

## L1 – SPRING BOOT & MICROSERVICES LAB ASSESSMENT

Name: Nagarjun N S

Emp ID: 12228

1. Batch: Java Batch-5 Duration: 1 Day [8 Hours] Please add the below notes to all the Questions. Note: 1. Perform W3H Analysis and write the Algorithm and Flowchart as per the given scenario. 2. Draw the UML diagrams for the given scenario. 3. Create a web-based application for the given scenario using Three Tier Architecture. 4. Use React JS as a Front End. Use Spring Boot as a Back End and use REST API as an Architecture Style. 5. Create a Database design as per the given scenario and use MySQL as Database. 6. Perform all the CRUD operations using Spring Data JPA [CRUD Repository]. 7. Make the application as user interactive. Apply style using CSS / Bootstrap. And perform the validations. 8. Maintain the Code Quality and Coding Standard.

Question:

**Q3) Hospital Management System:** Hospital Management System helps the professionals in healthcare to check out the clinical documents, diagnoses, patient records, and other relevant things in a single view and therefore results in the healthcare professionals making decisions at the right time.

**User Stories to be Developed:** Perform the CRUD operations between Hospital and Doctor tables

## W3H:

PROJECT NAME: Hospital Management System	
What? 1	How? 2
<b>1. What are the modules required?</b> <b>Ans:</b> a) Admin (System) Module b) Doctor Module c) Hospital module	<b>1) Admin (System):</b> <b>a) Login:</b> <b>Method 1:</b> The admin can login using their adminID, Email and password. <b>Method 2:</b> The admin can login using their username and password. <b>Method 3:</b> The admin can login to their account using social media accounts.
<b>2. What are the responsibilities of Admin (System)?</b> <b>Ans:</b> a) Admin (System) will handle all the CRUD operations of the Hospital details and the Doctor details.	<b>b) Adding Doctors:</b> <b>Method 1:</b> Admin can also add Doctors by using the DoctorID, Doctor Name, Specialization, Experience. <b>Method 2:</b> Admin can add Tenders with their DoctorID, Doctor Name.
<b>3. What can the Doctor do?</b> <b>Ans:</b> a) The Doctor can be able to update their personal details. b) Doctor can be able to perform the CRUD operation on their details. c) Doctor can be able to see their profile and can update it. f) Doctor can register themselves by providing details of their doctor id, name, ID card for verification	<b>c) Update / delete / view Doctor Details:</b> <b>Method 1:</b> Admin can be able to Update / delete / view Doctor details by providing separate buttons with appropriate details. <b>Method 2:</b> Admin can be able to Update / delete / view Doctor details by using the drop-down list with appropriate details.
<b>4. What can the Doctor see in their profile tab?</b> <b>Ans:</b> a) Doctor can be able to see their personal details. b) Doctors can be able to see their patient attended history. c) They can also check the revisit date of the patient, if they are updated.	<b>d) CRUD on Hospital Table:</b> <b>Method 1:</b> Admin can be able to Create/ Update / delete / view Hospital details by providing separate buttons. <b>Method 2:</b> Admin can be able to Create/ Update / delete / view Hospital details by using the drop-down list.
<b>5. What happens when the revisit date of patient is arrived?</b> <b>Ans:</b> Once the revisit date is arrived, the patient is informed via SMS and the application notification, and they can come to visit the doctor.	<b>2) Doctor:</b> <b>a) Login:</b> <b>Method 1:</b> The Doctor can login by using their email and password. <b>Method 2:</b> The Doctor can login by using mobile number and name and customer ID. <b>Method 3:</b> The Doctor can login to their account using social media accounts.
<b>6. How to make the payment?</b> <b>Ans:</b> Once the appointment is confirmed, then the user must pay a 50% of fee via NEFT or other online payment methods, because the doctor will be allocated to the patient on that time.	<b>b) CRUD Operations:</b> <b>Method 1:</b> Doctor can perform CRUD Operations for their account settings and their personal details with separate buttons. <b>Method 2:</b> Doctor can perform CRUD Operations for their account settings and personal details using a drop-down list and a confirm button.
<b>7. What about Hospital module?</b> <b>Ans:</b> The hospital module consists of only the hospital details and the doctors allocated to the hospitals.	<b>c) Check Patient info:</b> <b>Method 1:</b> Doctor can be able to check their Patient details like revisit date, etc. using the "Patient Info" option in the menu. <b>Method 2:</b> Doctor can check their Patient info by calling the Compounder every time.

<p><b>1) Admin (System):</b></p> <p><b>a) Login:</b></p> <p><b>Method 1:</b> The admin can login using their adminID, Email and password. (This method is used because the originality of the admin is verified).</p> <p><b>b) Adding Doctors:</b></p> <p><b>Method 1:</b> Admin can also add Doctors by using the DoctorID, Doctor Name, Specialization, Experience. (This method is used because the admin can add the Doctors by providing the Doctor details.)</p> <p><b>c) Update / delete / view Doctor Details:</b></p> <p><b>Method 1:</b> Admin can be able to Update / delete / view Doctor details by providing separate buttons with appropriate details. (Providing separate buttons with separate functionalities will help to use the application easily.)</p> <p><b>d) CRUD on Hospital Table:</b></p> <p><b>Method 1:</b> Admin can be able to Create/ Update / delete / view Hospital details by providing separate buttons. (Providing separate buttons with separate functionalities will help to use the application easily. And, yes, Admin only can be able to change the details of a hospital)</p> <p><b>2) Doctor:</b></p> <p><b>a) Login:</b></p> <p><b>Method 1:</b> The Doctor can login by using their email and password. (Login to the system by using their email and password will be a hassle-free experience and results in a secure way of authentication. Because the email ID is already a unique and verified one.)</p> <p><b>b) CRUD Operations:</b></p> <p><b>Method 1:</b> Doctor can perform CRUD Operations for their account settings and their personal details with separate buttons. (Providing separate buttons with separate functionalities will help to use the application easily.)</p> <p><b>c) Check Patient info:</b></p> <p><b>Method 1:</b> Doctor can be able to check their Patient details like revisit date, etc. using the “Patient Info” option in the menu. (By providing a separate tracking status for each patient will also help the doctor to access Patient details easily and the doctor doesn't need to bother by thinking questions like “what happened to the patient?”,etc... By this way, the tracking system will be very much efficient for doctors.)</p>	<p><b>1) Admin:</b></p> <p><b>a) Login:</b></p> <p><b>Method 2:</b> The admin can login using their username and password. (Admin can't login using their username and password as the originality is not verified.)</p> <p><b>Method 3:</b> The admin can login to their account using social media accounts. (Using Social media accounts to login will not be a feasible way and it is less secure).</p> <p><b>b) Adding Doctors:</b></p> <p><b>Method 2:</b> Admin can add Tenders with their DoctorID, Doctor Name. (If a Doctor is added using only the basic details like Doctor ID, Doctor name then these details are not sufficient for admin to check the originality.)</p> <p><b>c) Update / delete / view Doctor Details:</b></p> <p><b>Method 2:</b> Admin can be able to Update / delete / view Doctor details by using the drop-down list with appropriate details. (The drop-down list feature is not feasible, and it is hard to use, and it is time-consuming function too, because it includes two-clicks to operate a function).</p> <p><b>d) CRUD on Hospital module:</b></p> <p><b>Method 2:</b> Admin can be able to Create/ Update / delete / view Hospital details by using the drop-down list. (The drop-down list feature is not feasible, and it is hard to use, and it is time-consuming function too. Every time, the user must make 2 clicks for confirming an action.)</p> <p><b>2) Doctor:</b></p> <p><b>a) Login:</b></p> <p><b>Method 2:</b> The Doctor can login by using mobile number and name and customer ID. (Doctor logging in using mobile number, name and DoctorID will not be much secure as email and password)</p> <p><b>Method 3:</b> The Doctor can login to their account using social media accounts. (Using social media accounts to login will be a security threat. Many personal details will be provided while registering an account. Those details might have a chance to be hacked, and leaked and unauthorized usage)</p> <p><b>b) CRUD Operations:</b></p> <p><b>Method 2:</b> Doctor can perform CRUD Operations for their account settings and personal details using a drop-down list and a confirm button. (The drop-down list feature is not feasible, and it is hard to use, and it is time-consuming function too).</p> <p><b>c) Check Patient info:</b></p> <p><b>Method 2:</b> Doctor can check their Patient info by calling the Compounder every time. (If a doctor needs to call the compounder to track the Patient info and history, then there will be a huge time wasting of both doctor and compounder in the way of manually searching the Attendance note. So, it is not feasible.)</p>
Why? 3	Why not? 4

**ALGORITHMS:**

**ADMIN:**

Step 1: Start  
Step 2: Go to Login page  
Step 3: Enter the credentials  
Step 4: After completing step 3, click login  
Step 5: Go to homepage  
Step 6: Click on “Add Doctors”  
Step 7: Provide Doctor Details and submit  
Step 8: Logout  
Step 9: Stop

**Doctor:**

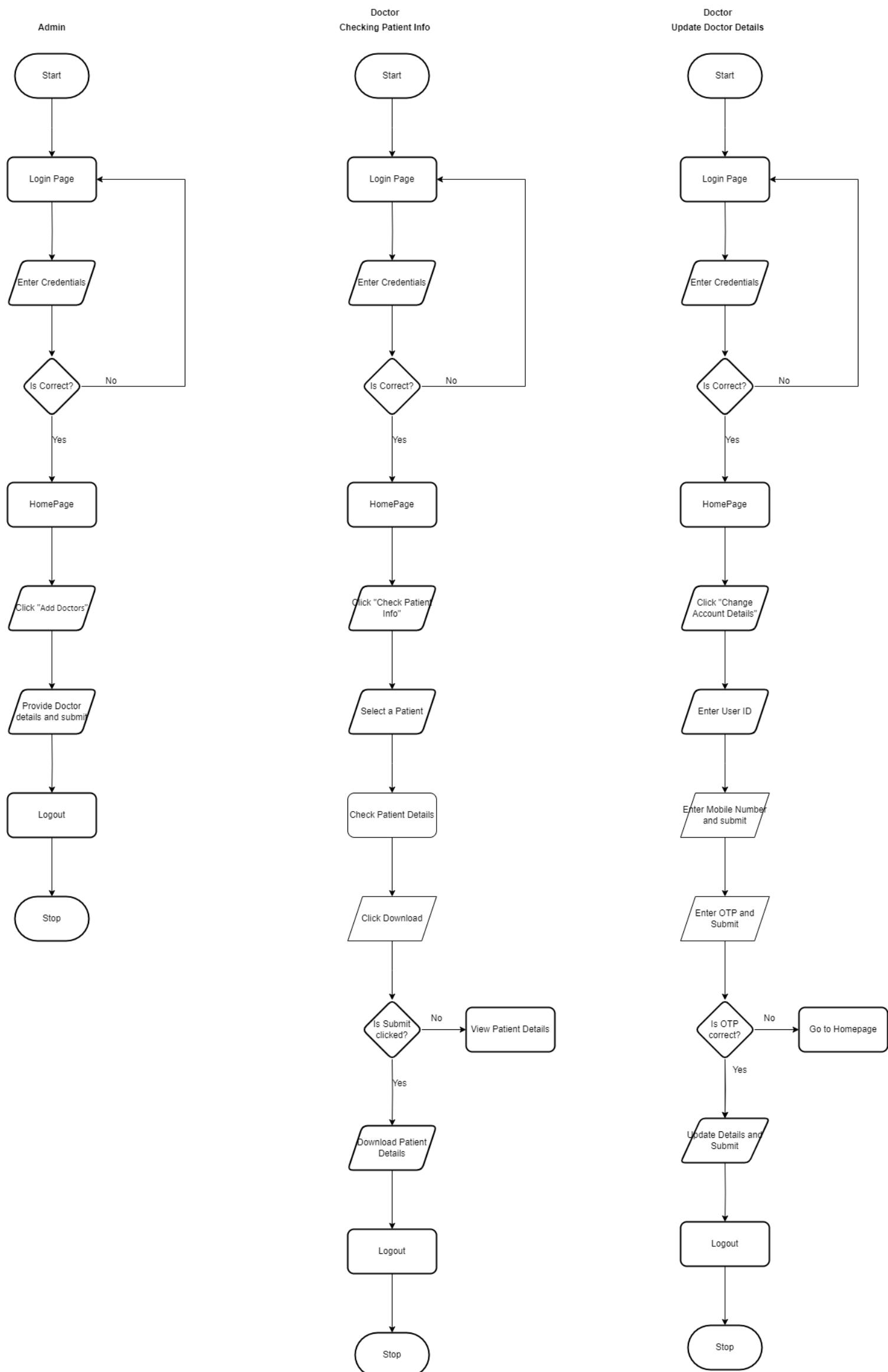
**Checking Patient Info:**

step 1: Start  
step 2: Login by entering your credentials  
step 3: Go to homepage  
step 4: Select “Patient info” option  
step 5: Select a Patient  
step 6: Check the patient details  
step 7: Download Patient details  
step 8: Logout  
step 9: stop

**Update Doctor Details:**

step 1: Start  
step 2: Login by entering your credentials  
step 3: Go to homepage  
step 4: select “Change Account Details”  
step 5: Enter Doctor ID  
step 6: Enter your Mobile number  
step 7: Enter OTP  
step 8: Click Submit  
step 9: Update Account details  
step 10: Confirm it  
step 11: Success msg displayed  
step 12: Logout  
step 13: Stop

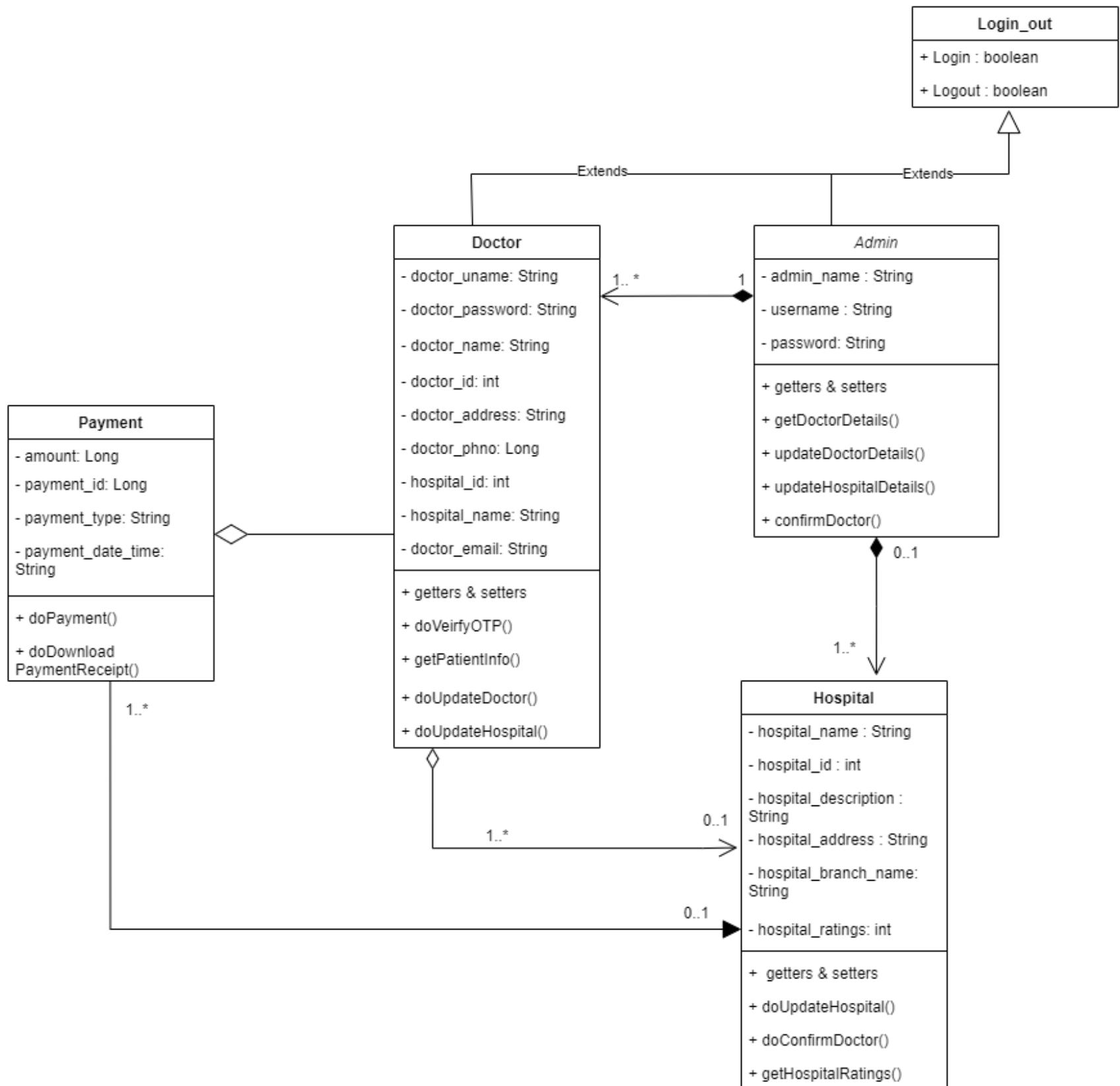
## FLOWCHARTS:



## UML DIAGRAMS:

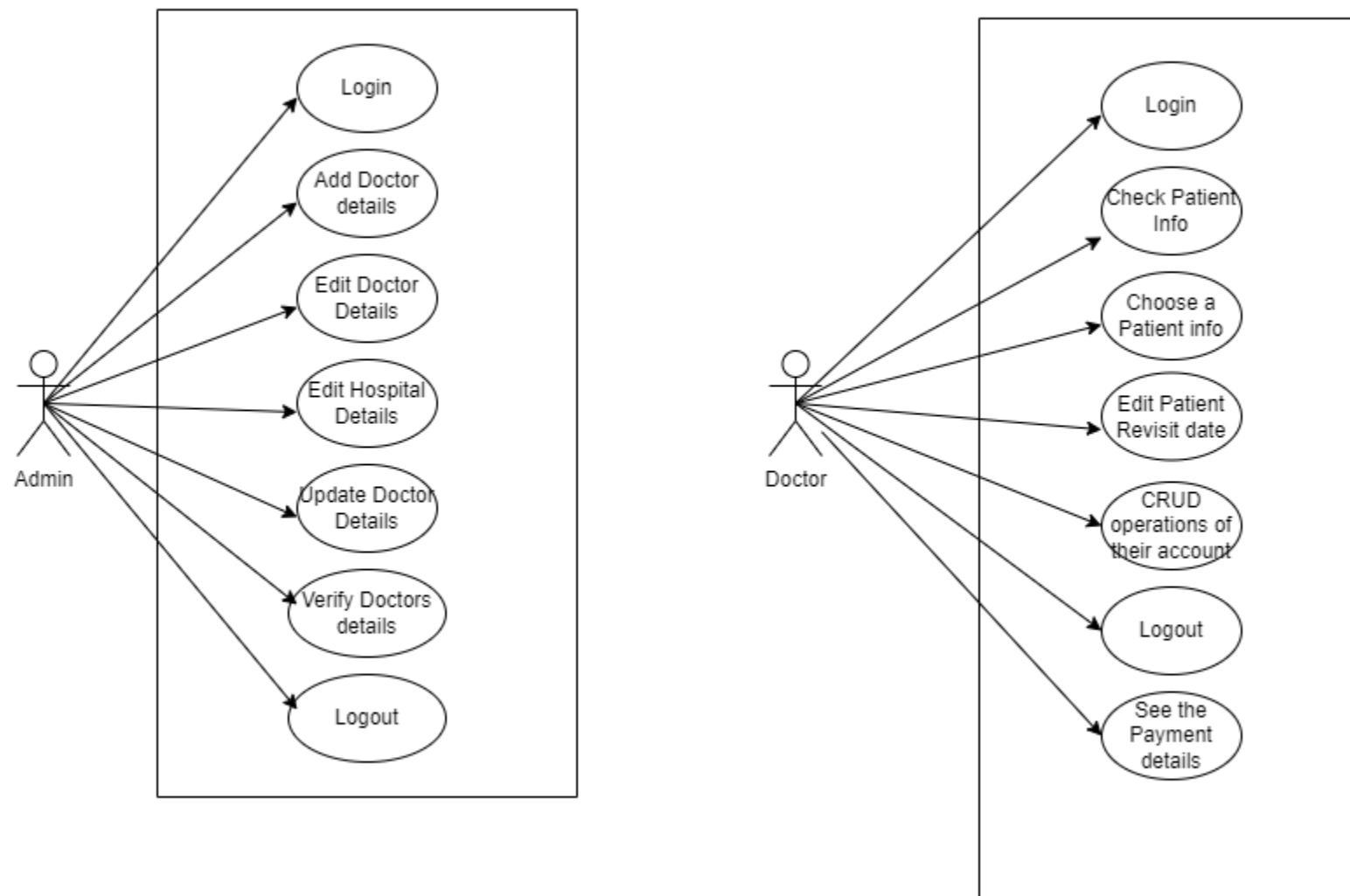
## **CLASS DIAGRAM:**

## HOSPITAL MANAGEMENT SYSTEM CLASS DIAGRAM

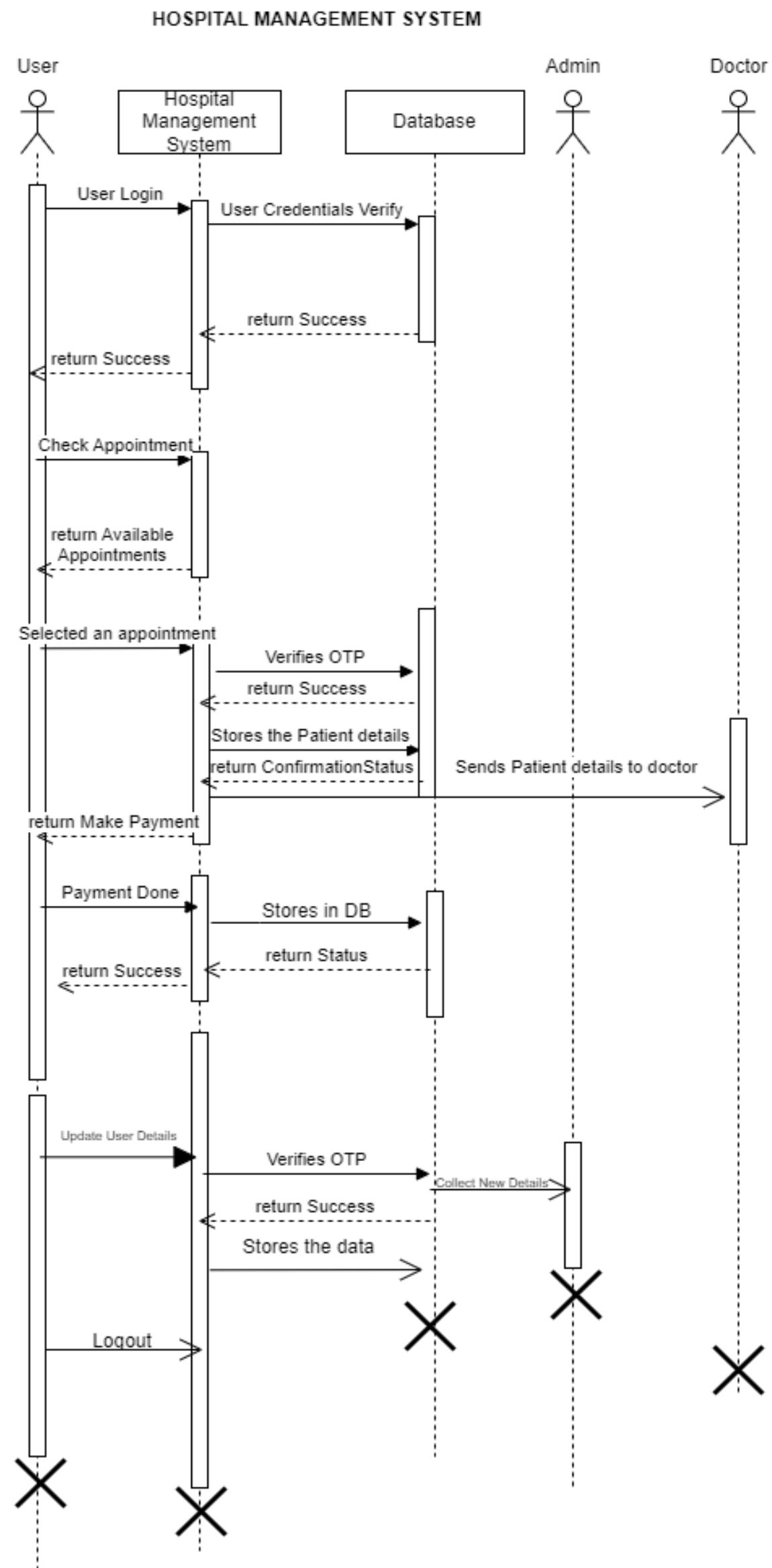


**USE CASE DIAGRAM:**

USE CASE DIAGRAM  
Hospital Management System



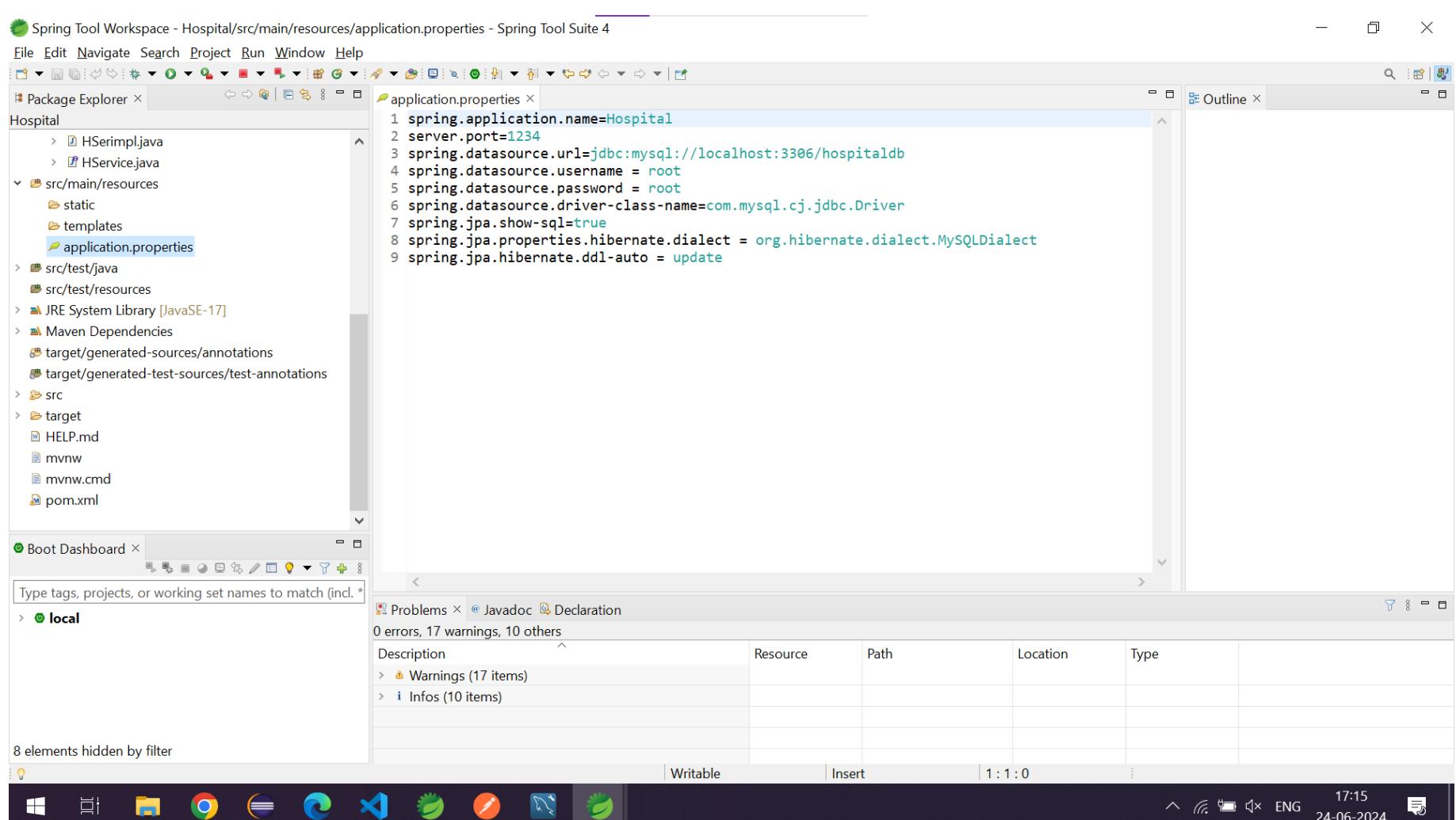
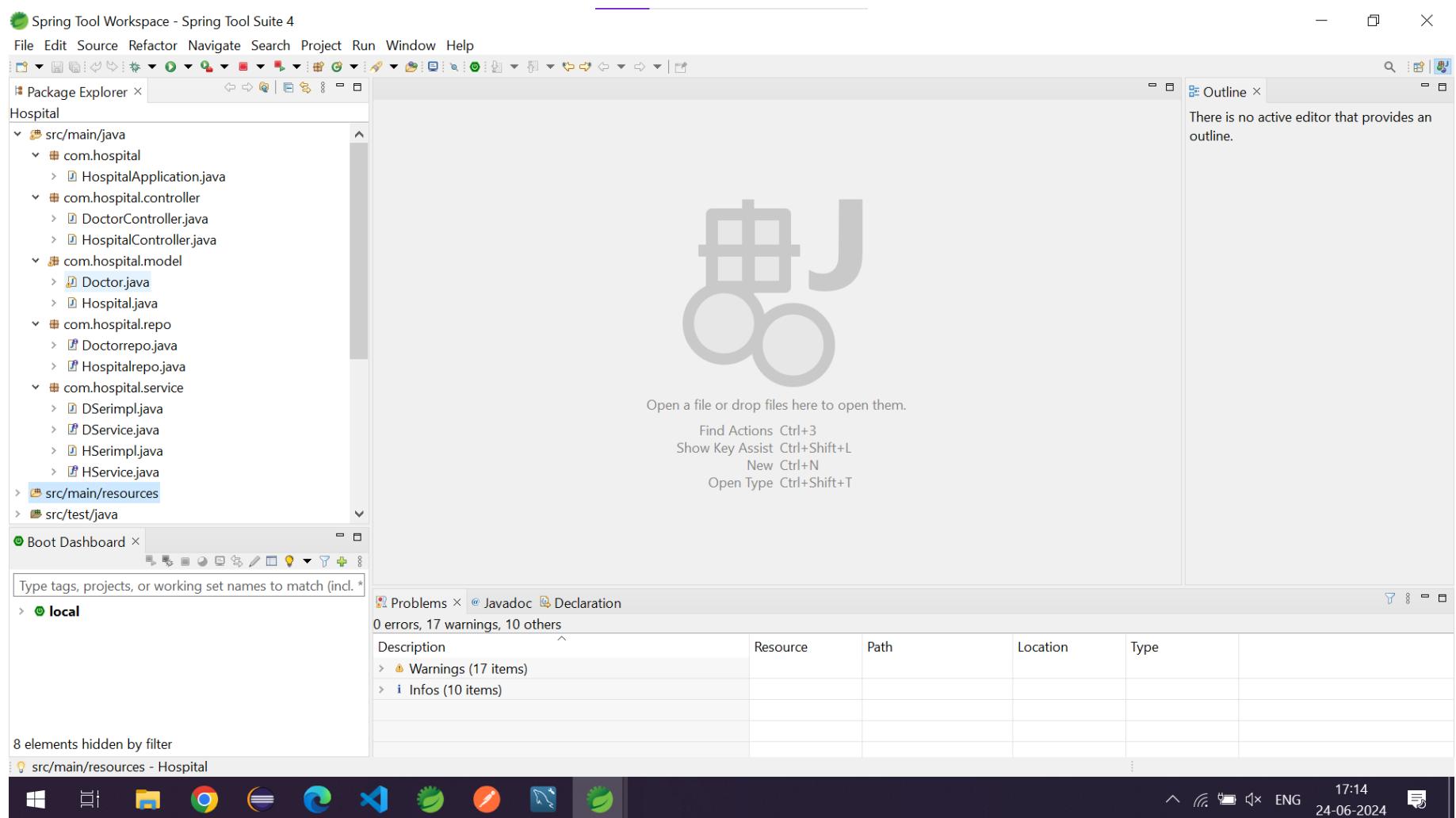
**SEQUENCE DIAGRAM:**



**FOLDER STRUCTURE:**

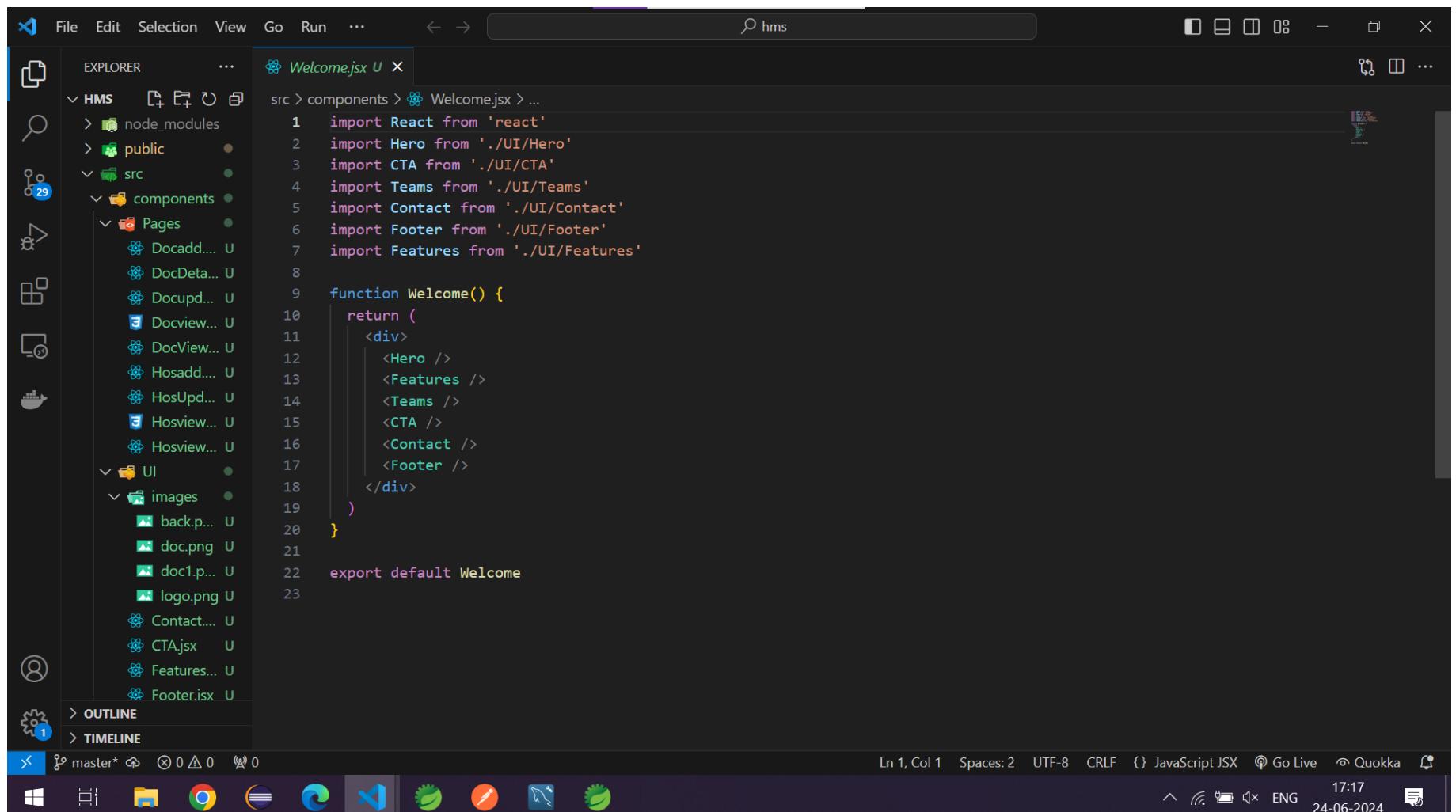
Spring Tool Suite:

## Backend Folder Structure:



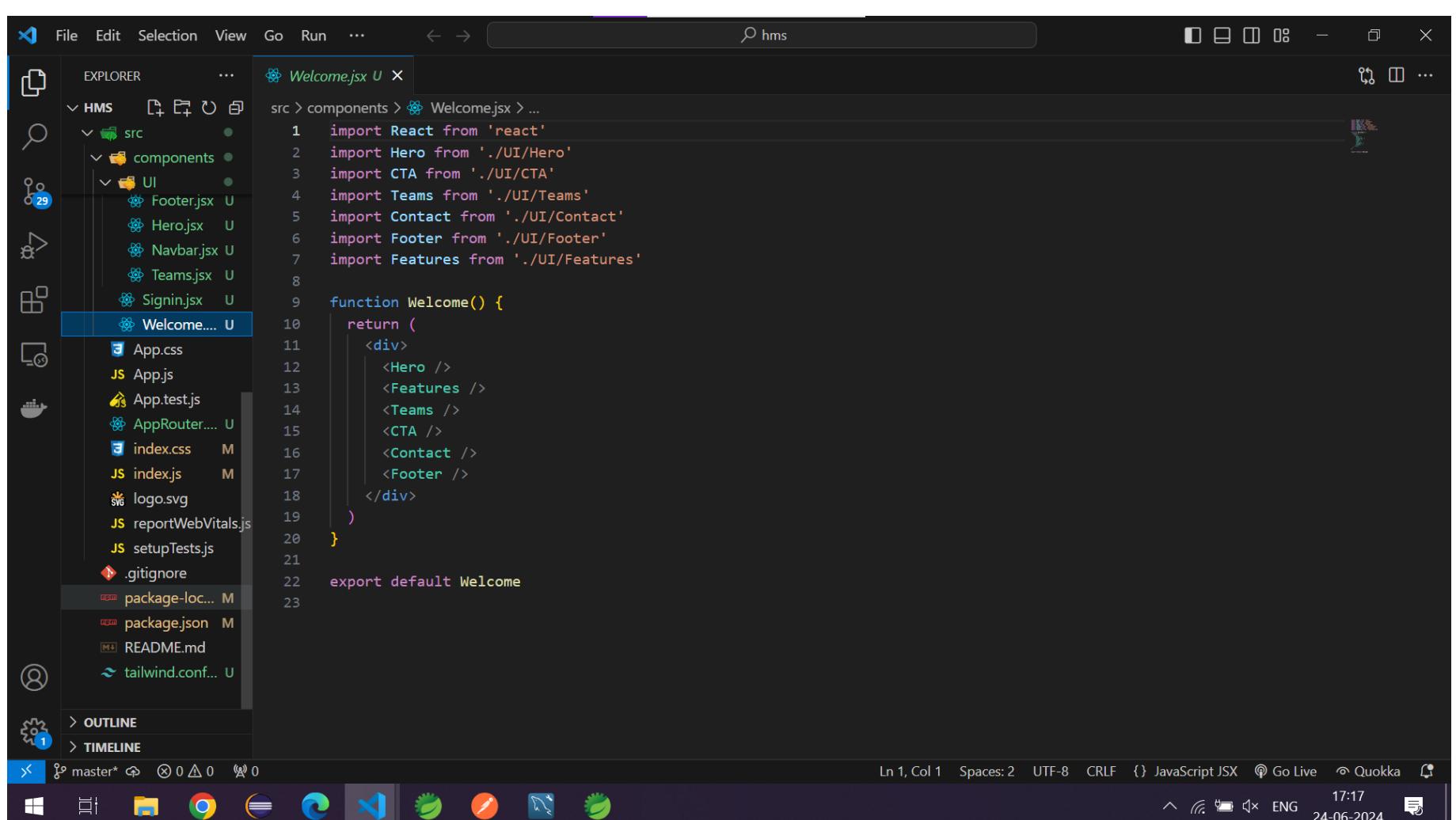
VS CODE

## VSCODE:



```
src > components > Welcome.jsx > ...
1 import React from 'react'
2 import Hero from './UI/Hero'
3 import CTA from './UI/CTA'
4 import Teams from './UI/Teams'
5 import Contact from './UI/Contact'
6 import Footer from './UI/Footer'
7 import Features from './UI/Features'

8
9 function Welcome() {
10   return (
11     <div>
12       <Hero />
13       <Features />
14       <Teams />
15       <CTA />
16       <Contact />
17       <Footer />
18     </div>
19   )
20 }
21
22 export default Welcome
23
```



```
src > components > Welcome.jsx > ...
1 import React from 'react'
2 import Hero from './UI/Hero'
3 import CTA from './UI/CTA'
4 import Teams from './UI/Teams'
5 import Contact from './UI/Contact'
6 import Footer from './UI/Footer'
7 import Features from './UI/Features'

8
9 function Welcome() {
10   return (
11     <div>
12       <Hero />
13       <Features />
14       <Teams />
15       <CTA />
16       <Contact />
17       <Footer />
18     </div>
19   )
20 }
21
22 export default Welcome
23
```

**CODING:**

**BACKEND CODE:**

**HospitalApplication.java:**

```
package com.hospital;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class HospitalApplication {

    public static void main(String[] args) {
        SpringApplication.run(HospitalApplication.class, args);
    }

}
```

### **CONTROLLER:**

#### **HospitalController.java**

```
package com.hospital.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.hospital.model.Hospital;
import com.hospital.service.HSerimpl;

@RestController
@RequestMapping("/hospital")
@CrossOrigin("*")
public class HospitalController {

    @Autowired
    HSerimpl service;
```

```
@PostMapping  
public String addHos(@RequestBody Hospital hos){  
String msg="";  
try{  
service.addHosp(hos);  
msg="Success";  
}  
catch(Exception e){  
e.printStackTrace();  
msg="Failure";  
}  
return msg;  
}
```

```
@GetMapping("{id}")  
public Hospital getHosp(@PathVariable("id") int id){  
return service.getHosp(id);  
}
```

```
@PutMapping  
public String upDocbyid(@RequestBody Hospital hos){  
String msg="";  
try{  
service.upHosp(hos);  
msg="Success";  
}  
catch(Exception e){  
e.printStackTrace();  
msg="Failure";  
}  
return msg;  
}
```

```
@GetMapping("/all")  
public List<Hospital> getAllDoc(){  
return service.getAllHospitals();  
}
```

```
@DeleteMapping("{id}")
public String delDoc(@PathVariable("id") int id) {
String msg="";
try{
service.delHosp(id);
msg="Success";
}
catch(Exception e){
e.printStackTrace();
msg="Failure";
}
return msg;
}
```

**DoctorController.java:**

```
package com.hospital.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.hospital.model.Doctor;
import com.hospital.service.DSerimpl;

@RestController
@CrossOrigin("*")
@RequestMapping("/doctor")
public class DoctorController {
```

```
@Autowired
DSerimpl service;

@PostMapping
public String addDoc(@RequestBody Doctor doc) {
String msg="";
try{
service.addDoc(doc);
msg="Success";
}
catch(Exception e){
e.printStackTrace();
msg="Failure";
}
return msg;
}

@GetMapping("{id}")
public Doctor getDocbyid(@PathVariable("id") int id) {
return service.getDoc(id);
}

@PutMapping
public String upDocbyid(@RequestBody Doctor doc) {
String msg="";
try{
service.upDoc(doc);
msg="Success";
}
catch(Exception e){
e.printStackTrace();
msg="Failure";
}
return msg;
}

@GetMapping("/all")
```

```

public List<Doctor> getAllDoc(){
    return service.viewAllDoc();
}

{@DeleteMapping("{id}"})
public String delDoc(@PathVariable("id") int id) {
    String msg="";
    try {
        service.delDoc(id);
        msg="Success";
    }
    catch(Exception e){
        e.printStackTrace();
        msg="Failure";
    }
    return msg;
}

```

**Model:**

**Doctor.java:**

```

package com.hospital.model;

import jakarta.persistence.CascadeType;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.ManyToOne;
import jakarta.persistence.OneToOne;
import jakarta.persistence.Table;

@Entity
@Table(name = "Doctor")
public class Doctor {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

```

```
@Column(name = "Doctor_ID")
private int did;

@Column(name = "Doctor_Name")
private String dname;

@Column(name = "Doctor_Experience")
private String dexperience;

@Column(name = "Specialization")
private String dspecial;

@ManyToOne(cascade = CascadeType.MERGE,targetEntity = Hospital.class)
@JoinColumn
private Hospital hospital;

public Doctor() {
    super();
    // TODO Auto-generated constructor stub
}

public Doctor(int did, String dname, String dexperience, String dspecial, Hospital hospital) {
    super();
    this.did = did;
    this.dname = dname;
    this.dexperience = dexperience;
    this.dspecial = dspecial;
    this.hospital = hospital;
}

@Override
public String toString() {
    return "Doctor [did=" + did + ", dname=" + dname + ", dexperience=" + dexperience + ", dspecial=" +
        dspecial
        + ", hospital=" + hospital + "]";
}

public int getDid() {
    return did;
}

public void setDid(int did) {
```

```
this.did = did;
}

public String getDname() {
    return dname;
}

public void setDname(String dname) {
    this.dname = dname;
}

public String getDexperience() {
    return dexperience;
}

public void setDexperience(String dexperience) {
    this.dexperience = dexperience;
}

public String getDspecial() {
    return dspecial;
}

public void setDspecial(String dspecial) {
    this.dspecial = dspecial;
}

public Hospital getHospital() {
    return hospital;
}

public void setHospital(Hospital hospital) {
    this.hospital = hospital;
}

}
```

### Hospital.java:

```
package com.hospital.model;
```

```
import org.springframework.web.bind.annotation.CrossOrigin;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;

@Entity
@CrossOrigin("*")
public class Hospital {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "Hospital_ID")
    private int hid;

    @Column(name = "Hospital_Name")
    private String hname;

    @Column(name = "City")
    private String hcity;

    @Column(name = "Rating")
    private int hrating;

    public Hospital() {
        super();
        // TODO Auto-generated constructor stub
    }

    public Hospital(int hid, String hname, String hcity, int hrating) {
        super();
        this.hid = hid;
        this.hname = hname;
        this.hcity = hcity;
        this.hrating = hrating;
    }

    @Override
    public String toString() {
        return "Hospital [hid=" + hid + ", hname=" + hname + ", hcity=" + hcity + ", hrating=" + hrating + "]";
    }
}
```

```
public int getHid() {
    return hid;
}

public void setHid(int hid) {
    this.hid = hid;
}

public String getHname() {
    return hname;
}

public void setHname(String hname) {
    this.hname = hname;
}

public String getHcity() {
    return hcity;
}

public void setHcity(String hcity) {
    this.hcity = hcity;
}

public int getHrating() {
    return hrating;
}

public void setHrating(int hrating) {
    this.hrating = hrating;
}

}
```

**Repo:**

Doctorrepo.java

package com.hospital.repo;

```
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.hospital.model.Doctor;

@Repository
public interface Doctorrepo extends JpaRepository<Doctor, Integer>{
    public void deleteById(int id);
}
```

Hospitalrepo.java:

```
package com.hospital.repo;
```

```
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
```

```
import com.hospital.model.Hospital;
```

```
@Repository
public interface Hospitalrepo extends JpaRepository<Hospital, Integer>{
    public void deleteById(int id);
}
```

### **Service:**

DSerimpl.java

```
package com.hospital.service;
```

```
import java.util.List;
```

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
```

```
import com.hospital.model.Doctor;
```

```
import com.hospital.repo.Doctorrepo;
```

```
@Service
```

```
public class DSerimpl implements DService{
```

```
@Autowired
```

```
Doctorrepo repo;
```

```
@Override  
public void addDoc(Doctor doc) {  
repo.save(doc);  
}
```

```
@Override  
public void delDoc(int id) {  
repo.deleteById(id);  
}
```

```
@Override  
public void upDoc(Doctor doc) {  
repo.save(doc);  
}
```

```
@Override  
public List<Doctor> viewAllDoc() {  
List<Doctor> doclist = repo.findAll();  
return doclist;  
}
```

```
@Override  
public Doctor getDoc(int id) {  
return repo.findById(id).get();  
}  
}
```

#### **DService.java:**

```
package com.hospital.service;  
  
import java.util.List;  
  
import com.hospital.model.Doctor;  
  
public interface DService{  
public void addDoc(Doctor doc);  
public void delDoc(int id);  
public void upDoc(Doctor doc);
```

```
public List<Doctor> viewAllDoc();
public Doctor getDoc(int id);
}
```

#### **HSerimpl.java:**

```
package com.hospital.service;
```

```
import java.util.List;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Service;
```

```
import com.hospital.model.Hospital;
```

```
import com.hospital.repo.Hospitalrepo;
```

```
@Service
```

```
public class HSerimpl implements HService{
```

```
@Autowired
```

```
Hospitalrepo repo;
```

```
@Override
```

```
public void addHosp(Hospital hos) {
```

```
repo.save(hos);
```

```
}
```

```
@Override
```

```
public List<Hospital> getAllHospitals() {
```

```
List<Hospital> hoslist = repo.findAll();
```

```
return hoslist;
```

```
}
```

```
@Override
```

```
public void upHosp(Hospital hos) {
```

```
repo.save(hos);
```

```
}
```

```
@Override
```

```
public void delHosp(int id) {
```

```

repo.deleteById(id);

}

@Override
public Hospital getHosp(int id) {
return repo.findById(id).get();
}

}

```

### **HService.java:**

```

package com.hospital.service;

import java.util.List;

import com.hospital.model.Hospital;

public interface HService {
public void addHosp(Hospital hos);
public List<Hospital> getAllHospitals();
public void upHosp(Hospital hos);
public void delHosp(int id);
public Hospital getHosp(int id);
}

```

### **Application.properties:**

```

spring.application.name=Hospital
server.port=1234
spring.datasource.url=jdbc:mysql://localhost:3306/hospitaldb
spring.datasource.username = root
spring.datasource.password = root
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQLDialect
spring.jpa.hibernate.ddl-auto = update

```

### **Front-End Code:**

#### **PAGES:**

##### **Docadd.jsx:**

```

import React from 'react';
import {Link,useNavigate} from 'react-router-dom';
import { useState,useEffect } from 'react';
import axios from 'axios';
import './Docview.css';

```

```

function Docadd() {
  const [idlist, setIdlist] = useState([]);
  const [inputData, setInputData] = useState({
    did: "-1",
    dname: "",
    dexperience: "",
    dspecial: "",
    hospital: {
      hid: "",
      hname: "",
      hcity: "",
      hrating: ""
    }
  });
}

const naviget = useNavigate();

useEffect(() => {
  axios
    .get("http://localhost:1234/hospital/all")
    .then((res) => {
      setIdlist(res.data);
    })
    .catch((err) => console.log(err));
}, []);

let handleidchange = (e) => {
  const selectedRecordId = e.target.value;
  console.log(selectedRecordId);
  if (selectedRecordId) {
    axios
      .get("http://localhost:1234/hospital/" + selectedRecordId)
      .then((response) => {
        console.log(response.data);
        setInputData({ ...inputData, hospital: response.data });
      })
      .catch((err) => {
        console.log(err);
      });
  }
}

let handleSubmit = (e) => {
  e.preventDefault();
  let result = validateValues(inputData);
  if (result === true) {
    console.log(inputData)
    axios
      .post("http://localhost:1234/doctor", inputData)
      .then((res) => {
        alert("Doctor added Successfully");
        naviget("/DocView");
      })
      .catch((err) => console.log(err));
  } else {
    alert("Please Enter Valid Inputs!!!");
  }
};

const validateValues = (inputData) => {
  if (inputData.dname.length === 0) {
    alert("Doctor Name should not be empty !!!");
    return false;
  } else if (inputData.dexperience.length === 0) {
    alert("Please enter Years of Experience!!!");
    return false;
  }
};

```

```

    }
    else if (inputData.dspecial.length === 0) {
        alert("Please enter Specialization !!!");
        return false;
    }
    else {
        return true;
    }
};

return (
    <div
        id="viewall"
        className="d-flex w-100 vh-100 justify-content-center align-items-center "
    >
    <div className="w-50 border bg-light p-5">
        <form onSubmit={handleSubmit}>
            <a href='/DocView' className='btn btn-info'>Go Back</a>
            <h1 className='display-4'>Add Doctor</h1><br />
            <div>
                <label htmlFor="dname">Doctor Name</label>
                <input
                    type="text"
                    name="dname"
                    className="form-control"
                    onChange={(e) =>
                        setInputData({ ...inputData, dname: e.target.value })
                    }
                />
            </div>
            <div>
                <label htmlFor="experience">Years of Experience</label>
                <input
                    type="text"
                    name="dexperience"
                    className="form-control"
                    onChange={(e) =>
                        setInputData({ ...inputData, dexperience: e.target.value })
                    }
                />
            </div>
            <div>
                <label htmlFor="special">Specialization</label>
                <input
                    type="text"
                    name="dspecial"
                    className="form-control"
                    onChange={(e) =>
                        setInputData({ ...inputData, dspecial: e.target.value })
                    }
                />
            </div>
            <div>
                <label htmlFor="hid">Hospital Branch ID</label> &ampnbsp
                <select className="form-control" name="hid" required title="Select Hospital ID" onChange={handleidchange}>
                    <option name="pid" selected="selected">Select Hospital Branch ID</option>
                    {idlist.map((d, i) => (
                        <option value={d.hid}>
                            {d.hid}
                        </option>
                    )))
                </select>
            </div>
            <br />

            <button className="btn btn-info ">Submit</button>
        </form>
    </div>
    </div>
)
}

```

```
export default Docadd
```

### DocDetails.jsx:

```
import React, { useEffect, useState } from 'react'
import { useNavigate, useParams } from 'react-router-dom';
import axios from 'axios';
import Navbar from '../UI/Navbar';

function DocDetails() {
  const {id} = useParams();
  const [inputData, setInputData] = useState({
    did: "-1",
    dname: "",
    dexperience:"",
    dspecial: "",
    hospital:
    {
      hid: "",
      hname: "",
      hcity: "",
      hrating: ""
    }
  });
  const naviget = useNavigate();

  useEffect(() => {
    axios
      .get("http://localhost:1234/doctor/" + id)
      .then((response) => setInputData(response.data))
      .catch((err) => {
        console.log(err)
      });
  }, []);

  return (
    <>
    <br />
    <Navbar />
    <div >
      <div className='container'>
        <div className='d-flex align-items-center justify-content-center mt-2'>
          <h1 className='display-1 text-center'>Doctor Full Detail</h1>
        </div>
        <div className='d-flex align-items-center justify-content-center mt-2'>
          <table className='table table-dark table-bordered table-hover table-striped w-100'>
            <thead>
              <tr align="center">
                <th>Doctor ID</th>
                <th>Doctor Name</th>
                <th>Years of Experience</th>
                <th>Specialization</th>
                <th>Hospital Branch ID</th>
                <th>Hospital Name</th>
                <th>City</th>
                <th>Hospital Rating</th>
              </tr>
            </thead>
            <tbody>
              <tr align="center">
                <td>{inputData.did}</td>
                <td>{inputData.dname}</td>
                <td>{inputData.dexperience}</td>
                <td>{inputData.dspecial}</td>
                <td>{inputData.hospital.hid}</td>
                <td>{inputData.hospital.hname}</td>
              </tr>
            </tbody>
          </table>
        </div>
      </div>
    </div>
  );
}
```

```

        <td>{inputData.hospital.hcity}</td>
        <td>{inputData.hospital.hrating}</td>
    </tr>
</tbody>
</table>
</div>
</div>
</div>
</div>
</>
)
}

export default DocDetails

```

### DocUpdate.jsx:

```

import React, { useEffect, useState } from 'react';
import axios from 'axios';
import { useNavigate, useParams } from 'react-router-dom';
import './Docview.css';

function Docupdate() {
    const {id} = useParams();
    const [idlist, setIdlist] = useState([]);
    const [inputData, setInputData] = useState({
        did: "-1",
        dname: "",
        dexperience:"",
        dspecial: "",
        hospital:
        {
            hid: "",
            hname: "",
            hcity: "",
            hrating: ""
        }
    });
    const naviget = useNavigate();

    useEffect(() => {
        axios
            .get("http://localhost:1234/doctor/" + id)
            .then((res) => {
                setInputData(res.data);
            })
            .catch((err) => console.log(err));
    }, []);

    useEffect(() => {
        axios
            .get("http://localhost:1234/hospital/all")
            .then((res) => {
                setIdlist(res.data);
            })
            .catch((err) => console.log(err));
    }, []);

    let handleidchange = (e) => {
        const selectedRecordId = e.target.value;
        console.log(selectedRecordId);
        if (selectedRecordId) {
            axios
                .get("http://localhost:1234/hospital/" + selectedRecordId)
                .then((response) => {
                    console.log(response.data);
                    setInputData({ ...inputData, hospital: response.data });
                })
        }
    }
}

```

```

        })
        .catch((err) => {
          console.log(err);
        });
      }
    }

    let handleSubmit = (e) => {
      e.preventDefault();
      let result = validateValues(inputData);
      if (result === true) {
        console.log(inputData)
        axios
          .put("http://localhost:1234/doctor", inputData)
          .then((res) => {
            alert("Doctor Updated Successfully");
            naviget("/DocView");
          })
          .catch((err) => console.log(err));
      } else {
        alert("Please Enter Valid Inputs!!!");
      }
    };
  };

  const validateValues = (inputData) => {
    if (inputData.dname.length === 0) {
      alert("Doctor Name should not be empty !!!");
      return false;
    } else if (inputData.dexperience.length === 0) {
      alert("Please enter Years of Experience!!!");
      return false;
    }
    else if (inputData.dspecial.length === 0) {
      alert("Please enter Specialization !!!");
      return false;
    }
    else {
      return true;
    }
  };
  return (
    <>
    <div
      id="viewall"
      className="d-flex w-100 vh-100 justify-content-center align-items-center "
    >
      <div className="w-50 border bg-light p-5">
        <form onSubmit={handleSubmit}>
          <a href='/DocView' className='btn btn-info'>Go Back</a>
          <h1 className='display-4'>Update Doctor Details</h1><br><br>
          <div>
            <label htmlFor="did">Doctor ID</label>
            <input
              disabled
              readOnly
              type="text"
              name="did"
              className="form-control"
              value={inputData.did}
              onChange={(e) =>
                setInputData({ ...inputData, did: e.target.value })
              }
            />
          </div>
          <div>
            <label htmlFor="ename">Doctor Name</label>
            <input
              type="text"
              name="dname"
              className="form-control"
              value={inputData.dname}
              onChange={(e) =>

```

```

        setInputData({ ...inputData, dname: e.target.value })
    }
    />
</div>
<div>
    <label htmlFor="Department">Years of Experience</label>
    <input
        type="text"
        name="dexperience"
        className="form-control"
        value={inputData.dexperience}
        onChange={(e) =>
            setInputData({ ...inputData, dexperience: e.target.value })
        }
    />
</div>
<div>
    <label htmlFor="hid">Hospital Branch ID</label> &ampnbsp
    <select className="form-control" name="hid" required title="Select Hospital Branch ID"
onChange={handleidchange}>
        <option name="pid">Select Hospital Branch ID</option>
        {idlist.map((d, i) => (
            <option selected value={d.hid}>
                {d.hid}
            </option>
        )))
    </select>
</div>
<br />

    <button className="btn btn-info ">Submit</button>
</form>

        </div>
    </div>
    </>
)
}

export default Docupdate

```

### DocView.css:

```

#viewwall{
    background-image: linear-gradient(143.6deg, rgba(192, 132, 252, 0) 20.79%, rgba(232, 121, 249, 0.26) 40.92%, rgba(204, 171, 238, 0) 70.35%);
}

```

### DocView.jsx:

```

import React from 'react';
import {Link,useNavigate} from 'react-router-dom';
import axios from 'axios';
import { useState,useEffect } from 'react';
import Navbar from '../UI/Navbar';

function DocView() {
    const [columns, setcolumns] = useState([]);
    const [records, setrecords] = useState([]);
    const [find, setFind] = useState([]);
    const navigate = useNavigate();

    useEffect(() => {
        axios
            .get("http://localhost:1234/doctor/all")
            .then((response) => {
                setcolumns(Object.keys(response.data[0]));
                setrecords(response.data);
            })
    }, [])
}

export default DocView

```

```

        })
        .catch((err) => {
            console.log(err);
        });
    }, []);
}

let handleSubmit = (did) => {
    const conf = window.confirm("Do you want to delete?");
    if(conf){
        axios
            .delete("http://localhost:1234/doctor/"+did)
            .then((res) => {
                alert("Record Deleted Successfully!");
                navigate("/DocView");
                window.location.reload();
            })
            .catch((err) => console.log(err));
    }
};

return (
    <>
        <div style={{height:"650px",backgroundImage: "linear-gradient(143.6deg, rgba(192, 132, 252, 0) 20.79%, rgba(232, 121, 249, 0.26) 40.92%, rgba(204, 171, 238, 0) 70.35%")}}>
            <div className='container' >
                <br />
                <a href='/' className='btn btn-info'>Home</a>
                <h1 className='display-1 text-center' style={{color:"Black"}}>Doctor Details</h1>

                {/* <div className='text-start input-group input-group-sm'>
                    <input type="search" onChange={(e)=>setFind(e.target.value)} placeholder="Search Employee
Name"></input>
                </div> */}

                <div className="text-end">
                    <Link to="/Docadd" className="btn btn-info">
                        + Add Doctor
                    </Link>
                </div>

                <br />

                <table className="table table-bordered table-hover table-striped w-100 border bg-white shadow px-5 pb-5
rounded ">
                    <thead>
                        <tr align="center">
                            <th>Doctor ID</th>
                            <th>Doctor Name</th>
                            <th>Years of Experience</th>
                            <th>Specialization</th>
                            <th>Hospital ID</th>
                            <th>Action</th>
                        </tr>
                    </thead>

                    <tbody>
                        {records.map((d,i) => (
                            <tr key={i} align="center">
                                <td>{d.did}</td>
                                <td>{d.dname}</td>
                                <td>{d.dexperience}</td>
                                <td>{d.dspecial}</td>
                                <td>{d.hospital.hid}</td>
                                <td>
                                    <Link to={`/docupdate/${d.did}`} className='btn btn-warning'>Update</Link>
                                    &nbsp;
                                    <button onClick={(e) => handleSubmit(d.did)} className='btn btn-danger'>Delete</button>
                                    &nbsp;
                                    <Link to={`/docfulldetail/${d.did}`} className='btn btn-warning'>Doctor Full
Details</Link>
                                </td>
                            </tr>
                        ))
                    </tbody>
                </table>
            </div>
        </div>
    </>
)

```

```

        ))}
      </tbody>
    </table>
  </div>
</div>
</>
)
}

export default DocView

```

### Hosadd.jsx:

```

import React, { useEffect, useState } from 'react'
import { useNavigate, Link } from 'react-router-dom';
import axios from 'axios';
import './Docview.css';

function Hosadd() {
  const [inputData, setInputData] = useState([]);
  const navigate = useNavigate();

  let handleSubmit = ((e) => {
    axios
      .post("http://localhost:1234/hospital", inputData)
      .then((res) => {
        alert("Hospital Added Successfully!");
        navigate("/Hosview");
      })
      .catch((err) => console.log(err));
  });
  return (
    <>
      <div id="viewall" className="d-flex w-100 vh-100 justify-content-center align-items-center ">
        <div className="w-50 border bg-light p-5">
          <form onSubmit={handleSubmit}>
            <a href='/Hosview' className='btn btn-info'>Go Back</a>
            <h1 className='display-4'>Add Hospital Details</h1> <br />
          <div>
            <label htmlFor="basic">Hospital Name</label>
            <input
              type="text"
              name="hname"
              className="form-control"
              onChange={(e) =>
                setInputData({ ...inputData, hname: e.target.value })
              }
            />
          </div>

          <div>
            <label htmlFor="hcity">City</label>
            <input
              type="text"
              name="hcity"
              className="form-control"
              onChange={(e) =>
                setInputData({ ...inputData, hcity: e.target.value })
              }
            />
          </div>

          <div>
            <label htmlFor="hrating">Rating</label>
            <input
              type="number"
              name="hrating"
              className="form-control"
              onChange={(e) =>
                setInputData({ ...inputData, hrating: e.target.value })
              }
            />
          </div>
        </div>
      </div>
    </>
  );
}

```

```

        }
      />
    </div>

    <br />
    <button className="btn btn-info ">Submit</button>
  </form>
</div>
</div>
</>
)
}

export default Hosadd

```

## HosUpdate.jsx:

```

import React, { useEffect, useState } from 'react'
import axios from 'axios';
import { useNavigate, useParams } from 'react-router-dom';

function HosUpdate() {
  const {id} = useParams();
  const [inputData, setInputData] = useState([]);
  const naviget = useNavigate();

  useEffect(() => {
    axios
      .get("http://localhost:1234/hospital/"+id)
      .then((res) => {
        setInputData(res.data);
      })
      .catch((err) => console.log(err));
  }, []);

  let handleSubmit = (e) => {
    e.preventDefault();
    axios
      .put("http://localhost:1234/hospital", inputData)
      .then((res) => {
        alert("Hospital Updated Successfully");
        naviget("/Hosview");
      })
      .catch((err) => console.log(err));
  };
  return (
    <>
      <div id="viewall" className="d-flex w-100 vh-100 justify-content-center align-items-center">
        <div className="w-50 border bg-light p-5">
          <form onSubmit={handleSubmit}>
            <a href='/Hosview' className='btn btn-info'>Go Back</a>
            <h1 className='display-4'>Update Hospital Details</h1> <br />

            <div>
              <label htmlFor="pid">Hospital ID</label>
              <input disabled
                type="number"
                name="hid"
                className="form-control"
                value={inputData.hid}
                onChange={(e) =>
                  setInputData({ ...inputData, hid: e.target.value })
                }
              />
            </div>

            <div>
              <label htmlFor="hname">Hospital Name</label>
              <input

```

```

        type="text"
        name="hname"
        className="form-control"
        value={inputData.hname}
        onChange={(e) =>
          setInputData({ ...inputData, hname: e.target.value })
        }
      />
    </div>

    <div>
      <label htmlFor="hcity">City</label>
      <input
        type="text"
        name="hcity"
        className="form-control"
        value={inputData.hcity}
        onChange={(e) =>
          setInputData({ ...inputData, hcity: e.target.value })
        }
      />
    </div>

    <div>
      <label htmlFor="hrating">Rating</label>
      <input
        type="number"
        name="hrating"
        className="form-control"
        value={inputData.hrating}
        onChange={(e) =>
          setInputData({ ...inputData, hrating: e.target.value })
        }
      />
    </div>

    <br />
    <button className="btn btn-info ">Submit</button>
  </form>
</div>
</div>
</>
)
}

export default HosUpdate

```

### HosView.css:

```

#hosview{
  background-image: linear-gradient(143.6deg, rgba(192, 132, 252, 0) 20.79%, rgba(232, 121, 249, 0.26) 40.92%, rgba(204, 171, 238, 0) 70.35%);
}

```

### Hosview.jsx:

```

import React from 'react';
import {useNavigate, Link} from 'react-router-dom';
import { useState, useEffect } from 'react';
import axios from 'axios';
import Navbar from '../UI/Navbar';

function Hosview() {
  const [columns, setcolumns] = useState([]);
  const [records, setrecords] = useState([]);
  const navigate = useNavigate();
}

```

```

useEffect(() => {
  axios
    .get("http://localhost:1234/hospital/all")
    .then((response) => {
      setcolumns(Object.keys(response.data[0]));
      setrecords(response.data);
    })
    .catch((err) => {
      console.log(err);
    });
}, []);

let handleSubmit = (hid) => {
  const conf = window.confirm("Do you want to delete?");
  if(conf){
    axios
      .delete("http://localhost:1234/hospital/"+hid)
      .then((res) => {
        alert("Record Deleted Successfully!");
        navigate("/Hosview");
        window.location.reload();
      })
      .catch((err) => console.log(err));
  }
};

return (
  <>
  <div style={{height:"650px",backgroundImage: "linear-gradient(143.6deg, rgba(192, 132, 252, 0) 20.79%, rgba(232, 121, 249, 0.26) 40.92%, rgba(204, 171, 238, 0) 70.35%")}}>
    <div className='container'>
      <br />
      <a href='/' className='btn btn-info'>Home</a>
      <h1 className='display-1 text-center' style={{color:"black"}}>Hospital Details</h1>

      <div className="text-end">
        <Link to="/Hosadd" className="btn btn-info">
          + Add Hospital
        </Link>
      </div>

      <br />

      <table className="table table-bordered table-hover table-striped w-100 border bg-white shadow px-5 pb-5 rounded ">
        <thead>
          <tr align="center">
            {/* {columns.map((c,i) => (
              <th key={i}> {c} </th>
            ))} */}
            <th>Hospital ID</th>
            <th>Hospital Name</th>
            <th>Hospital City</th>
            <th>Hospital Rating</th>
            <th>Action</th>
          </tr>
        </thead>

        <tbody>
          {records.map((d,i) => (
            <tr key={i} align="center">
              <td>{d.hid}</td>
              <td>{d.hname}</td>
              <td>{d.hcity}</td>
              <td>{d.hrating}</td>
              <td>
                <Link to={`/Hosupdate/${d.hid}`} className='btn btn-warning'>Update</Link>
                &ampnbsp
                <button onClick={(e) => handleSubmit(d.hid)} className='btn btn-danger'>Delete</button>
                &ampnbsp
              </td>
            </tr>
          ))}
        </tbody>
      </table>
    </div>
  </div>
)

```

```

        </td>
    </tr>
  )}
</tbody>
</table>
</div>
</div>
</>
)
}

export default Hosview

```

## UI Folder:

### Contact.jsx:

```

export default () => {

  const contactMethods = [
    {
      icon:
        <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" strokeWidth={1.5}
stroke="currentColor" className="w-6 h-6">
          <path strokeLinecap="round" strokeLinejoin="round" d="M15 10.5a3 3 0 11-6 0 3 3 0 016 0z" />
          <path strokeLinecap="round" strokeLinejoin="round" d="M19.5 10.5c0 7.142-7.5 11.25-7.5 11.25S4.5
17.642 4.5 10.5a7.5 7.5 0 1115 0z" />
        </svg>
      ,
      contact: "Mountain View, California, United State.",
      title: "Our office"
    },
    {
      icon:
        <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" strokeWidth={1.5}
stroke="currentColor" className="w-6 h-6">
          <path strokeLinecap="round" strokeLinejoin="round" d="M2.25 6.75c0 8.284 6.716 15 15 15h2.25a2.25
2.25 0 002.25-2.25v-1.372c0-.516-.351-.966-.852-1.091l-4.423-1.106c-.44-.11-.902.055-1.173.417l-.97
1.293c-.282.376-.769.542-1.21.38a12.035 12.035 0 01-7.143-7.143c-.162-.441.004-.928.38-
1.211.293-.97c.363-.271.527-.734.417-1.173L6.963 3.102a1.125 1.125 0 00-1.091-.852H4.5A2.25 2.25 0 002.25 4.5v2.25z
/>
        </svg>
      ,
      contact: "+1 (555) 000-000",
      title: "Phone"
    },
    {
      icon:
        <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" strokeWidth={1.5}
stroke="currentColor" className="w-6 h-6">
          <path strokeLinecap="round" strokeLinejoin="round" d="M21.75 6.75v10.5a2.25 2.25 0 01-2.25 2.25h-15a2.25 2.25 0 01-2.25-2.25V6.75m19.5 0A2.25 2.25 0 0019.5 4.5h-15a2.25 2.25 0 00-2.25 2.25m19.5 0v.243a2.25 2.25 0
01-1.07 1.916l-7.5 4.615a2.25 2.25 0 01-2.36 0L3.32 8.91a2.25 2.25 0 01-1.07-1.916V6.75" />
        </svg>
      ,
      contact: "Support@example.com",
      title: "Email"
    },
  ]

  return (
    <main className="py-14">
      <div className="max-w-screen-xl mx-auto px-4 text-gray-600 md:px-8">
        <div className="max-w-xl space-y-3">
          <h3 className="text-indigo-600 font-semibold">
            Contact
          </h3>
          <p className="text-gray-800 text-3xl font-semibold sm:text-4xl">
            Let us know how we can help
          </p>

```

```

        <p>
          We're here to help and answer any question you might have, We look forward to hearing from
        you .
        </p>
      </div>
      <div>
        <ul className="mt-12 flex flex-wrap gap-x-12 gap-y-6 items-center lg:gap-x-24">
          {
            contactMethods.map((item, idx) => (
              <li key={idx}>
                <h4 className="text-gray-800 text-lg font-medium">{item.title}</h4>
                <div className="mt-3 flex items-center gap-x-3">
                  <div className="flex-none text-gray-400">
                    {item.icon}
                  </div>
                  <p>{item.contact}</p>
                </div>
              </li>
            ))
          }
        </ul>
      </div>
    </div>
  </main>
)
}

```

## CTA.jsx:

```

export default () => {

  const contactMethods = [
    {
      icon:
        <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" strokeWidth={1.5}
stroke="currentColor" className="w-6 h-6">
          <path strokeLinecap="round" strokeLinejoin="round" d="M15 10.5a3 3 0 11-6 0 3 3 0 016 0z" />
          <path strokeLinecap="round" strokeLinejoin="round" d="M19.5 10.5c0 7.142-7.5 11.25-7.5 11.25S4.5
17.642 4.5 10.5a7.5 7.5 0 1115 0z" />
        </svg>
      ,
      contact: "Mountain View, California, United State.",
      title: "Our office"
    },
    {
      icon:
        <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" strokeWidth={1.5}
stroke="currentColor" className="w-6 h-6">
          <path strokeLinecap="round" strokeLinejoin="round" d="M2.25 6.75c0 8.284 6.716 15 15 15h2.25a2.25
2.25 0 002.25-2.25v-1.372c0-.516-.351-.966-.852-1.091l-4.423-1.106c-.44-.11-.902.055-1.173.417l-.97
1.293c-.282.376-.769.542-1.21.38a12.035 12.035 0 01-7.143-7.143c-.162-.441.004-.928.38-
1.211.293-.97c.363-.271.527-.734.417-1.173L6.963 3.102a1.125 1.125 0 00-1.091-.852H4.5A2.25 2.25 0 002.25 4.5v2.25z"
/>
        </svg>
      ,
      contact: "+1 (555) 000-000",
      title: "Phone"
    },
    {
      icon:
        <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" strokeWidth={1.5}
stroke="currentColor" className="w-6 h-6">
          <path strokeLinecap="round" strokeLinejoin="round" d="M21.75 6.75v10.5a2.25 2.25 0 01-2.25 2.25h-15a2.25 2.25 0 01-2.25-2.25V6.75m19.5 0A2.25 2.25 0 0019.5 4.5h-15a2.25 2.25 0 00-2.25 2.25m19.5 0v.243a2.25 2.25 0 01-1.07 1.916l-7.5 4.615a2.25 2.25 0 01-2.36 0L3.32 8.91a2.25 2.25 0 01-1.07-1.916V6.75" />
        </svg>
      ,
      contact: "Support@example.com",
      title: "Email"
    },
  ],
}

```

```

return (
  <main className="py-14">
    <div className="max-w-screen-xl mx-auto px-4 text-gray-600 md:px-8">
      <div className="max-w-xl space-y-3">
        <h3 className="text-indigo-600 font-semibold">
          Contact
        </h3>
        <p className="text-gray-800 text-3xl font-semibold sm:text-4xl">
          Let us know how we can help
        </p>
        <p>
          We're here to help and answer any question you might have, We look forward to hearing from
          you .
        </p>
      </div>
      <div>
        <ul className="mt-12 flex flex-wrap gap-x-12 gap-y-6 items-center lg:gap-x-24">
          {
            contactMethods.map((item, idx) => (
              <li key={idx}>
                <h4 className="text-gray-800 text-lg font-medium">{item.title}</h4>
                <div className="mt-3 flex items-center gap-x-3">
                  <div className="flex-none text-gray-400">
                    {item.icon}
                  </div>
                  <p>{item.contact}</p>
                </div>
              </li>
            ))
          }
        </ul>
      </div>
    </div>
  </main>
)
}

```

## Features.jsx:

```

export default () => {

  const features = [
    {
      icon:
        <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" strokeWidth={1.5}
        stroke="currentColor" className="w-6 h-6">
          <path strokeLinecap="round" strokeLinejoin="round" d="M3.75 13.5l10.5-11.25L12 10.5h8.25L9.75
        21.75 12 13.5H3.75z" />
        </svg>,
      title: "Faster Refresh",
      desc: "Intake more Electrolytic foods for Refreshment 😊"
    },
    {
      icon:
        <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" strokeWidth={1.5}
        stroke="currentColor" className="w-6 h-6">
          <path strokeLinecap="round" strokeLinejoin="round" d="M3 13.125C3 12.504 3.504 12 4.125
        12h2.25c.621 0 1.125.504 1.125 1.125v6.75C7.5 20.496 6.996 21 6.375 21h-2.25A1.125 1.125 0 013 19.875v-6.75zM9.75
        8.625c0-.621.504-1.125 1.125-1.125h2.25c.621 0 1.125.504 1.125 1.125v11.25c0 .621-.504 1.125-1.125 1.125h-2.25a1.125
        1.125 0 01-1.125-1.125V8.625zM16.5 4.125c0-.621.504-1.125 1.125-1.125h2.25C20.496 3 21 3.504 21
        4.125v15.75c0 .621-.504 1.125-1.125 1.125h-2.25a1.125 1.125 0 01-1.125-1.125V4.125z" />
        </svg>,
      title: "Steps",
      desc: "Walk 10k Steps per day for better health 🏃"
    },
    {
      icon:
        <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" strokeWidth={1.5}
        stroke="currentColor" className="w-6 h-6">

```

```

        <path strokeLinecap="round" strokeLinejoin="round" d="M16.5 10.5V6.75a4.5 4.5 0 10-9 0v3.75m-.75
11.25h10.5a2.25 2.25 0 002.25-2.25v-6.75a2.25 2.25 0 00-2.25-2.25H6.75a2.25 2.25 0 00-2.25 2.25v6.75a2.25 2.25 0
002.25 2.25z" />
        </svg>,
        title: "Heart security",
        desc: "Avoid eating more Oil Foods! It can damage your Heart ❤️"
    },
    {
        icon:
            <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" strokeWidth={1.5}
stroke="currentColor" className="w-6 h-6">
                <path strokeLinecap="round" strokeLinejoin="round" d="M6.429 9.75L2.25 1214.179 2.25m0-4.515.571 3
5.571-3m-11.142 0L2.25 7.5 12 2.25l9.75 5.25-4.179 2.25m0 0L21.75 121-4.179 2.25m0 014.179 2.25L12 21.75 2.25
16.514.179-2.25m11.142 0l-5.571 3-5.571-3" />
            </svg>,
        title: "Build your Muscles",
        desc: "Do HITT exercises daily to get a Muscular Body 💪"
    },
    {
        icon:
            <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" strokeWidth={1.5}
stroke="currentColor" className="w-6 h-6">
                <path strokeLinecap="round" strokeLinejoin="round" d="M9 12.75L11.25 15 15 15 9.75m-3-7.036A11.959
11.959 0 013.598 6 11.99 11.99 0 003 9.749c0 5.592 3.824 10.29 9 11.623 5.176-1.332 9-6.03 9-11.622 0-1.31-.21-
2.571-.598-3.751h-.152c-3.196 0-6.1-1.248-8.25-3.285z" />
            </svg>,
        title: "Use Prescribed Medicines only",
        desc: "Eat only prescribed medicines 🧠"
    },
    {
        icon:
            <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" strokeWidth={1.5}
stroke="currentColor" className="w-6 h-6">
                <path strokeLinecap="round" strokeLinejoin="round" d="M9.813 15.904L9 18.75l-.813-2.846a4.5 4.5 0
00-3.09-3.09L2.25 1212.846-.813a4.5 4.5 0 003.09-3.09L9 5.251.813 2.846a4.5 4.5 0 003.09 3.09L15.75 121-2.846.813a4.5
4.5 0 00-3.09 3.09zM18.259 8.715L18 9.75l-.259-1.035a3.375 3.375 0 00-2.455-2.456L14.25 61.036-.259a3.375 3.375 0
002.455-2.456L18 2.25l.259 1.035a3.375 3.375 0 002.456 2.456L21.75 61-1.035.259a3.375 3.375 0 00-2.456 2.456zM16.894
20.567L16.5 21.75l-.394-1.183a2.25 2.25 0 00-1.423-1.423L13.5 18.75l1.183-.394a2.25 2.25 0 001.423-1.423L13.5
20.567 1.183.394 1.183a2.25 2.25 0 001.423 1.423L1.183.394-1.183.394a2.25 2.25 0 00-1.423 1.423z" />
            </svg>,
        title: "Be Flexible",
        desc: "Do Exercise and be more Flexible ⚡"
    },
]
}

return (
<section className="py-14">
    <div className="max-w-screen-xl mx-auto px-4 text-gray-600 md:px-8">
        <div className="relative max-w-2xl mx-auto sm:text-center">
            <div className="relative z-10">
                <h3 className="text-gray-800 text-3xl font-semibold sm:text-4xl">
                    Let's help Power your Inner Beast
                </h3>
                <p className="mt-3">
                    Take more Vitamins and Proteins daily!
                </p>
            </div>
            <div className="absolute inset-0 max-w-xs mx-auto h-44 blur-[118px]" style={{ background: "linear-gradient(152.92deg, rgba(192, 132, 252, 0.2) 4.54%, rgba(232, 121, 249, 0.26) 34.2%, rgba(192, 132, 252, 0.1) 77.55%)" }}></div>
        </div>
        <div className="relative mt-12">
            <ul className="grid gap-8 sm:grid-cols-2 lg:grid-cols-3">
                {
                    features.map((item, idx) => (
                        <li key={idx} className="bg-white space-y-3 p-4 border rounded-lg">
                            <div className="text-indigo-600 pb-3">
                                {item.icon}
                            </div>
                            <h4 className="text-lg text-gray-800 font-semibold">
                                {item.title}
                            </h4>
                            <div className="text-gray-600 sm:grid grid-cols-2 gap-2">
                                {item.list.map((listItem, index) => (
                                    <div key={index} className="flex items-center gap-2">
                                        {listItem.icon}
                                        {listItem.title}
                                    </div>
                                ))}
                            </div>
                        </li>
                    ))
                }
            </ul>
        </div>
    </div>
</section>

```

```

                <p>
                    {item.desc}
                </p>
            </li>
        ))
    }
</ul>
</div>
</div>
</section>
)
}

```

## Footer.jsx:

```

import logo from './images/logo.png';
export default () => {

  const footerNavs = [
    {
      href: 'javascript:void()',
      name: 'Terms'
    },
    {
      href: 'javascript:void()',
      name: 'License'
    },
    {
      href: 'javascript:void()',
      name: 'Privacy'
    },
    {
      href: 'javascript:void()',
      name: 'About us'
    }
  ]
  return (
    <footer className="pt-10">
      <div className="max-w-screen-xl mx-auto px-4 text-gray-600 md:px-8">
        <div className="space-y-6 sm:max-w-md sm:mx-auto sm:text-center">
          <img src={logo} className="w-32 sm:mx-auto" />
          <p>
            Nulla auctor metus vitae lectus iaculis, vel euismod massa efficitur.
          </p>
          <div className="items-center gap-x-3 space-y-3 sm:flex sm:justify-center sm:space-y-0">
            <a href="javascript:void(0)" className="block py-2 px-4 text-center text-white font-medium bg-indigo-600 duration-150 hover:bg-indigo-500 active:bg-indigo-700 rounded-lg shadow-lg hover:shadow-none">
              Get Appointment
            </a>
            <a href="javascript:void(0)" className="flex items-center justify-center gap-x-2 py-2 px-4 text-gray-700 hover:text-gray-500 font-medium duration-150 active:bg-gray-100 border rounded-lg md:inline-flex">
              Get Consultant
              <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 20 20" fill="currentColor" className="w-5 h-5">
                <path fillRule="evenodd" d="M2 10a.75.75 0 0 1-.75-.75h12.59l-2.1-1.95a.75.75 0 0 1 1.113.5 3.25a.75.75 0 0 0 1.11-3.5 3.25a.75.75 0 0 1 11-1.02-1.112.1-1.95H2.75A.75.75 0 0 0 10z" clipRule="evenodd" />
              </svg>
            </a>
          </div>
        </div>
        <div className="mt-10 py-10 border-t items-center justify-between sm:flex">
          <p>© 2024 Care+ Inc. All rights reserved.</p>
          <ul className="flex flex-wrap items-center gap-4 mt-6 sm:text-sm sm:mt-0">
            {
              footerNavs.map((item, idx) => (
                <li className="text-gray-800 hover:text-gray-500 duration-150">
                  <a key={idx} href={item.href}>
                    {item.name}
                  </a>
                </li>
              ))
            }
          </ul>
        </div>
      </div>
    </footer>
  )
}

```

```

        </ul>
    </div>
</div>
</footer>
)
}

```

## Hero.jsx:

```

import { useEffect, useState } from 'react'
import {Link} from 'react-router-dom';
import logo from './images/logo.png';
import doc from './images/doc1.png';

export default () => {

  const [state, setState] = useState(false)

  // Replace javascript:void(0) paths with your paths
  const navigation = [
    { title: "Doctor", path: "/Docview" },
    { title: "Our Branches", path: "/Hosview" },
    { title: "Customers", path: "javascript:void(0)" },
    { title: "Pricing", path: "javascript:void(0)" }
  ]

  useEffect(() => {
    document.onclick = (e) => {
      const target = e.target;
      if (!target.closest(".menu-btn")) setState(false);
    };
  }, [])

  const Brand = () => (
    <div className="flex items-center justify-between py-3 md:block">
      <a href="/">
        <img
          src={logo}
          width={60}
          height={50}
          alt="Float UI logo"
        />
      </a>
      <div className="md:hidden">
        <button className="menu-btn text-gray-500 hover:text-gray-800"
          onClick={() => setState(!state)}
        >
          {
            state ? (
              <svg xmlns="http://www.w3.org/2000/svg" className="h-6 w-6" viewBox="0 0 20 20"
              fill="currentColor">
                <path fillRule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0
                11.414 1.414L11.414 10L4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293
                5.707a1 1 0 010-1.414z" clipRule="evenodd" />
              </svg>
            ) : (
              <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" strokeWidth={1.5}
              stroke="currentColor" className="w-6 h-6">
                <path strokeLinecap="round" strokeLinejoin="round" d="M3.75 6.75h16.5M3.75 12h16.5-
                16.5 5.25h16.5" />
              </svg>
            )
          }
        </button>
      </div>
    </div>
  )
}

return (
  <div className='relative'>

```

```

<div className='absolute inset-0 blur-xl h-[580px]' style={{ background: "linear-gradient(143.6deg, rgba(192, 132, 252, 0) 20.79%, rgba(232, 121, 249, 0.26) 40.92%, rgba(204, 171, 238, 0) 70.35%)" }}></div>
<div className='relative'>
  <header>
    <div className={`md:hidden ${state ? "mx-2 pb-5" : "hidden"} }>
      <Brand />
    </div>
    <nav className={`pb-15 md:text-sm ${state ? "absolute top-0 inset-x-0 bg-white shadow-lg rounded-xl border mx-2 mt-2 md:shadow-none md:border-none md:mx-0 md:mt-0 md:relative md:bg-transparent" : ""}`}>
      <div className="gap-x-14 items-center max-w-screen-xl mx-auto px-4 md:flex md:px-5">
        <Brand />
        <div className={`flex-1 items-center mt-8 md:mt-0 md:flex ${state ? 'block' : 'hidden'} `}>
          <ul className="flex-1 justify-center items-center space-y-6 md:flex md:space-x-6 md:space-y-0">
            {
              navigation.map((item, idx) => {
                return (
                  <li key={idx} className="text-gray-700 hover:text-gray-900">
                    <a href={item.path} className="block">
                      {item.title}
                    </a>
                  </li>
                )
              })
            }
          </ul>
          <div className="items-center justify-end mt-6 space-y-6 md:flex md:mt-0">
            <Link to="/Signin">
              <a href="javascript:void(0)" className="flex items-center justify-center gap-x-1 py-2 px-4 text-white font-medium bg-gray-800 hover:bg-gray-700 active:bg-gray-900 rounded-full md:inline-flex">
                Sign in
                <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 20 20" fill="currentColor" className="w-5 h-5">
                  <path fillRule="evenodd" d="M7.21 14.77a.75.75 0 0 1-.02-1.06L11.168 10 7.23 6.29a.75.75 0 1 0 -.04-1.08l4.5 4.25a.75.75 0 0 1 1.08l-4.5 4.25a.75.75 0 0 1 -1.06-.02z" clipRule="evenodd" />
                </svg>
              </a>
            </Link>
          </div>
        </div>
      </div>
    </nav>
  </header>
  <section>
    <div className="max-w-screen-xl mx-auto px-6 py-5 gap-24 text-gray-600 overflow-hidden md:px-9 md:flex">
      <div className='flex-none space-y-5 max-w-xl'>
        <a href="javascript:void(0)" className='inline-flex gap-x-6 items-center rounded-full p-1 pr-6 border text-sm font-medium duration-150 hover:bg-white'>
          <span className='inline-block rounded-full px-3 py-1 bg-indigo-600 text-white'>
            News
          </span>
        <p className='flex items-center'>
          Read the latest Health news from here
          <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 20 20" fill="currentColor" className="w-5 h-5">
            <path fillRule="evenodd" d="M7.21 14.77a.75.75 0 0 1-.02-1.06L11.168 10 7.23 6.29a.75.75 0 1 0 -.04-1.08l4.5 4.25a.75.75 0 0 1 1.08l-4.5 4.25a.75.75 0 0 1 -1.06-.02z" clipRule="evenodd" />
          </svg>
        </p>
      </a>
      <h1 className="text-4xl text-gray-800 font-extrabold sm:text-5xl">
        Care+ Hospital
      </h1>
      <p>
        🌟 Trusted by Millions of Peoples ❤️
      </p>
      <div className='flex items-center gap-x-3 sm:text-sm'>
        <Link to="/DocView">
          <a href="javascript:void(0)" className="flex items-center justify-center gap-x-1 py-2 px-4 text-white font-medium bg-gray-800 duration-150 hover:bg-gray-700 active:bg-gray-900 rounded-full md:inline-flex">

```

```

        Doctor Details
        <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 20 20" fill="currentColor"
className="w-5 h-5">
            <path fillRule="evenodd" d="M7.21 14.77a.75.75 0 01.02-1.06L11.168 10 7.23
6.29a.75.75 0 11.04-1.08l4.5 4.25a.75.75 0 010 1.08l-4.5 4.25a.75.75 0 01-1.06-.02z" clipRule="evenodd" />
        </svg>
    </a>
    </Link>
    <Link to="/Hosview">
        <a href="javascript:void(0)" className="flex items-center justify-center gap-x-1 py-2
px-4 text-gray-700 hover:text-gray-900 font-medium duration-150 md:inline-flex">
            Hospital Details
            <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 20 20" fill="currentColor"
className="w-5 h-5">
                <path fillRule="evenodd" d="M7.21 14.77a.75.75 0 01.02-1.06L11.168 10 7.23
6.29a.75.75 0 11.04-1.08l4.5 4.25a.75.75 0 010 1.08l-4.5 4.25a.75.75 0 01-1.06-.02z" clipRule="evenodd" />
            </svg>
        </a>
        </Link>
    </div>
</div>&nbsp; &nbsp; &nbsp; &nbsp;
<div className='flex-1 hidden md:block'>
    {/* Replace with your image */}
    <img src={doc} className=" w-full max-w-sm" />
    {/* https://www.digiclove.com/wp-content/uploads/2023/03/hrms.png */}
</div>
</div>
</section>
</div>
</div>
)
}
}

```

## Teams.jsx:

```

export default () => {

    const team = [
        {
            avatar: "https://d2az3zd39o5d63.cloudfront.net/linkedin-profile-picture-squinch.jpg",
            name: "Martiana dialan",
            title: "Senior Ortho",
            desc: "Ortho Doctor",
            linkedin: "javascript:void(0)",
            twitter: "javascript:void(0)",
        },
        {
            avatar: "https://api.uifaces.co/our-content/donated/xZ4wg2Xj.jpg",
            name: "Micheal colorand",
            title: "Junior Compounder",
            desc: "Compounder - Intern",
            linkedin: "javascript:void(0)",
            twitter: "javascript:void(0)",
        },
        {
            avatar: "https://randomuser.me/api/portraits/women/79.jpg",
            name: "Brown Luis",
            title: "General",
            desc: "General Doctor",
            linkedin: "javascript:void(0)",
            twitter: "javascript:void(0)",
        },
        {
            avatar: "https://randomuser.me/api/portraits/women/63.jpg",
            name: "Lysa sandiago",
            title: "Head of Departments",
            desc: "Chief of all Departments",
            linkedin: "javascript:void(0)",
            twitter: "javascript:void(0)",
        },
        {
            avatar: "https://randomuser.me/api/portraits/men/86.jpg",

```

```

        name: "Daniel martin",
        title: "Medicine Consultant",
        desc: "Chief Medical Rep",
        linkedin: "javascript:void(0)",
        twitter: "javascript:void(0)",
    },
{
    avatar: "https://randomuser.me/api/portraits/men/46.jpg",
    name: "Vicky tanson",
    title: "Dean",
    desc: "Dean of Care+ Hospital",
    linkedin: "javascript:void(0)",
    twitter: "javascript:void(0)",
},
]

return (
    <section className="py-14">
        <div className="max-w-screen-xl mx-auto px-4 text-center md:px-8">
            <div className="max-w-xl mx-auto">
                <h3 className="text-gray-800 text-3xl font-semibold sm:text-4xl">
                    Meet our team
                </h3>
                <p className="text-gray-600 mt-3">
                    Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been
                    the industry's standard dummy.
                </p>
            </div>
            <div className="mt-12">
                <ul className="grid gap-8 sm:grid-cols-2 md:grid-cols-3">
                    {
                        team.map((item, idx) => (
                            <li key={idx}>
                                <div className="w-24 h-24 mx-auto">
                                    <img
                                        src={item.avatar}
                                        className="w-full h-full rounded-full"
                                        alt=""
                                    />
                                </div>
                                <div className="mt-2">
                                    <h4 className="text-gray-700 font-semibold sm:text-lg">{item.name}</h4>
                                    <p className="text-indigo-600">{item.title}</p>
                                    <p className="text-gray-600 mt-2">{item.desc}</p>
                                    <div className="mt-4 flex justify-center gap-4 text-gray-400">
                                        <a href={item.twitter}>
                                            <svg className="w-5 h-5 duration-150 hover:text-gray-500"
                                                fill="currentColor" viewBox="0 0 48 48"><g clip-path="url(#clip0_17_80)"><path fill="currentColor" d="M15.1 43.5c18.11
0 28.017-15.006 28.017-28.016 0-.422-.01-.853-.029-1.275A19.998 19.998 0 0 0 4 9.11c-1.795.798-3.7 1.32-5.652 1.546a9.9
9.9 0 0 0 4.33-5.445 19.794 19.794 0 0 1-6.251 2.39 9.86 9.86 0 0 0-16.788 8.979A27.97 27.97 0 0 1 13.346 6.299 9.859 9.859 0
006.393 19.44a9.86 9.86 0 0 1-4.462-1.228v.122a9.844 9.844 0 0 0 7.901 9.656 9.788 9.788 0 0 1-4.442.169 9.867 9.867 0
009.195 6.843A19.75 19.75 0 0 1 39.078 27.937 27.937 0 0 0 15.1 43.5z" /></g></defs><clipPath id="clip0_17_80"><path
fill="currentColor" d="M0 0h48v48H0z" /></clipPath></defs></svg>
                                        </a>
                                        <a href={item.linkedin}>
                                            <svg className="w-5 h-5 duration-150 hover:text-gray-500" fill="none"
                                                viewBox="0 0 48 48"><g clip-path="url(#clip0_17_68)"><path fill="currentColor" d="M44.447 0h3.544c1.584 0 0 1.547 0
3.46v44.53c0 46.444 1.584 48 3.544 48h40.903c46.407 48 48 46.444 48 44.54v3.46c48 1.546 46.406 0 44.447 0zM14.24
40.903h7.116v17.991h7.125v22.912zM10.678 14.87a4.127 4.127 0 0 1-4.134-4.125 4.127 4.127 0 0 1 4.134-4.125 4.125 4.125 0
010 8.25zm30.225 26.034h-7.115v29.766c0-2.653-.047-6.075-3.704-6.075-3.703 0-4.265 2.896-4.265 5.887v11.325h-7.107v17.991h6.826v3.13h.093c.947-1.8 3.272-3.702 6.731-3.702 7.21 0 8.541 4.744 8.541 10.912v12.572z"
/></g></defs><clipPath id="clip0_17_68"><path fill="currentColor" d="M0 0h48v48H0z" /></clipPath></defs></svg>
                                        </a>
                                    </div>
                                </div>
                            </li>
                        )))
                    }
                </ul>
            </div>
        </div>
    </section>

```

```
)  
}
```

## Components Folder:

### Signin.jsx:

```
import {Link, useNavigate} from 'react-router-dom';

export default function Signin() {
  const navigate = useNavigate();

  let handleClick = (e) => {
    alert("Sign in success!!!");
    navigate("/");
  }
  return (
    <>
    {/* This example requires updating your template:
      ...
      <html class="h-full bg-white">
      <body class="h-full">
      ...
      */}

    <div>
      <Link to="/">
        <button type="button" class="w-full flex items-center justify-center w-1/2 px-5 py-2 text-sm text-gray-700 transition-colors duration-200 bg-white border rounded-lg gap-x-2 sm:w-auto dark:hover:bg-gray-800 dark:bg-gray-900 hover:bg-gray-100 dark:text-gray-200 dark:border-gray-700">
          <svg class="w-5 h-5 rtl:rotate-180" xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" stroke-width="1.5" stroke="currentColor">
            <path stroke-linecap="round" stroke-linejoin="round" d="M6.75 15.75L3 12m0 0l3.75-3.75M3 12h18" />
          </svg>
          <span>Go back</span>
        </button>
      </Link>
    </div>
    <div className="flex min-h-full flex-1 flex-col justify-center px-6 py-12 lg:px-8">
      <div className="sm:mx-auto sm:w-full sm:max-w-sm">
        
        <h2 className="mt-10 text-center text-2xl font-bold leading-9 tracking-tight text-gray-900">
          Sign in to your account
        </h2>
      </div>

      <div className="mt-10 sm:mx-auto sm:w-full sm:max-w-sm">
        <form className="space-y-6" action="#" method="POST">
          <div>
            <label htmlFor="email" className="block text-sm font-medium leading-6 text-gray-900">
              Email address
            </label>
            <div className="mt-2">
              <input
                id="email"
                name="email"
                type="email"
                autoComplete="email"
                required
                className="block w-full rounded-md border-0 py-1.5 text-gray-900 shadow-sm ring-1 ring-inset ring-gray-300 placeholder:text-gray-400 focus:ring-2 focus:ring-inset focus:ring-indigo-600 sm:text-sm sm:leading-6"
              />
            </div>
          </div>

        <div>
```

```

        <div className="flex items-center justify-between">
          <label htmlFor="password" className="block text-sm font-medium leading-6 text-gray-900">
            Password
          </label>
          <div className="text-sm">
            <a href="#" className="font-semibold text-indigo-600 hover:text-indigo-500">
              Forgot password?
            </a>
          </div>
        </div>
        <div className="mt-2">
          <input
            id="password"
            name="password"
            type="password"
            autoComplete="current-password"
            required
            className="block w-full rounded-md border-0 py-1.5 text-gray-900 shadow-sm ring-1 ring-inset ring-gray-300 placeholder:text-gray-400 focus:ring-2 focus:ring-inset focus:ring-indigo-600 sm:text-sm sm:leading-6"
            />
        </div>
      </div>

      <div>
        <button
          type="submit"
          className="flex w-full justify-center rounded-md bg-indigo-600 px-3 py-1.5 text-sm font-semibold leading-6 text-white shadow-sm hover:bg-indigo-500 focus-visible:outline focus-visible:outline-2 focus-visible:outline-offset-2 focus-visible:outline-indigo-600"
          onClick={handleClick}
        >
          Sign in
        </button>
      </div>
    </form>

    <p className="mt-10 text-center text-sm text-gray-500">
      Not a member?{' '}
      <a href="#" className="font-semibold leading-6 text-indigo-600 hover:text-indigo-500">
        Start a 14 day free trial
      </a>
    </p>
  </div>
</>
)
}

```

## Welcome.jsx:

```

import React from 'react'
import Hero from './UI/Hero'
import CTA from './UI/CTA'
import Teams from './UI/Teams'
import Contact from './UI/Contact'
import Footer from './UI/Footer'
import Features from './UI/Features'

function Welcome() {
  return (
    <div>
      <Hero />
      <Features />
      <Teams />
      <CTA />
      <Contact />
      <Footer />
    </div>
  )
}

```

```
export default Welcome
```

## SRC Folder:

### AppRouter.jsx:

```
import React from 'react';
import {BrowserRouter as Router,Route,Routes,Link} from 'react-router-dom';
import Welcome from './components/Welcome';
import Signin from './components/Signin';
import DocView from './components/Pages/DocView';
import Hosview from './components/Pages/Hosview';
import Docadd from './components/Pages/Docadd';
import Docupdate from './components/Pages/Docupdate';
import DocDetails from './components/Pages/DocDetails';
import Hosadd from './components/Pages/Hosadd';
import HosUpdate from './components/Pages/HosUpdate';

function AppRouter() {
  return (
    <div>
      <Router>
        <Routes>
          <Route path='/' element={<Welcome />} />
          <Route path='/Signin' element={<Signin />} />

          <Route path='/DocView' element={<DocView />} />
          <Route path='/Docadd' element={<Docadd />} />
          <Route path='/docupdate/:id' element={<Docupdate />} />
          <Route path='/docfulldetail/:id' element={<DocDetails />} />

          <Route path='/Hosview' element={<Hosview />} />
          <Route path="/Hosadd" element={<Hosadd />} />
          <Route path='/HosUpdate/:id' element={<HosUpdate />} />
        </Routes>
      </Router>
    </div>
  )
}

export default AppRouter
```

### Index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import reportWebVitals from './reportWebVitals';
import AppRouter from './AppRouter';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <AppRouter />
  </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

### Index.css:

```
@tailwind base;
@tailwind components;
```

```

@tailwind utilities;

body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
  'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
  sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
  monospace;
}

```

## Tailwind.config.js:

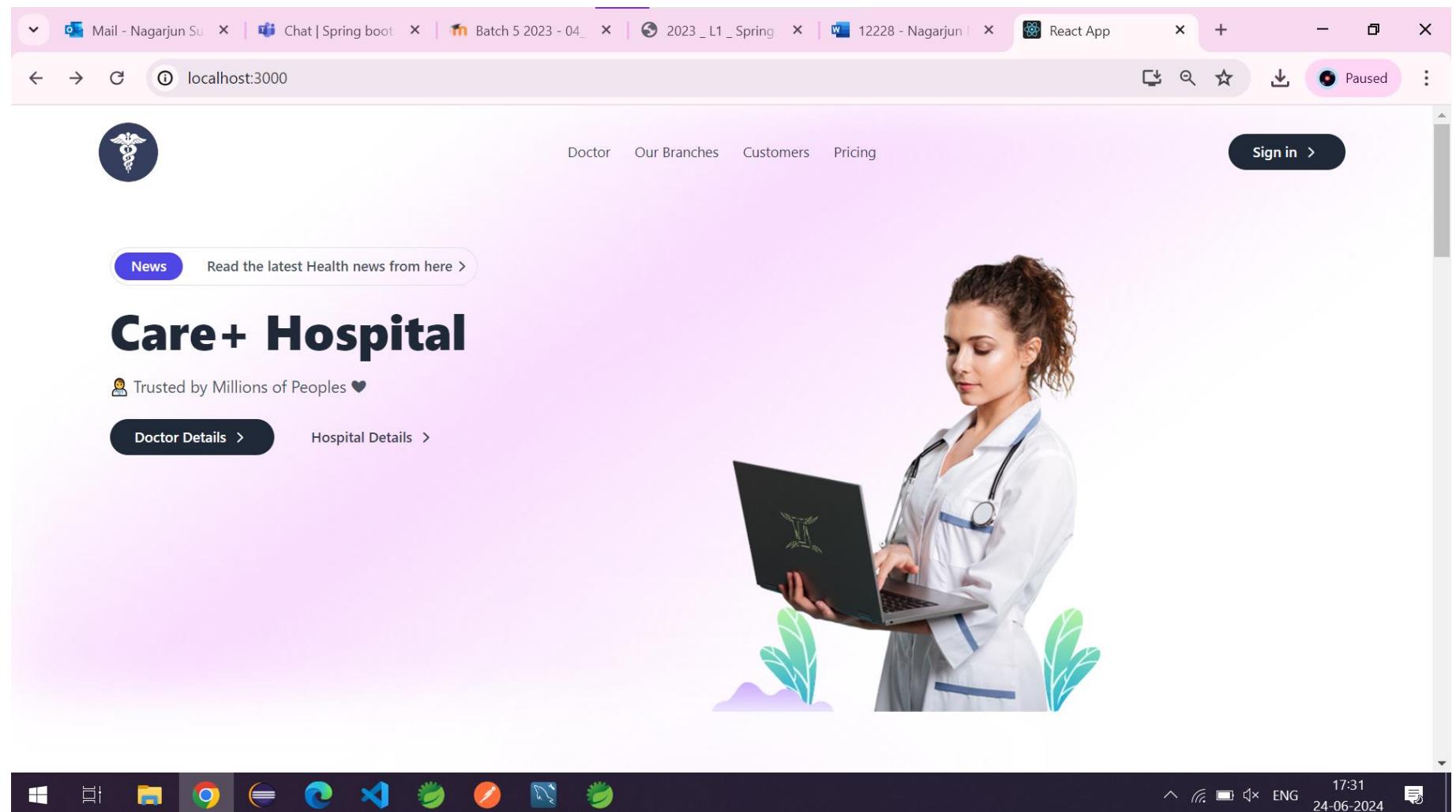
```

/** @type {import('tailwindcss').Config} */
module.exports = {
  content: ["./src/**/*.{html,js,jsx}"],
  theme: {
    extend: {},
  },
  plugins: [],
}

```

## SCREENSHOTS:

### Homepage:



Mail - Nagarjun Su | Chat | Spring boot | Batch 5 2023 - 04\_ | 2023 \_ L1 \_ Spring | 12228 - Nagarjun | React App

localhost:3000

# Let's help Power your Inner Beast

Take more Vitamins and Proteins daily!

 Faster Refresh  
Intake more Electrolytic foods for Refreshment 😊

 Steps  
Walk 10k Steps per day for better health 💪

 Heart security  
Avoid eating more Oil Foods! It can damage your Heart ❤️

 Build your Muscles  
Do HITT exercises daily to get a Muscular Body 💪

 Use Prescribed Medicines only  
Eat only prescribed medicines 💊

 Be Flexible  
Do Exercise and be more Flexible 💪

Windows Mail Chat | Spring boot Batch 5 2023 - 04\_ 2023 \_ L1 \_ Spring 12228 - Nagarjun React App

localhost:3000

# Meet our team

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy.



**Martiana dialan**  
Senior Ortho  
Ortho Doctor  
[Twitter](#) [LinkedIn](#)



**Micheal colorand**  
Junior Compounder  
Compounder - Intern  
[Twitter](#) [LinkedIn](#)



**Brown Luis**  
General  
General Doctor  
[Twitter](#) [LinkedIn](#)



**Lysa sandiago**  
Head of Departments  
Chief of all Departments  
[Twitter](#) [LinkedIn](#)



**Daniel martin**  
Medicine Consultant  
Chief Medical Rep  
[Twitter](#) [LinkedIn](#)



**Vicky tanson**  
Dean  
Dean of Care+ Hospital  
[Twitter](#) [LinkedIn](#)

17:31 24-06-2024

Mail - Nagarjun Su | Chat | Spring boot | Batch 5 2023 - 04\_ | 2023 \_ L1 \_ Spring | 12228 - Nagarjun | React App

localhost:3000

Get care from us

Feel free to Contact Us!

Get started Learn more

Contact

## Let us know how we can help

We're here to help and answer any question you might have, We look forward to hearing from you .

Our office	Phone	Email
Mountain View, California, United State.	+1 (555) 000-000	Support@example.com

17:32 24-06-2024

Mail - Nagarjun Su | Chat | Spring boot | Batch 5 2023 - 04\_ | 2023 \_ L1 \_ Spring | 12228 - Nagarjun | React App

localhost:3000

We're here to help and answer any question you might have, We look forward to hearing from you .

Our office	Phone	Email
Mountain View, California, United State.	+1 (555) 000-000	Support@example.com



Nulla auctor metus vitae lectus iaculis, vel euismod massa efficitur.

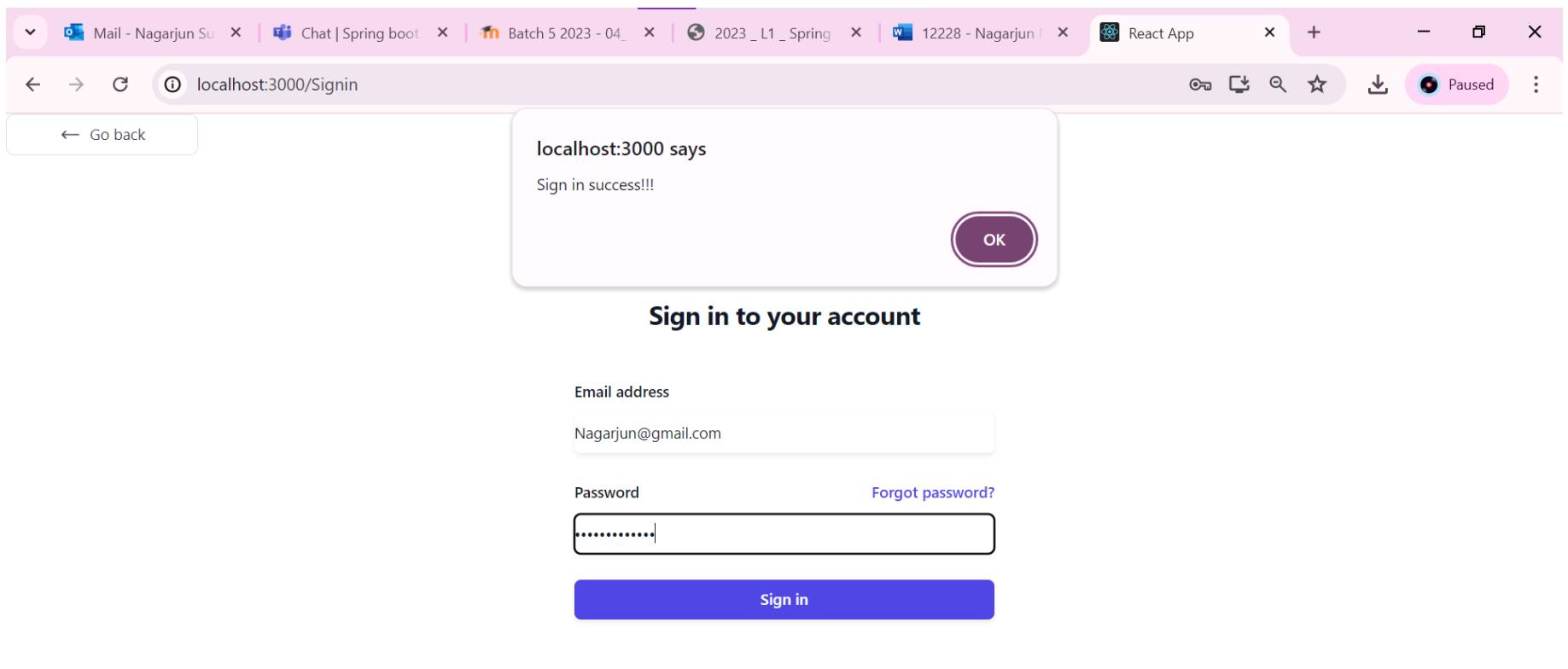
Get Appointment Get Consultant →

© 2024 Care+ Inc. All rights reserved.

Terms License Privacy About us

17:32 24-06-2024

**Sign-In page:**



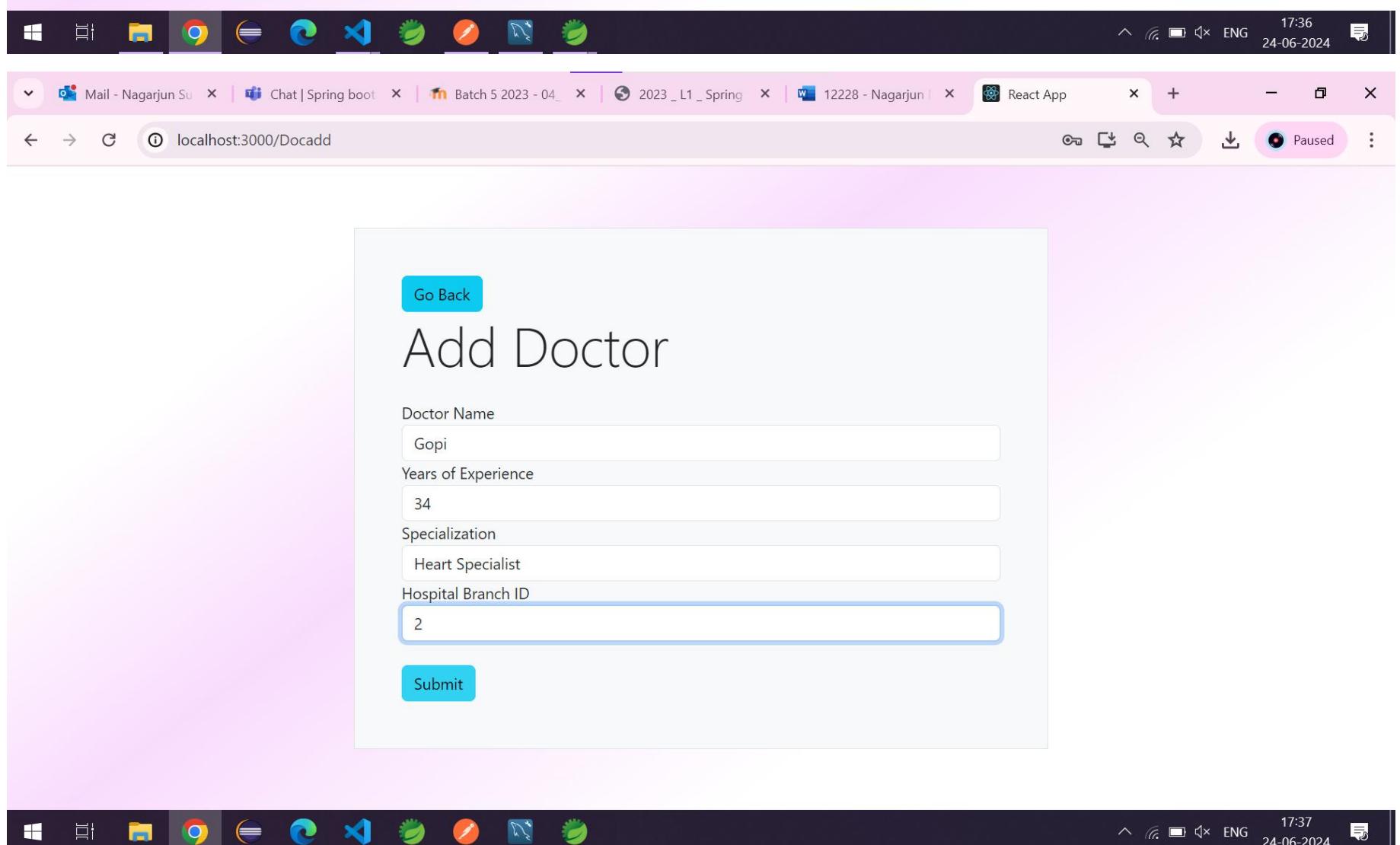
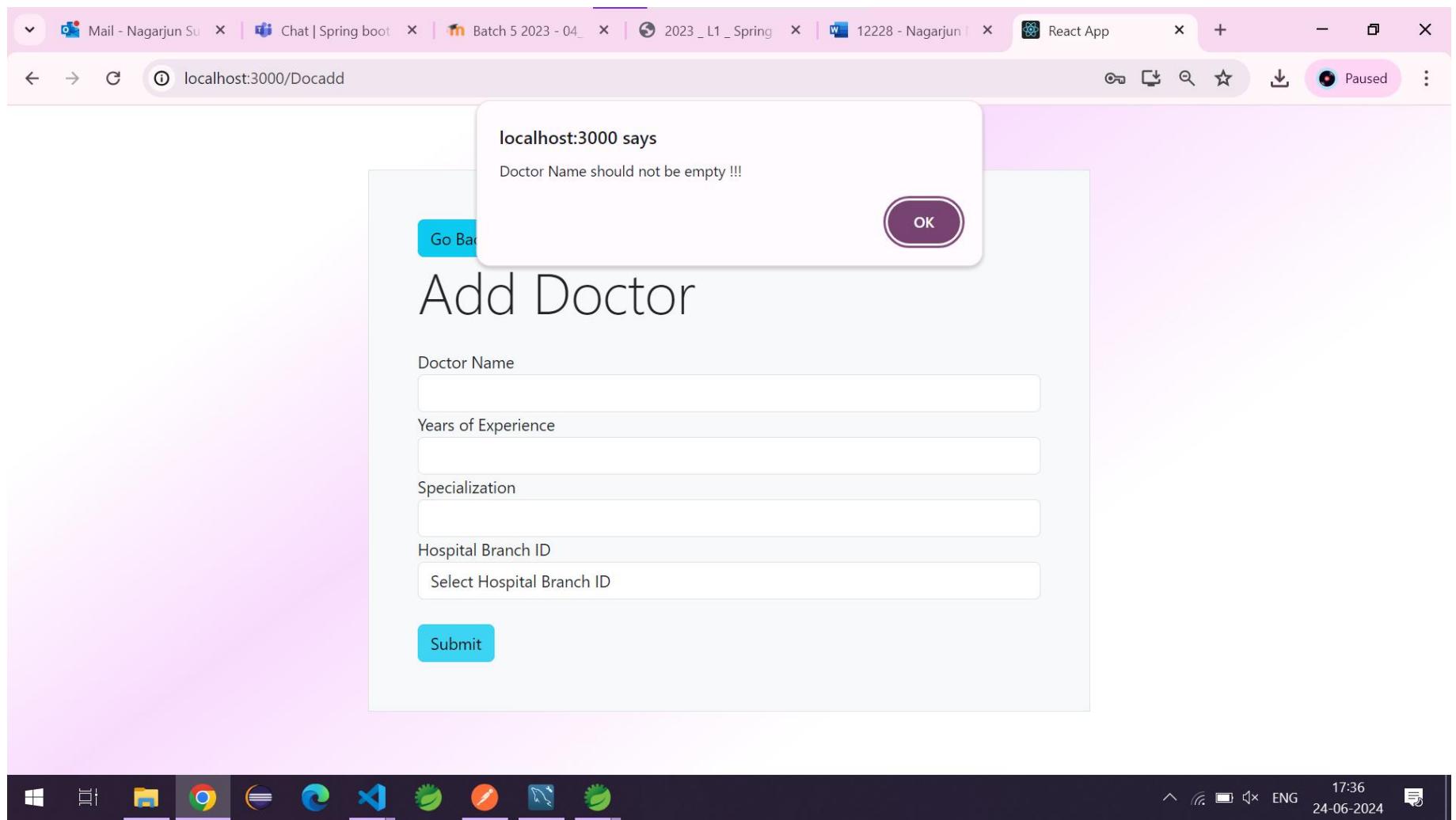
### **DOCTOR VIEWALL:**

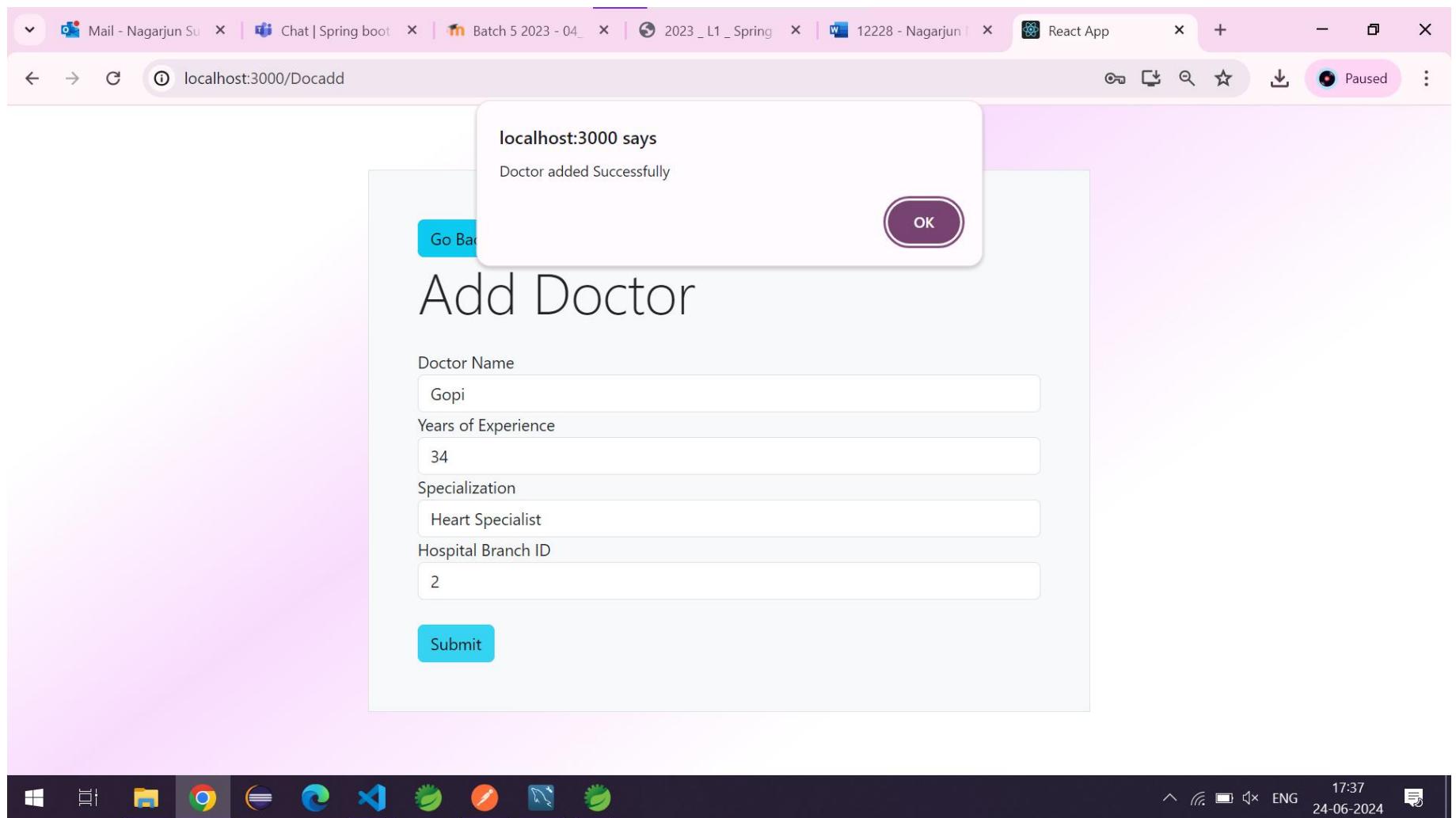
A screenshot of a web browser window titled "localhost:3000/DocView". The page has a header with a "Home" button and a "Doctor Details" title. On the right, there is a blue "+ Add Doctor" button. Below the title is a table with two rows of doctor information. The columns are: Doctor ID, Doctor Name, Years of Experience, Specialization, Hospital ID, and Action. The first row has Doctor ID 1, Doctor Name Nagarjun, Years of Experience 12, Specialization Dentist, Hospital ID 1, and Action buttons for Update, Delete, and Doctor Full Details. The second row has Doctor ID 2, Doctor Name Gokul, Years of Experience 2, Specialization ENT, Hospital ID 2, and Action buttons for Update, Delete, and Doctor Full Details.



### **DOCTOR INSERT:**

### **VALIDATIONS:**





A screenshot of a web browser window titled "localhost:3000/DocView". The page displays a table of doctor details with columns: Doctor ID, Doctor Name, Years of Experience, Specialization, Hospital ID, and Action. The table contains three rows:

Doctor ID	Doctor Name	Years of Experience	Specialization	Hospital ID	Action
1	Nagarjun	12	Dentist	1	<button>Update</button> <button>Delete</button> <button>Doctor Full Details</button>
2	Gokul	2	ENT	2	<button>Update</button> <button>Delete</button> <button>Doctor Full Details</button>
3	Gopi	34	Heart Specialist	2	<button>Update</button> <button>Delete</button> <button>Doctor Full Details</button>

At the top left is a "Home" button, and at the top right is a "+ Add Doctor" button. The browser's address bar shows "localhost:3000/DocView".

**DOCTOR FIND or FULL DETAILS:**

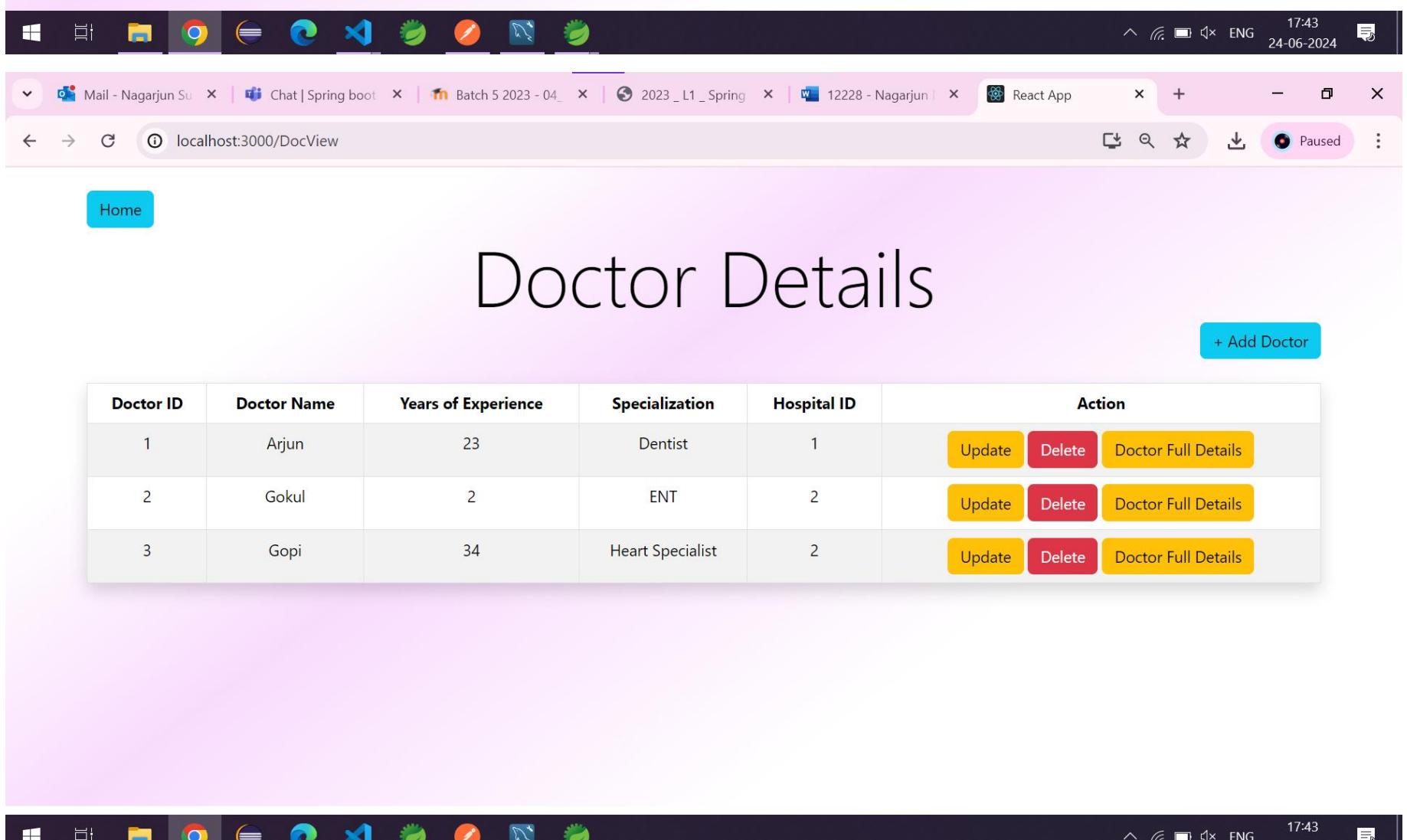
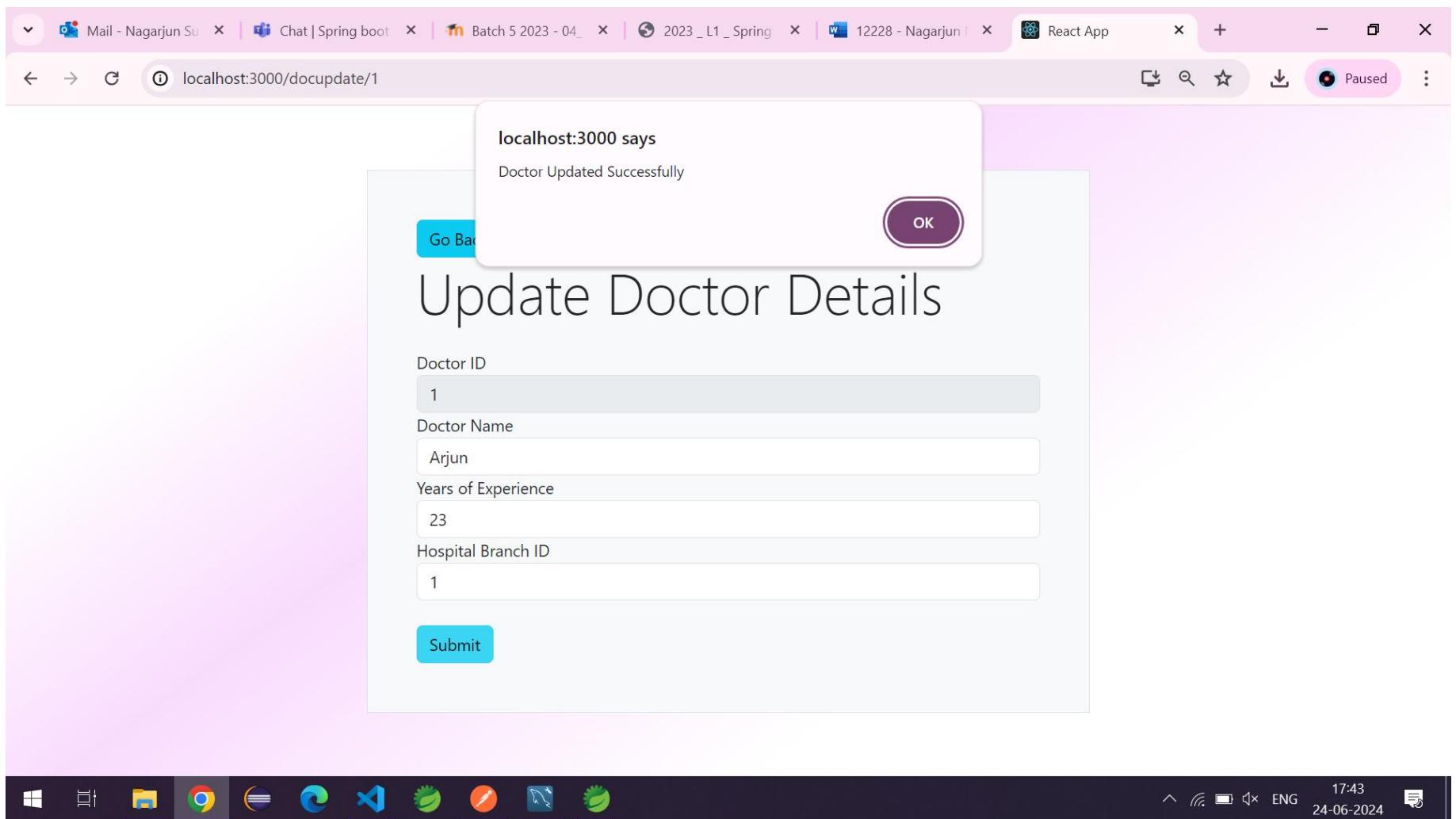
A screenshot of a web browser window titled "Doctor Full Detail". The browser has multiple tabs open, including "Mail - Nagarjun Su", "Chat | Spring boot", "Batch 5 2023 - 04", "2023 \_ L1 \_ Spring", "12228 - Nagarjun", and "React App". The main content area displays a table with the following data:

Doctor ID	Doctor Name	Years of Experience	Specialization	Hospital Branch ID	Hospital Name	City	Hospital Rating
1	Nagarjun	12	Dentist	1	Rajaji	Madurai	4

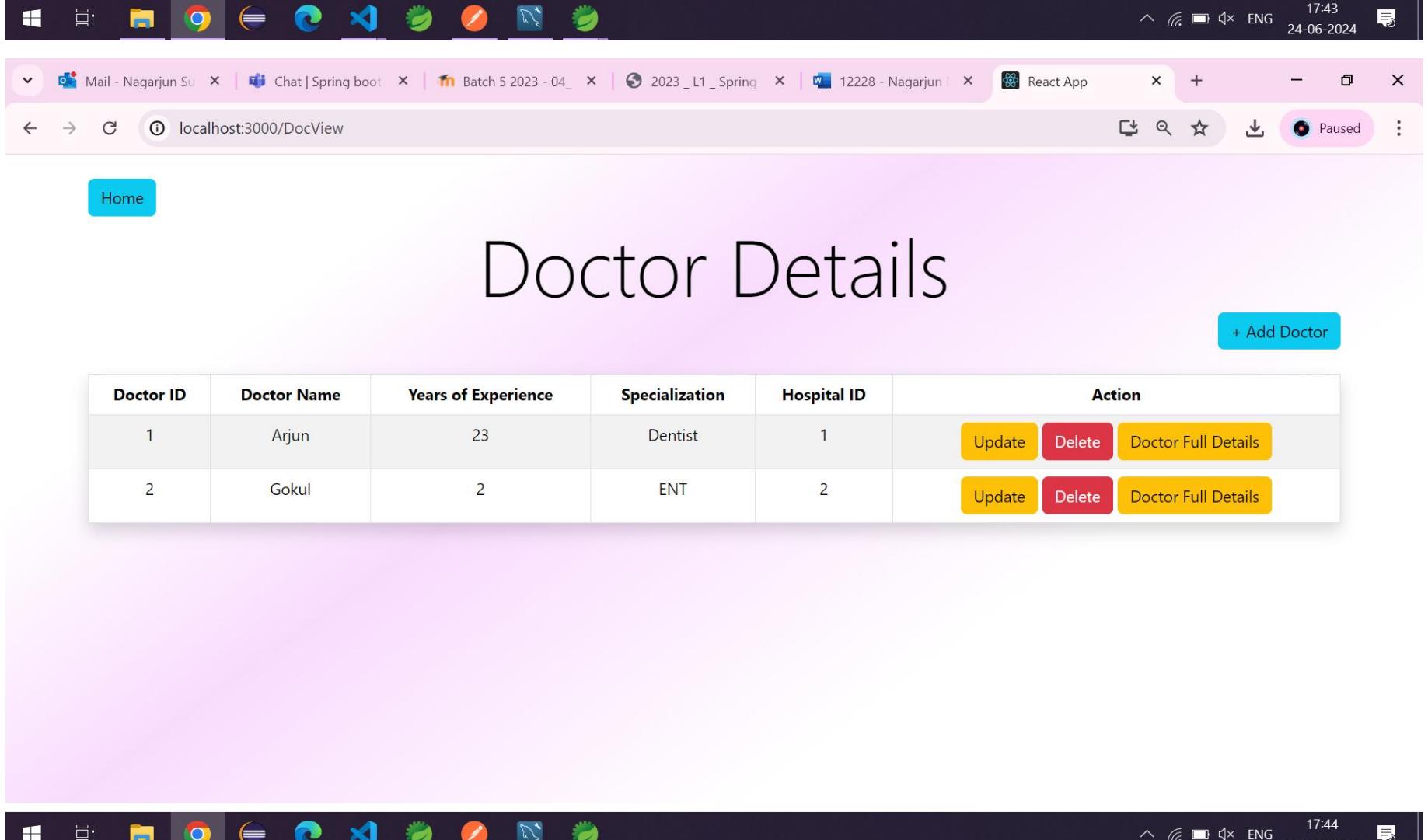
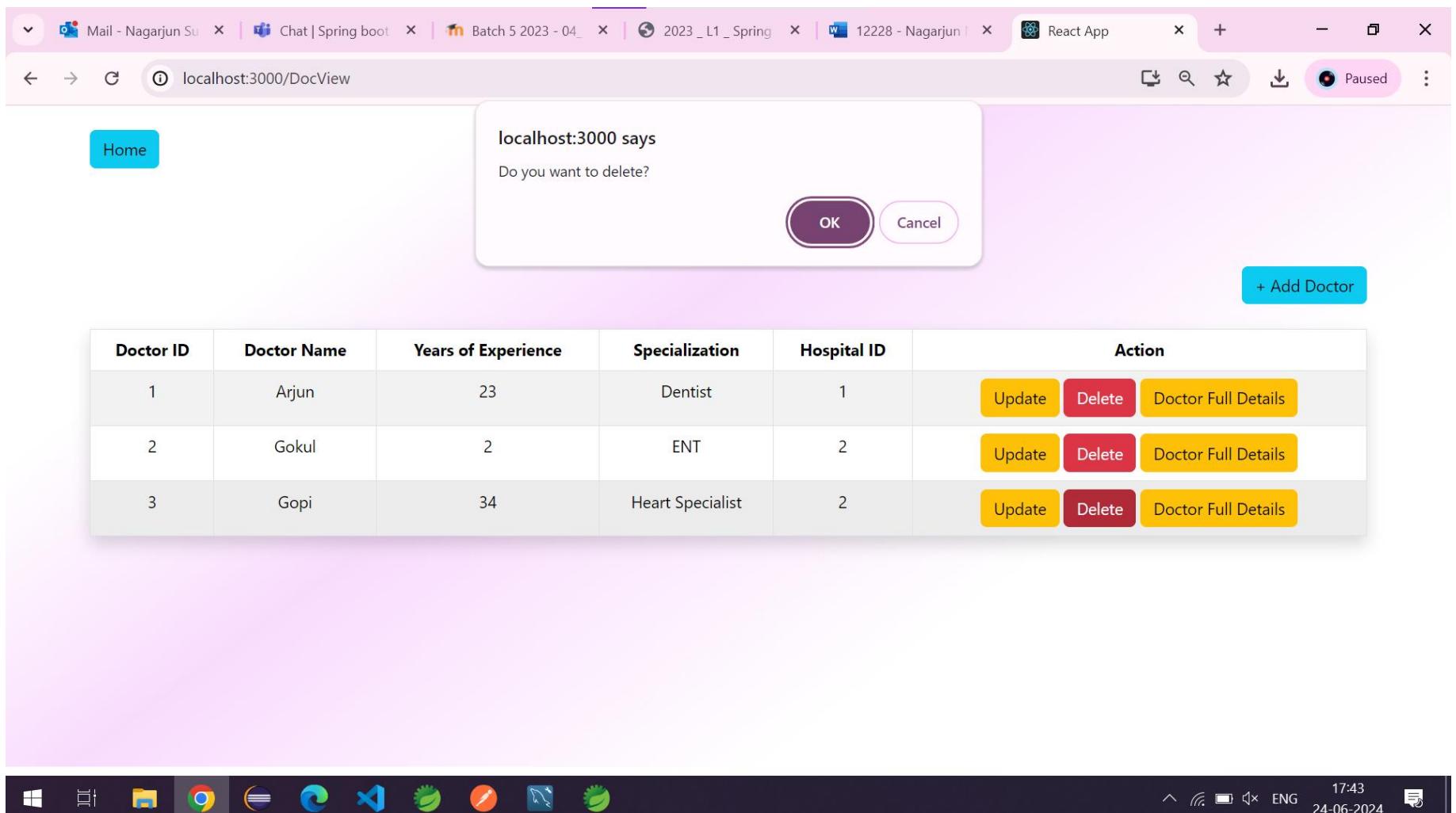
A screenshot of a web browser window titled "Update Doctor Details". The browser has the same tabs as the previous screenshot. The main content area shows a form with the following fields:

- Doctor ID: 1
- Doctor Name: Arjun
- Years of Experience: 23
- Hospital Branch ID: 1

A "Submit" button is visible at the bottom of the form.



**DOCTOR DELETE:**



**HOSPITAL:**

**HOSPITAL VIEWALL:**

A screenshot of a web browser window titled "localhost:3000/Hosview". The page displays a table of hospital information with columns: Hospital ID, Hospital Name, Hospital City, Hospital Rating, and Action. Two rows of data are shown:

Hospital ID	Hospital Name	Hospital City	Hospital Rating	Action
1	Rajaji	Madurai	4	<button>Update</button> <button>Delete</button>
2	Rio Hospital	Chennai	3	<button>Update</button> <button>Delete</button>

At the top left, there is a "Home" button. At the top right, there is a "+ Add Hospital" button. The browser's address bar shows "localhost:3000/Hosview". The system tray at the bottom right shows the date as 24-06-2024 and the time as 17:45.

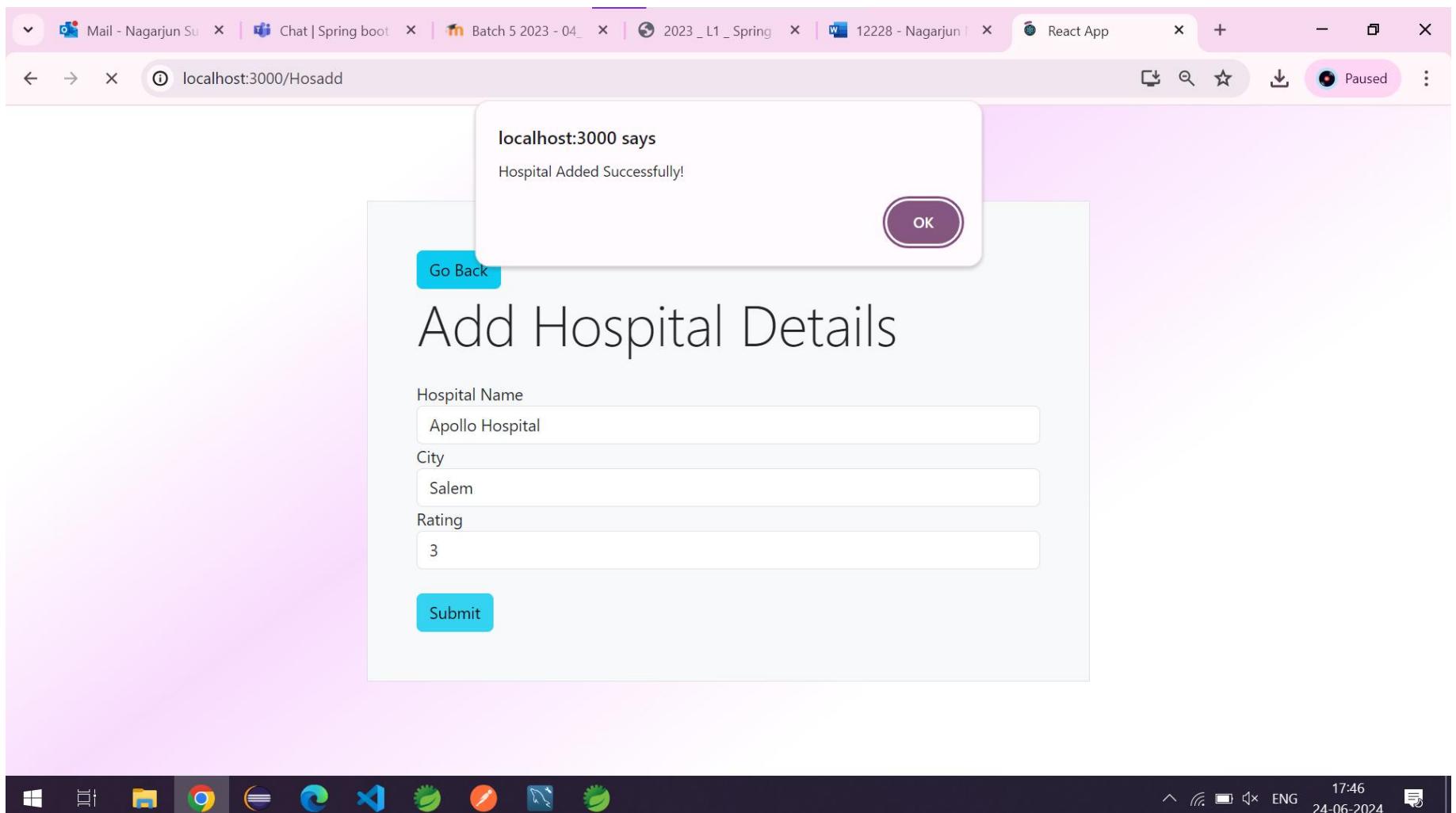
#### HOSPITAL ADD:

A screenshot of a web browser window titled "localhost:3000/Hosadd". The page displays a form titled "Add Hospital Details" with fields for Hospital Name, City, and Rating. A "Go Back" button is located at the top left of the form area.

The form fields are as follows:

- Hospital Name: Apollo Hospital
- City: Salem
- Rating: 3

At the bottom of the form is a "Submit" button. The browser's address bar shows "localhost:3000/Hosadd". The system tray at the bottom right shows the date as 24-06-2024 and the time as 17:45.

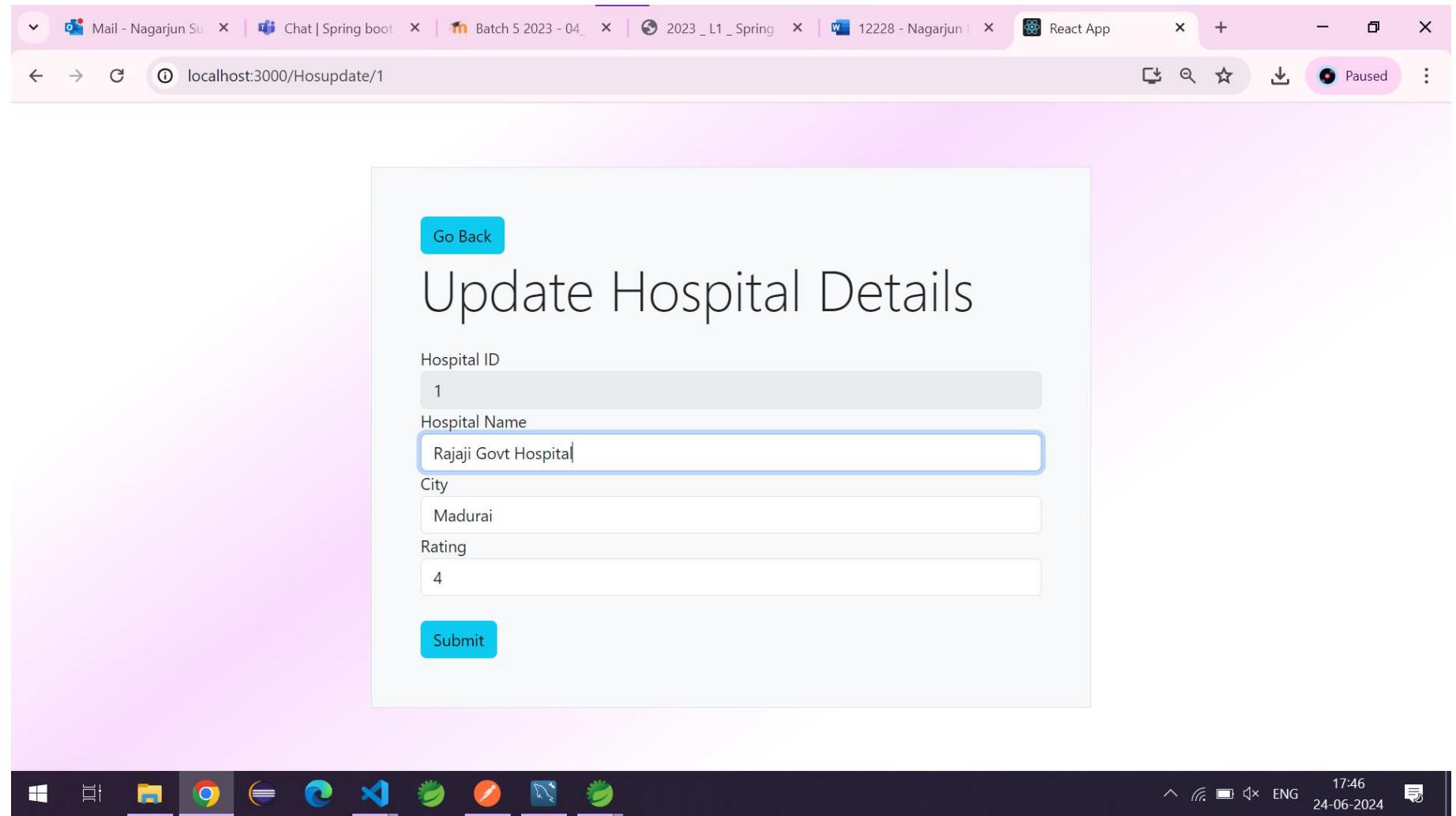


A screenshot of a web browser window titled "localhost:3000/Hosview". The page displays a table of hospital details with three entries:

Hospital ID	Hospital Name	Hospital City	Hospital Rating	Action
1	Rajaji	Madurai	4	<a href="#">Update</a> <a href="#">Delete</a>
2	Rio Hospital	Chennai	3	<a href="#">Update</a> <a href="#">Delete</a>
4	Apollo Hospital	Salem	3	<a href="#">Update</a> <a href="#">Delete</a>

The "Home" button is highlighted in blue at the top left, and a "+ Add Hospital" button is located at the top right. The browser status bar shows the date and time as 24-06-2024 17:46.

## HOSPITAL UPDATE:



localhost:3000/Hosupdate/1

Go Back

## Update Hospital Details

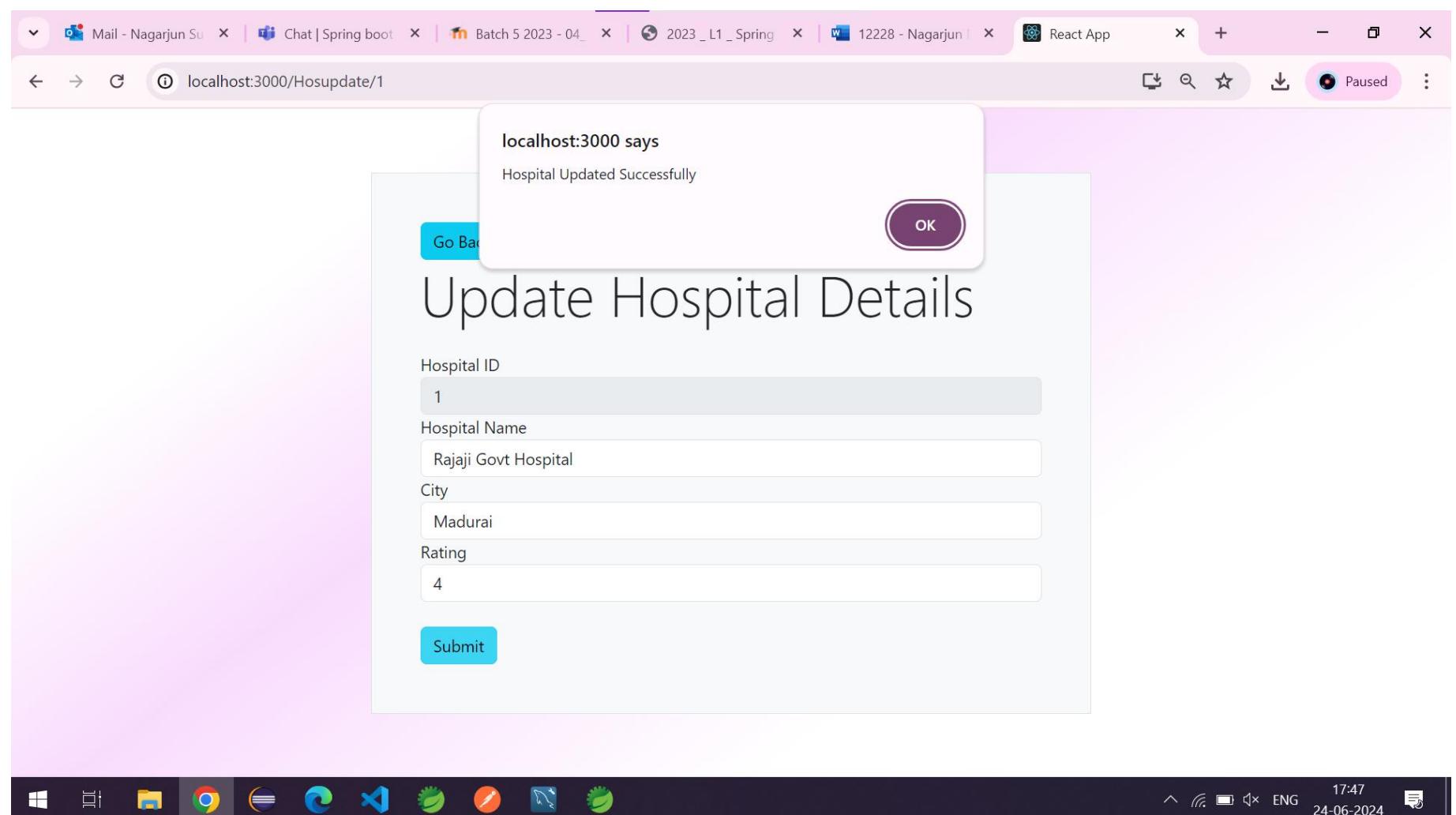
Hospital ID  
1

Hospital Name  
Rajaji Govt Hospital

City  
Madurai

Rating  
4

Submit



localhost:3000 says

Hospital Updated Successfully

OK

localhost:3000 says

## Update Hospital Details

Hospital ID  
1

Hospital Name  
Rajaji Govt Hospital

City  
Madurai

Rating  
4

Submit

A screenshot of a web browser window titled "localhost:3000/Hosview". The page displays a table of hospital data with columns: Hospital ID, Hospital Name, Hospital City, Hospital Rating, and Action. The table contains three rows of data. A blue "Home" button is at the top left, and a green "+ Add Hospital" button is at the top right. The browser's address bar shows the URL "localhost:3000/Hosview". The system tray at the bottom shows various icons and the date/time "24-06-2024 17:47".

Hospital ID	Hospital Name	Hospital City	Hospital Rating	Action
1	Rajaji Govt Hospital	Madurai	4	<button>Update</button> <button>Delete</button>
2	Rio Hospital	Chennai	3	<button>Update</button> <button>Delete</button>
4	Apollo Hospital	Salem	3	<button>Update</button> <button>Delete</button>

#### HOSPITAL DELETE:

A screenshot of a web browser window titled "localhost:3000/Hosview". A confirmation dialog box is centered on the screen with the message "localhost:3000 says Do you want to delete?". It has two buttons: "OK" (highlighted) and "Cancel". The browser's address bar shows the URL "localhost:3000/Hosview". The system tray at the bottom shows various icons and the date/time "24-06-2024 17:48".

Hospital ID	Hospital Name	Hospital City	Hospital Rating	Action
1	Rajaji Govt Hospital	Madurai	4	<button>Update</button> <button>Delete</button>
2	Rio Hospital	Chennai	3	<button>Update</button> <button>Delete</button>
4	Apollo Hospital	Salem	3	<button>Update</button> <button>Delete</button>

Screenshot of a web browser showing a Spring Boot application at [localhost:3000/Hosview](http://localhost:3000/Hosview). The page displays "Hospital Details" with two entries in a table:

Hospital ID	Hospital Name	Hospital City	Hospital Rating	Action
1	Rajaji Govt Hospital	Madurai	4	<button>Update</button> <button>Delete</button>
2	Rio Hospital	Chennai	3	<button>Update</button> <button>Delete</button>

A blue button "+ Add Hospital" is located in the top right corner. The browser's tab bar shows multiple open tabs related to the project.

#### DB TABLE STRUCTURE:

Screenshot of MySQL Workbench showing the database structure for the "models\_manyone" schema.

The Schemas pane lists available databases: account, books, cake, casc, courier, creche, cricket, emproll, food\_db, hql, hql, jpa\_db.

The Query editor pane contains the following SQL code:

```

1 • create database hospitaldb;
2 • use hospitaldb;
3 • show tables;
4 • desc doctor;
5 • select * from doctor;
6 • select * from hospital;
7 • truncate table doctor;

```

The Result Grid pane shows the structure of the "doctor" table:

Field	Type	Null	Key	Default	Extra
doctor_id	int	NO	PRI	HULL	auto_increment
doctor_experience	varchar(255)	YES		HULL	
doctor_name	varchar(255)	YES		HULL	
specialization	varchar(255)	YES		HULL	
hospital_hospital_id	int	YES	MUL	HULL	

The Session pane shows the following history:

#	Time	Action	Message	Duration / Fetch
224	15:44:47	use hospitaldb	0 row(s) affected	0.000 sec
225	15:44:48	show tables	0 row(s) returned	0.000 sec / 0.000 sec
226	17:49:30	desc hospitaldb	Error Code: 1146. Table 'hospitaldb.hospitaldb' doesn't exist	0.016 sec
227	17:49:34	desc doctor	5 row(s) returned	0.016 sec / 0.000 sec

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: models\_manyone

SCHEMAS

Filter objects

account books cake casc courier creche cricket emproll food\_db hcql hql jpa\_db

Administration Schemas

No object selected

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid

Field Type Null Key Default Extra

hospital\_id int NO PRI NULL auto\_increment

city varchar(255) YES NULL

hospital\_name varchar(255) YES NULL

rating int YES NULL

Action Output

# Time Action Message

226 17:49:30 desc hospitaldb Error Code: 1146. Table 'hospitaldb.hospitaldb' doesn't exist

227 17:49:34 desc doctor 5 row(s) returned

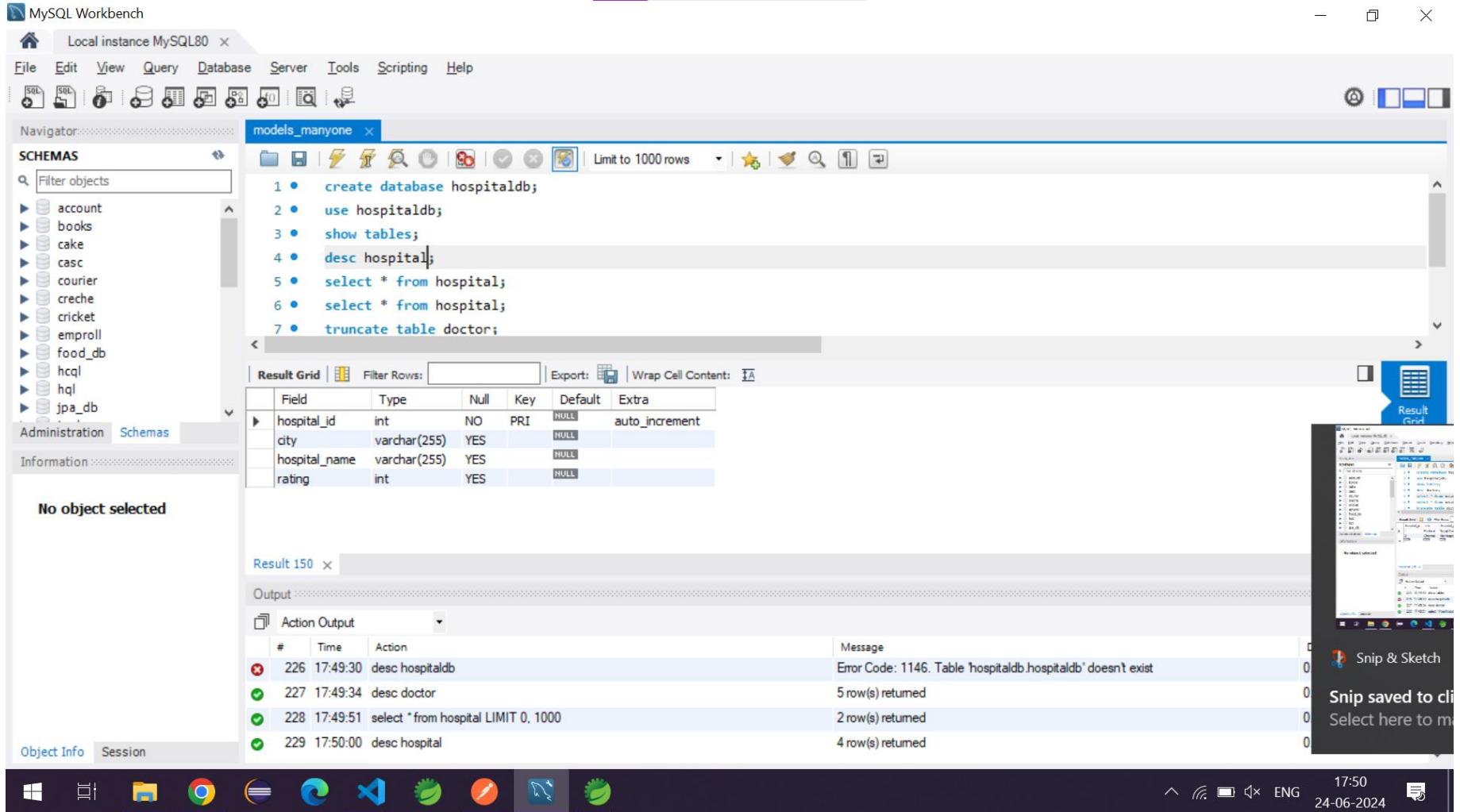
228 17:49:51 select \* from hospital LIMIT 0, 1000 2 row(s) returned

229 17:50:00 desc hospital 4 row(s) returned

Object Info Session

Snip & Sketch Snip saved to clipboard Select here to manage

17:50 24-06-2024



## POSTMAN SS:

### POST Method:

Home Workspaces Explore Search Postman Sign In Create Account

You are using the Lightweight API Client, sign in or create an account to work with collections, environments and unlock all free features in Postman.

History New Import DEL http://localhost:1234/doctor [CONFLICT] POST http://localhost:1234/doctor PUT http://localhost:1234/doctor/1 POST http://localhost:1234/doctor/all GET http://localhost:1234/doctor/hospital POST http://localhost:1234/doctor/hospital/1 PUT http://localhost:1234/doctor/hospital/1/2 POST http://localhost:1234/doctor/hospital/1/2/3 POST http://localhost:1234/doctor/hospital/1/2/3/4 POST http://localhost:1234/doctor/hospital/1/2/3/4/5 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51/52 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51/52/53 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51/52/53/54 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51/52/53/54/55 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51/52/53/54/55/56 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51/52/53/54/55/56/57 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51/52/53/54/55/56/57/58 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51/52/53/54/55/56/57/58/59 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51/52/53/54/55/56/57/58/59/60 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51/52/53/54/55/56/57/58/59/60/61 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51/52/53/54/55/56/57/58/59/60/61/62 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51/52/53/54/55/56/57/58/59/60/61/62/63 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51/52/53/54/55/56/57/58/59/60/61/62/63/64 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51/52/53/54/55/56/57/58/59/60/61/62/63/64/65 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51/52/53/54/55/56/57/58/59/60/61/62/63/64/65/66 POST http://localhost:1234/doctor/hospital/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19

## GET METHOD:

The screenshot shows the Postman interface with a successful GET request to `http://localhost:1234/doctor/2`. The response status is 200 OK, and the response body is a JSON object representing a doctor's profile:

```
1 "did": 2,
2 "dname": "Gokul",
3 "dexperience": "2",
4 "dspecial": "ENT",
5 "hospital": {
6     "hid": 2,
7     "hname": "Rio Hospital",
8     "hcity": "Chennai",
9     "hstate": "Tamil Nadu"
10 }
```

## PUT METHOD:

The screenshot shows the Postman interface with a successful PUT request to `http://localhost:1234/doctor`. The response status is 200 OK, and the response body is a JSON object representing a doctor's profile:

```
1 {"did":4,
2 "dname":"Senthil",
3 "dexperience":12,
4 "dspecial":"Bone Marrow Specialist",
5 "hospital":{
6     "hid":2,
7     "hname":"Rio Hospital",
8     "hcity":"Chennai",
9     "hstate":"Tamil Nadu"
10 }
```

The response body also includes the message "Success".

## GET ALL:

You are using the Lightweight API Client, sign in or create an account to work with collections, environments and **unlock all free features in Postman**.

**History**

New Import

DEL http://lc PUT http://lc [CONFLICT] POST http:// POST http:// PUT http://lc GET http://lc POST http:// GET http://lc

HTTP http://localhost:1234/doctor

Save </>

Today

GET http://localhost:1234/doctor/all  
PUT http://localhost:1234/doctor  
GET http://localhost:1234/doctor/2  
GET http://localhost:1234/doctor/3  
GET http://localhost:1234/doctor/4  
POST http://localhost:1234/doctor  
POST http://localhost:1234/hospital  
POST http://localhost:1234/hospital  
GET http://localhost:1234/doctor/all  
POST http://localhost:1234/doctor  
POST http://localhost:1234/hospital  
POST http://localhost:1234/doctor  
GET http://localhost:1234/doctor/all  
PUT http://localhost:1234/doctor/all

GET http://localhost:1234/doctor/all

Send

Params Auth Headers (8) Body Pre-req. Tests Settings Cookies Beautify

raw JSON

Body

200 OK 11 ms 694 B Save Response

Pretty Raw Preview Visualize JSON

hid: 1,  
hname: "Rajaji Govt Hospital",  
hcity: "Madurai",  
hrating: 4

did: 2,  
dname: "Gokul",

Console Not connected to a Postman account

17:53 24-06-2024 ENG

## **DELETE:**

The screenshot shows the Postman Lightweight API Client interface. The top navigation bar includes 'Home', 'Workspaces', 'Explore', a search bar 'Search Postman', 'Sign In', 'Create Account', and standard window controls. A message at the top states: 'You are using the Lightweight API Client, sign in or create an account to work with collections, environments and unlock all free features in Postman.' The left sidebar, titled 'History', lists various API requests made today, such as 'DELETE http://localhost:1234/doctor/4', 'GET http://localhost:1234/doctor/all', and multiple 'POST' and 'PUT' requests. The main workspace displays a 'DELETE' request to 'http://localhost:1234/doctor/4'. The 'Headers' tab shows 8 items. The 'Body' tab is selected, showing the response body: '1 Success'. The status bar indicates a 200 OK response with 23 ms and 259 B. The bottom navigation bar includes 'Console' and a note 'Not connected to a Postman account'.