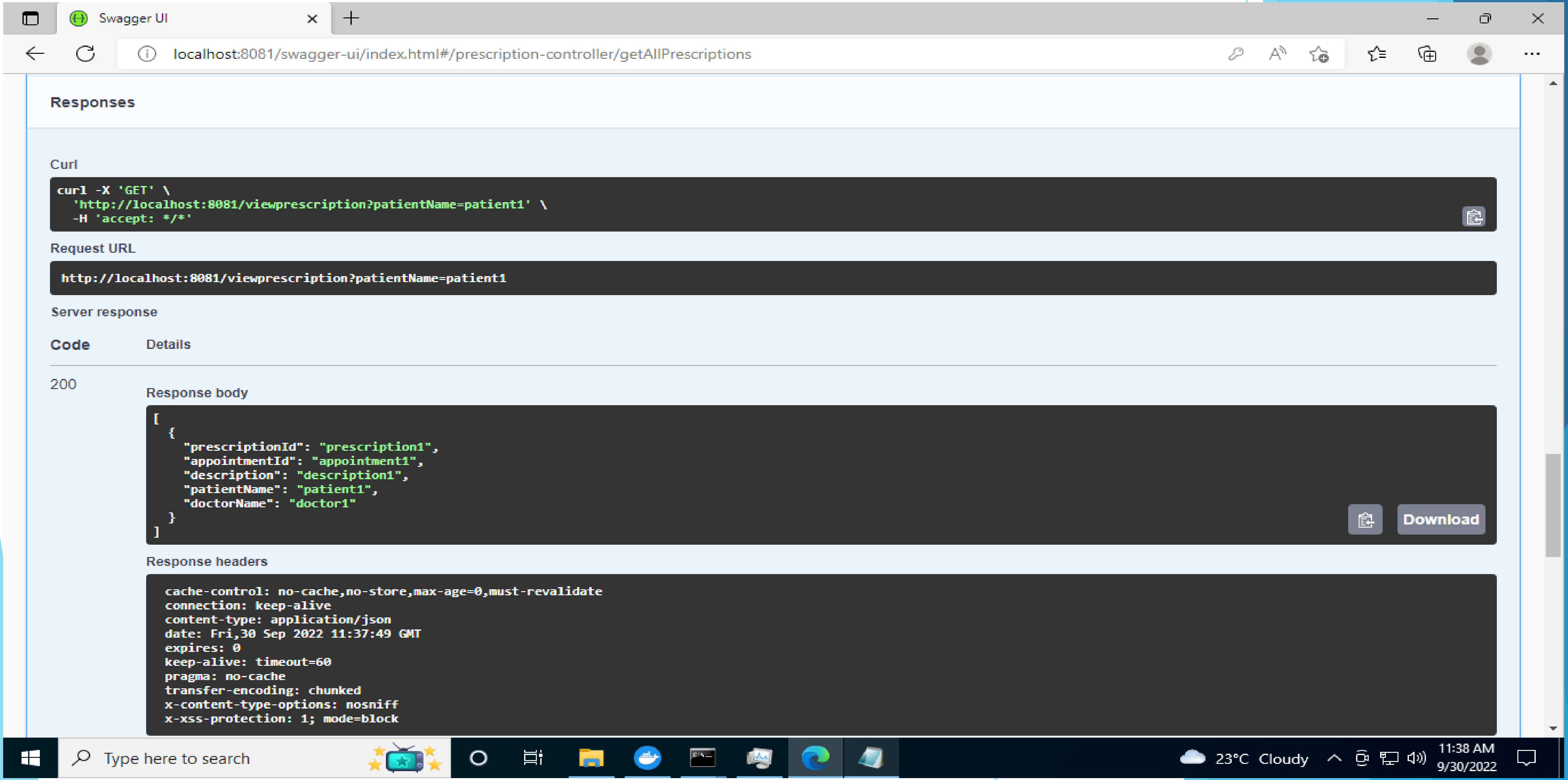
```
{
  "prescriptionId": "prescription1",
  "appointmentId": "appointment1",
  "description": "description1",
  "patientName": "patient1",
  "doctorName": "doctor1"
}
```

Download

Response headers

```
cache-control: no-cache,no-store,max-age=0,must-revalidate
connection: keep-alive
content-type: application/json
date: Fri,30 Sep 2022 11:37:16 GMT
expires: 0
```

Swagger UI GET request:



Swagger UI
localhost:8081/swagger-ui/index.html#/prescription-controller/getAllPrescriptions

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8081/viewprescription?patientName=patient1' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:8081/viewprescription?patientName=patient1
```

Server response

Code | Details

200

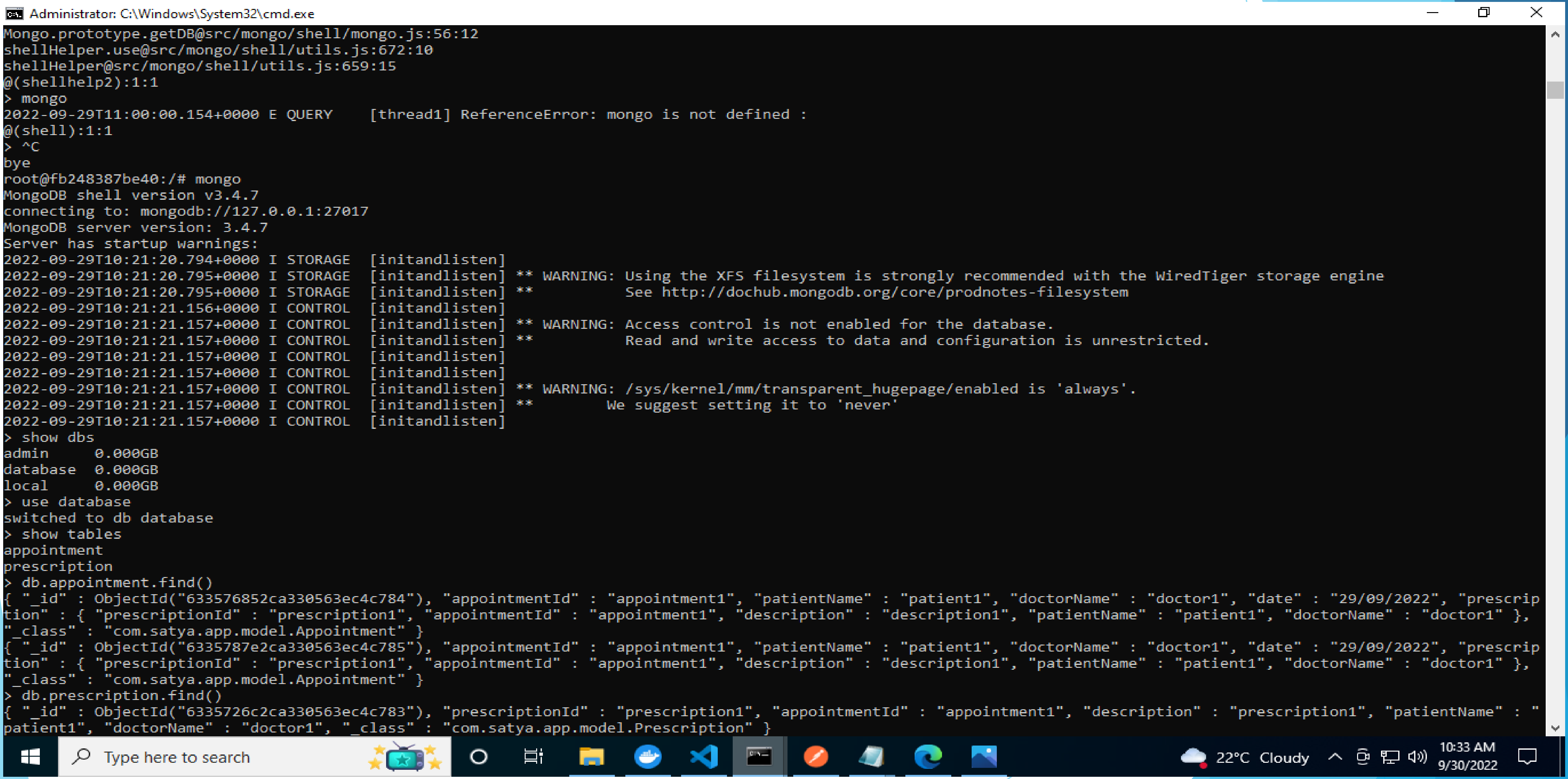
Response body

```
[
  {
    "prescriptionId": "prescription1",
    "appointmentId": "appointment1",
    "description": "description1",
    "patientName": "patient1",
    "doctorName": "doctor1"
  }
]
```

Response headers

```
 cache-control: no-cache,no-store,max-age=0,must-revalidate
 connection: keep-alive
 content-type: application/json
 date: Fri,30 Sep 2022 11:37:49 GMT
 expires: 0
 keep-alive: timeout=60
 pragma: no-cache
 transfer-encoding: chunked
 x-content-type-options: nosniff
 x-xss-protection: 1; mode=block
```

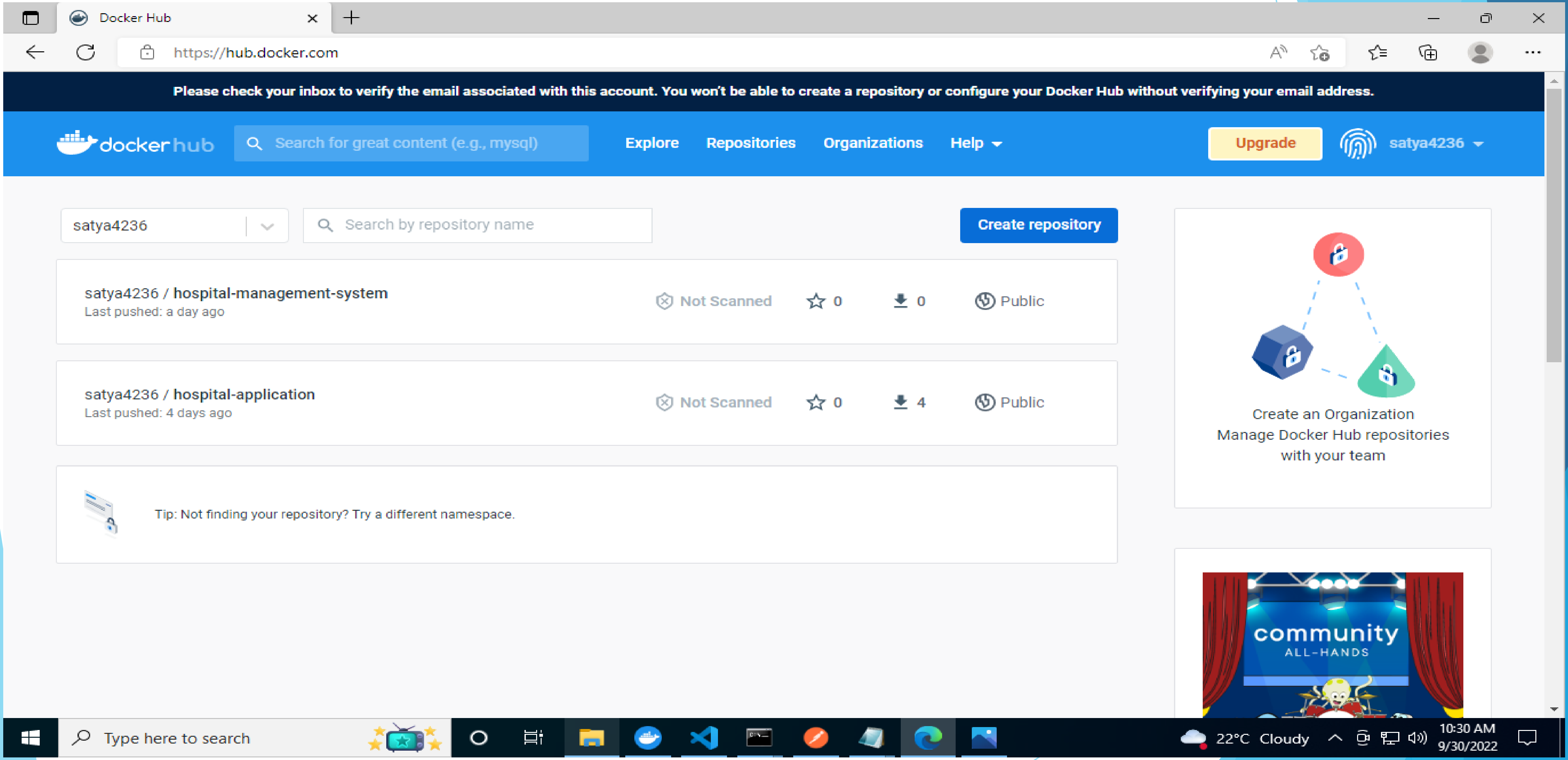
Mongodb bash:



Administrator: C:\Windows\System32\cmd.exe

```
Mongo.prototype.getDB@src/mongo/shell/mongo.js:56:12
shellHelper.use@src/mongo/shell/utils.js:672:10
shellHelper@src/mongo/shell/utils.js:659:15
@(shellhelp2):1:1
> mongo
2022-09-29T11:00:00.154+0000 E QUERY    [thread1] ReferenceError: mongo is not defined :
@(shell):1:1
> ^C
bye
root@fb248387be40:/# mongo
MongoDB shell version v3.4.7
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.7
Server has startup warnings:
2022-09-29T10:21:20.794+0000 I STORAGE  [initandlisten]
2022-09-29T10:21:20.795+0000 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-09-29T10:21:20.795+0000 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-09-29T10:21:21.156+0000 I CONTROL  [initandlisten]
2022-09-29T10:21:21.157+0000 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-09-29T10:21:21.157+0000 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2022-09-29T10:21:21.157+0000 I CONTROL  [initandlisten]
2022-09-29T10:21:21.157+0000 I CONTROL  [initandlisten]
2022-09-29T10:21:21.157+0000 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2022-09-29T10:21:21.157+0000 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2022-09-29T10:21:21.157+0000 I CONTROL  [initandlisten]
> show dbs
admin     0.000GB
database  0.000GB
local     0.000GB
> use database
switched to db database
> show tables
appointment
prescription
> db.appointment.find()
{ "_id" : ObjectId("633576852ca330563ec4c784"), "appointmentId" : "appointment1", "patientName" : "patient1", "doctorName" : "doctor1", "date" : "29/09/2022", "prescrip
tion" : { "prescriptionId" : "prescription1", "appointmentId" : "appointment1", "description" : "description1", "patientName" : "patient1", "doctorName" : "doctor1" },
"_class" : "com.satya.app.model.Appointment" }
{ "_id" : ObjectId("6335787e2ca330563ec4c785"), "appointmentId" : "appointment1", "patientName" : "patient1", "doctorName" : "doctor1", "date" : "29/09/2022", "prescrip
tion" : { "prescriptionId" : "prescription1", "appointmentId" : "appointment1", "description" : "description1", "patientName" : "patient1", "doctorName" : "doctor1" },
"_class" : "com.satya.app.model.Appointment" }
> db.prescription.find()
{ "_id" : ObjectId("6335726c2ca330563ec4c783"), "prescriptionId" : "prescription1", "appointmentId" : "appointment1", "description" : "prescription1", "patientName" : "
patient1", "doctorName" : "doctor1", "_class" : "com.satya.app.model.Prescription" }
```

Docker hub repository:



Docker Hub

https://hub.docker.com

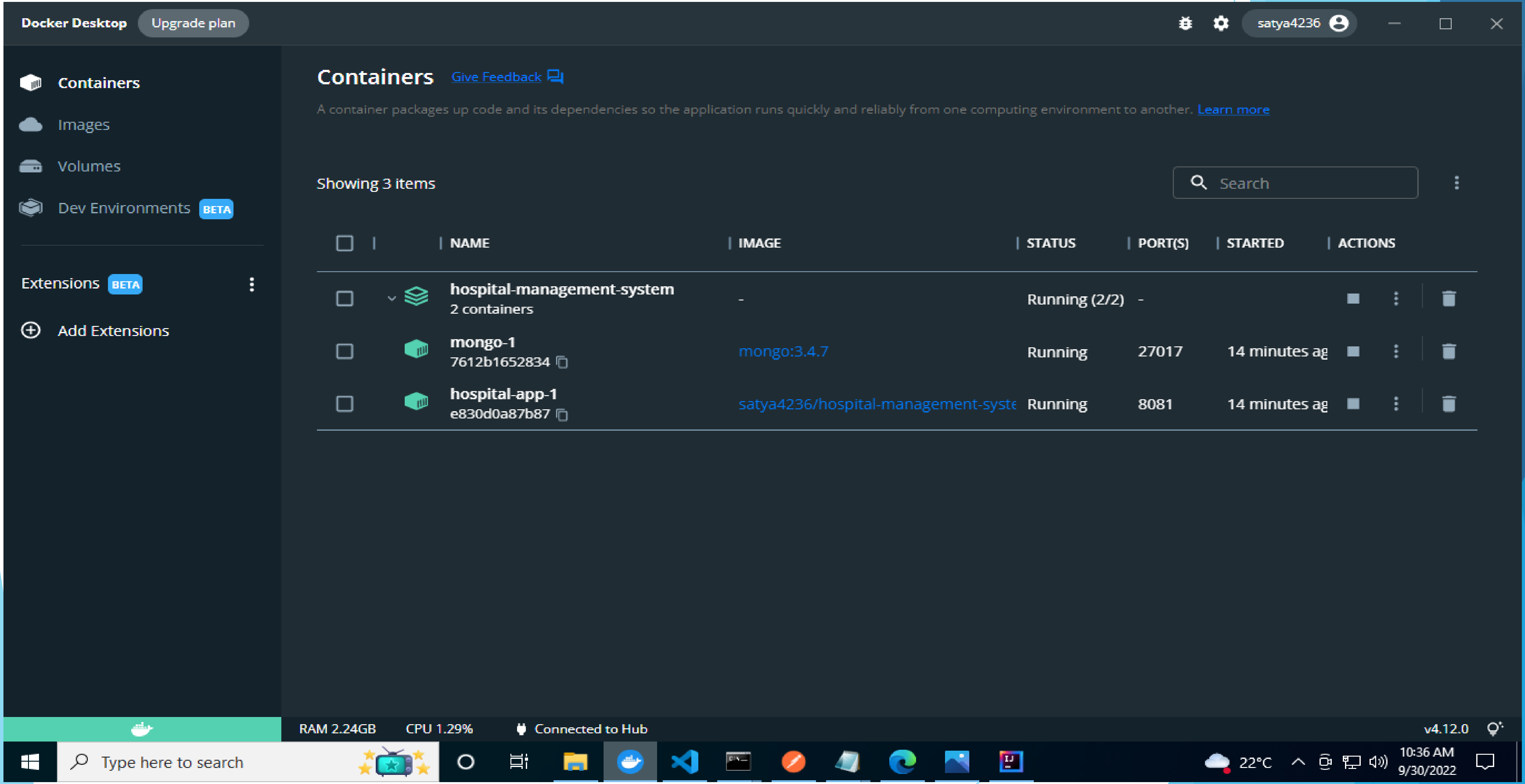
Please check your inbox to verify the email associated with this account. You won't be able to create a repository or configure your Docker Hub without verifying your email address.

satya4236 / hospital-management-system Last pushed: a day ago	Not Scanned	☆ 0	⬇ 0	Public
satya4236 / hospital-application Last pushed: 4 days ago	Not Scanned	☆ 0	⬇ 4	Public

Tip: Not finding your repository? Try a different namespace.

Create an Organization
Manage Docker Hub repositories
with your team

Docker Desktop running images in Container



Docker Desktop | Upgrade plan | satya4236

Containers | Images | Volumes | Dev Environments BETA | Extensions BETA | Add Extensions

Containers

Give Feedback

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more

Showing 3 items

NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
hospital-management-system 2 containers	-	Running (2/2)	-		
mongo-1 7612b1652834	mongo:3.4.7	Running	27017	14 minutes ag	
hospital-app-1 e830d0a87b87	satya4236/hospital-management-syste	Running	8081	14 minutes ag	

RAM 2.24GB CPU 1.29% Connected to Hub v4.12.0

Type here to search 22°C 10:36 AM 9/30/2022

Thank you