

A decorative vertical bar on the left side of the slide, featuring a light brown background with several thin, vertical orange lines. To the right of these lines are several orange circles of varying sizes, some of which have a slight drop shadow.

HOSPITAL MANAGEMENT SYSTEM

**Submitted by -
Mopidevi Deepthi**

INTRODUCTION

- Hospital Management System brings together all the information and processes of a hospital, in a single platform.
- Hospital management system is the inevitable part of the lifecycle of the modern medical institution. It automates numerous daily operations and enables smooth interactions of the users.
- The system automatically generates a highly-efficient process and makes it quick.



PROJECT BRIEFING

- The application allows
 - Patient to book their appointments and can view.
 - Doctors look into the appointments they had appointed.
- Doctors posts and gets the prescription.
- Patients gets the prescription without any delay.
- One of the main requirement here is Security.



MICROSERVICE

- Single microservice
- Consists of two schemas
 - Appointment
 - Prescription
- Patient can book an appointment by passing required details.

Ex :

localhost:8083/patient/bookappointment

- Based on patient situation, a prescription can post by doctor.

Ex : localhost:8083/prescription/save

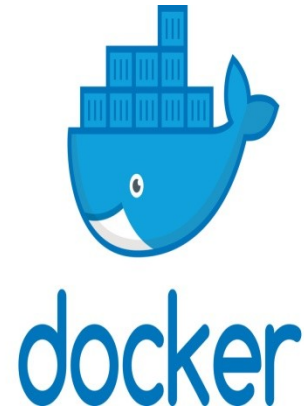


- Doctor and patient can view their respective appointments.
Ex : localhost:8083/doctor/appointments
Ex : localhost:8083/patient/myappointments
- Finally, patient receives his prescription.
Ex : localhost:8083/prescription/view



TOOLS USED

- Apache Maven
- Java
- Docker
- Spring Boot
- Mongo DB

The Apache Maven logo, featuring the word "Maven" in a bold, black, sans-serif font. A small, stylized feather icon is positioned between the 'v' and 'e'.The Spring Boot logo, featuring a green square background. On the left is a white icon of a leaf or a drop. To the right of the icon, the words "spring" and "Boot" are written in a white, sans-serif font, with "spring" on the top line and "Boot" on the bottom line.

APACHE MAVEN

- Maven is a project management and comprehension tool that provides developers a complete build lifecycle framework. Development team can automate the project's build infrastructure in almost no time as Maven uses a standard directory layout and a default build lifecycle.
- Maven uses Convention over Configuration, which means developers are not required to create build process themselves.
- Developers do not have to mention each and every configuration detail. Maven provides sensible default behavior for projects.



DOCKER

- *Docker* is an open source platform that enables developers to build, deploy, run, update and manage *containers*—standardized, executable components that combine application source code with the operating system libraries and dependencies required to run that code in any environment.
- *Docker container* starts with a simple text file containing instructions for how to build the Docker container image.
- *Docker images* contain executable application source code as well as all the tools, libraries, and dependencies that the application code needs to run as a container.



- Some docker commands –
 - `mvn install dockerfile:build`
 - `docker ps`
 - `docker images`
 - `docker-compose -f docker-compose-mongo.yml up -d`

- Docker image –

Ex: `deepthi007/hospital-management-system:0.0.1-SNAPSHOT`



JAVA

- Java is a programming language and a platform. Java is a high level, robust, object-oriented and secure programming language.
- A general-purpose programming language made for developers to write once run anywhere that is compiled Java code can run on all platforms that support Java.



SPRING BOOT

- Spring Boot is a project that is built on the top of the Spring Framework. It provides an easier and faster way to set up, configure, and run both simple and web-based applications.
- It is a Spring module that provides the *Rapid Application Development* feature to the Spring Framework. It is used to create a stand-alone Spring-based application that you can just run because it needs minimal Spring configuration.
- In Spring Boot, there is no requirement for XML configuration (deployment descriptor). It uses convention over configuration software design paradigm that means it decreases the effort of the developer.



MONGO DB

- MongoDB is a cross-platform, document oriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document.
- Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not enforce a schema.
- A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.



POSTMAN

- Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster. A comprehensive set of tools that help accelerate the API Lifecycle - from design, testing, documentation, and mocking to discovery.



Recycle Bin

hospital-management-system [src > main > java > com > mongo > app > controller > PatientController]

File Edit View Navigate Code Refactor Build Run Tools Git Window Help

Search Postman

Sign In Create Account

Working locally in Scratch Pad. Switch to a Workspace

Home Workspaces Explore

Scratch Pad

New Import Overview

POST localhost GET localhost GET localhost POST localhost GET localhost + ... No Environment

Collections

+ ...

APIs

Environments

Mock Servers

Monitors

History

You don't have any collections

Collections let you group related requests, making them easier to access and run.

Create Collection

localhost:8083/patient/bookapointme

POST localhost:8083/patient/bookapointme

Send

Body

raw JSON Beautify

```
1 {
2   "appointmentId": "1",
3   "patientName": "pat1",
4   "doctorName": "doc1",
5   "date": "7 sept",
6   "prescription": null
7 }
```

200 OK 49 ms 419 B Save Response

Pretty JSON

```
1 {
2   "appointmentId": 1,
3   "patientName": "pat1",
4   "doctorName": "doc1",
5   "date": "7 sept",
6   "prescription": null
7 }
```

Code snippet

cURL

```
1 curl --location --request POST
2   'localhost:8083/patient/
3   bookapointment' \
4   --header 'Authorization: Basic
5   cGF0aWVudDpwYXNzd29yZA==' \
6   --header 'Content-Type: application/json'
7   \
8   --header 'Cookie:
9   JSESSIONID=7EA72A8F58367C40580790FC781
10  0A817' \
11  --data-raw '{
12    "appointmentId": "1",
13    "patientName": "pat1",
14    "doctorName": "doc1",
15    "date": "7 sept",
16    "prescription": null
17  }'
```

Windows Taskbar

Type here to search

24°C Rain showers

ENG 11:13 AM

INTL 9/13/2022

Recycle Bin

Cywin64 Terminal

Git Bash

IntelliJ IDEA Community Edition

Microsoft Edge

Docker Desktop

MongoDB Compass

Postman

hospital-management-system [C:\Users\deepthi\Mini Project\hospital-management-system] - PatientController.java

File Edit View Navigate Code Refactor Build Run Tools Git Window Help

src \ main \ java \ com \ mongo \ app \ controller \ PatientController

All in hospital-management-system (4)

Git

Sign In Create Account

Home Workspaces Explore

Search Postman

Working locally in Scratch Pad. Switch to a Workspace

Scratch Pad New Import Overview POST localhost GET localhost GET localhost POST localhost GET localhost

Collections

APIs

Environments

Mock Servers

Monitors

History

You don't have any collections

Collections let you group related requests, making them easier to access and run.

Create Collection

localhost:8083/prescription/view?patientName=pat1

GET localhost:8083/prescription/view?patientName=pat1

Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION
patientName	pat1	
Key	Value	Description

Body Cookies (1) Headers (10) Test Results

200 OK 53 ms 764 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "prescriptionId": 1,
3   "appointmentId": 1,
4   "description": "prescription1",
5   "patientName": "pati",
6   "doctorName": "doc1"
7 }
8
```

Find and Replace Console

Runner Trash

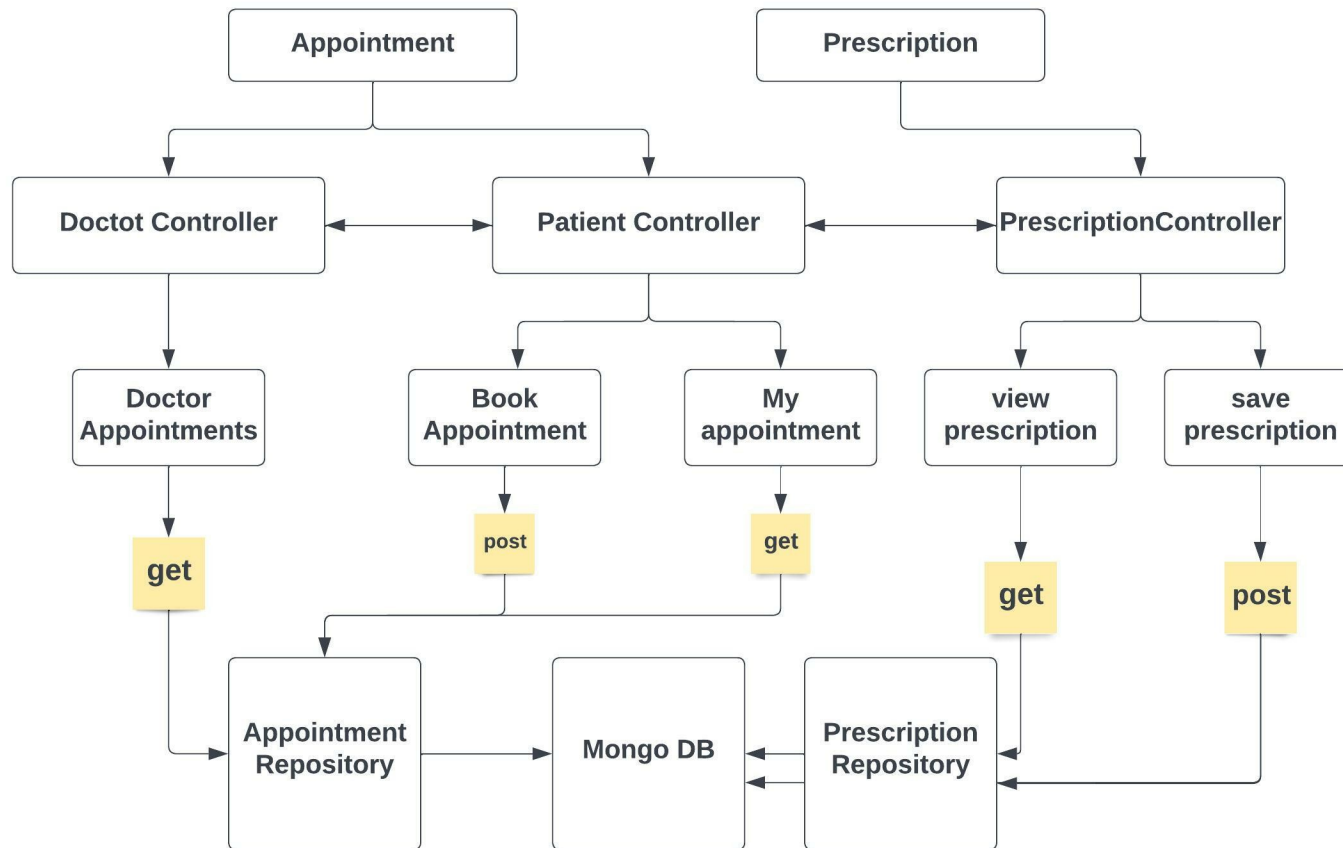
Type here to search

24°C Rain showers

ENG 11:15 AM

INTL 9/13/2022

ARCHITECTURE



USE CASE

Use case 1

- Patients to keep thier records of all appointments.
- Also useful for the patients who are unable to go to hospital.

Use case 2

- Hospital management to have records of all the appointments as well as prescription given by doctor along with patient details.



TESTING

- The tool used for testing is SonarLint.
- The code coverage is-

Class	-	100%
Method	-	81%
Line	-	86%



SWAGGER

The screenshot displays the Swagger UI interface in a web browser. The URL bar shows `localhost:8083/swagger-ui/index.html#/patient-controller/book`. The main content area is titled **POST /patient/bookappointment**. Below the title, there is a **Parameters** section with the text "No parameters" and buttons for **Cancel** and **Reset**. The **Request body** section is marked as **required** and has a dropdown menu set to `application/json`. The request body is a JSON object:

```
{  "appointmentId": 1,  "patientName": "pat1",  "doctorName": "doc1",  "date": "7 Sept",  "prescription": {    "prescriptionId": 1,    "appointmentId": 1,    "description": "prescription1",    "patientName": "pat1",    "doctorName": "doc1"  } }
```

Below the request body, there is a blue **Execute** button and a **Clear** button. The **Responses** section is currently empty. At the bottom, a **Curl** section shows the command:

```
curl -X 'POST' \
```

The Windows taskbar at the bottom shows the system clock as 11:30 AM on 9/13/2022, with the language set to ENG and the currency as Indian Rupee.

Swagger UI

localhost:8083/swagger-ui/index.html#/patient-controller/book

Curl

```
curl -X 'POST' \
  'http://localhost:8083/patient/bookappointment' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "appointmentId": 1,
    "patientName": "pat1",
    "doctorName": "doc1",
    "date": "7 Sept",
    "prescription": {
      "prescriptionId": 1,
      "appointmentId": 1,
      "description": "prescription1",
      "patientName": "pat1",
      "doctorName": "doc1"
    }
  }'
```

Request URL

http://localhost:8083/patient/bookappointment

Server response

Code Details

200

Response body

```
{
  "appointmentId": 1,
  "patientName": "pat1",
  "doctorName": "doc1",
  "date": "7 Sept",
  "prescription": {
    "prescriptionId": 1,
    "appointmentId": 1,
    "description": "prescription1",
    "patientName": "pat1",
    "doctorName": "doc1"
  }
}
```

Response headers

```
cache-control: no-cache,no-store,max-age=0,must-revalidate
connection: keep-alive
content-type: application/json
```

Type here to search

Indian Rupee

ENG 11:30 AM
INTL 9/13/2022

Thank
you!!

