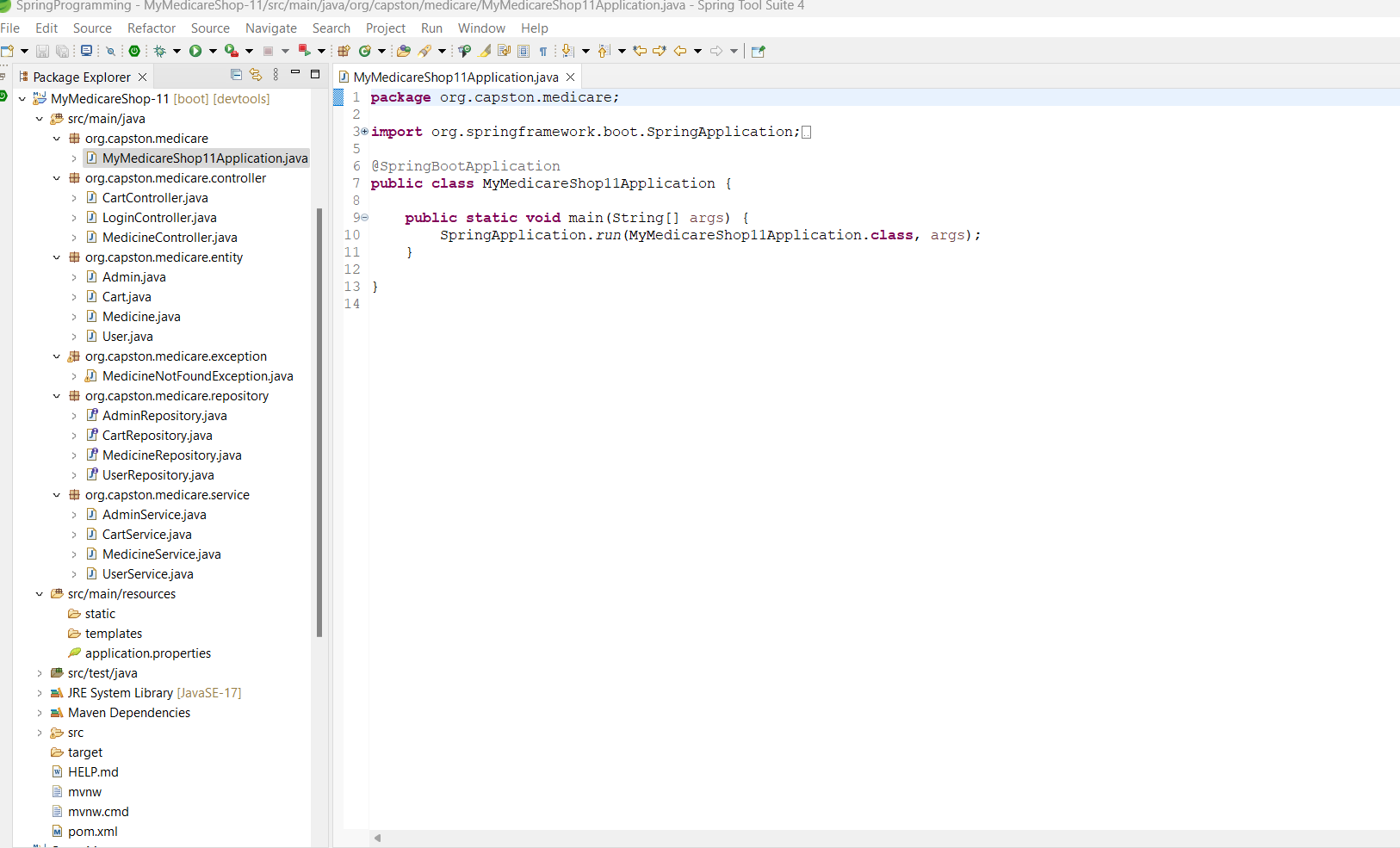
GitHubLink:

1)Backend Code



* CartController:

package org.capston.medicare.controller;

import java.util.List;

import org.capston.medicare.entity.Cart;

import org.capston.medicare.entity.Medicine;

import org.capston.medicare.entity.User;

import org.capston.medicare.exception.MedicineNotFoundException;

import org.capston.medicare.service.CartService;

import org.capston.medicare.service.MedicineService;

import org.capston.medicare.service.UserService;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

@CrossOrigin("http://localhost:4200/")

@RequestMapping("/cart")

public class CartController {

@Autowired

private CartService cartService;

@Autowired

private UserService userService;

@Autowired

private MedicineService medicineService;

@GetMapping("/getAll/{userId}")

public List<Cart> getCart(@PathVariable long userId)

{

return cartService.getAllByUserCart(userId);

}

@GetMapping("/add/{medicineId}/{userId}")

public ResponseEntity<Cart> addCart(@PathVariable("medicineId") long medicineId,@PathVariable("userId") long userId) throws MedicineNotFoundException

{

Medicine medicine=medicineService.getMedicineById(medicineId);

User user=userService.getUserById(userId);

Cart cart=new Cart(user, medicine);

cart.setUnit(1);

cart.setTotalPrice(cart.getUnit()\* ((Medicine) medicine).getPrice());

Cart addCart=cartService.addToCart(cart);

return new ResponseEntity<Cart>(addCart,HttpStatus.CREATED);

}

@DeleteMapping("/delete/{cartId}")

public ResponseEntity<?> deleteCart(@PathVariable long cartId) {

cartService.deleteCart(cartId);

return new ResponseEntity<>(HttpStatus.OK);

}

@PutMapping("/update-cart/{cid}")

public void updateCart(@RequestBody Cart cart, @PathVariable long cid) {

cartService.updateCart(cart, cid);

}

}

* LoginController:

package org.capston.medicare.controller;

import java.util.List;

import org.capston.medicare.entity.Admin;

import org.capston.medicare.entity.User;

import org.capston.medicare.service.AdminService;

import org.capston.medicare.service.UserService;

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;

@RestController

@RequestMapping("/welcome")

@CrossOrigin("http://localhost:4200")

public class LoginController {

private AdminService adminService;

private UserService userService;

public LoginController(AdminService adminService, UserService userService) {

super();

this.adminService = adminService;

this.userService = userService;

}

// Admin

@PostMapping("/add\_admin")

public String addAdmin(@RequestBody Admin admin) {

return adminService.addAdmin(admin);

}

@GetMapping("/getAll\_admin")

public List<Admin> getAllAdmin() {

return adminService.getAllAdmin();

}

//User

@PostMapping("/add\_user")

public User addUser(@RequestBody User user)

{

return userService.addUser(user);

}

@GetMapping("/getAll\_user")

public List<User> getAllUser(){

return userService.getAllUsers();

}

@GetMapping("/getuser/{id}")

public User getUserById(@PathVariable long id)

{

return userService.getUserById(id);

}

@PutMapping("/update\_user")

public User updateUser(@RequestBody User user)

{

return userService.updateUser(user);

}

@DeleteMapping("/delete\_user/{id}")

public String deleteUser(@PathVariable("id")Long id)

{

return userService.deleteUser(id);

}

}

* MedicineController:

package org.capston.medicare.controller;

import java.util.List;

import org.capston.medicare.entity.Medicine;

import org.capston.medicare.exception.MedicineNotFoundException;

import org.capston.medicare.service.MedicineService;

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;

@RestController

@RequestMapping("/medicines")

@CrossOrigin("http://localhost:4200")

public class MedicineController {

public static String uploadDir = "medicine-imageName";

private MedicineService medicineService;

public MedicineController(MedicineService medicineService) {

super();

this.medicineService = medicineService;

}

@PostMapping("/add\_medicine" )

public Medicine addMedicine(@RequestBody Medicine m)

{

return medicineService.insertMedicine(m);

}

@GetMapping("/getAll\_medicine")

public List<Medicine> getAllMedicines() {

return medicineService.getMedicines();

}

@GetMapping("/getMedicineById")

public Medicine getMedicineById(@PathVariable("id") long id) throws MedicineNotFoundException {

return medicineService.getMedicineById(id);

}

@DeleteMapping("/delete\_medicine/{id}")

public String deleteMedicine(@PathVariable("id") long id) {

return medicineService.deleteMedicine(id);

}

@PutMapping("/update\_medicine")

public String updateMedicine(@RequestBody Medicine m) throws MedicineNotFoundException

{

return medicineService.updateMedicine( m);

}

@GetMapping("/get-sorted-product")

public List<Medicine> getSortedProducts(){

return medicineService.getSortedMedicine();

}

}

* AdminEntity:

package org.capston.medicare.entity;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

@Entity

public class Admin {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Long id;

private String email;

private String password;

public Admin() {

// TODO Auto-generated constructor stub

}

public Admin(String email, String password) {

super();

this.email = email;

this.password = password;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

* CartEntity:

package org.capston.medicare.entity;

import com.fasterxml.jackson.annotation.JsonIgnore;

import jakarta.persistence.CascadeType;

import jakarta.persistence.Entity;

import jakarta.persistence.FetchType;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.JoinColumn;

import jakarta.persistence.OneToOne;

@Entity

public class Cart {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private Integer unit;

private Double totalPrice;

public Cart() {

// TODO Auto-generated constructor stub

}

@OneToOne(cascade = CascadeType.ALL,fetch = FetchType.EAGER)

@JsonIgnore

@JoinColumn(nullable = false, name = "user\_id")

private User user;

@OneToOne

@JsonIgnore

private Medicine medicine;

public Cart(Long id, Integer unit, Double totalPrice) {

super();

this.id = id;

this.unit = unit;

this.totalPrice = totalPrice;

}

public Cart(User user,Medicine medicine) {

super();

this.user = user;

this.medicine = medicine;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public Integer getUnit() {

return unit;

}

public void setUnit(Integer unit) {

this.unit = unit;

}

public Double getTotalPrice() {

return totalPrice;

}

public void setTotalPrice(Double totalPrice) {

this.totalPrice = totalPrice;

}

public User getUser() {

return user;

}

public void setUser(User user) {

this.user = user;

}

public Medicine getMedicine() {

return medicine;

}

public void setMedicine(Medicine medicine) {

this.medicine = medicine;

}

}

* MedicineEntity:

package org.capston.medicare.entity;

import com.fasterxml.jackson.annotation.JsonIgnore;

import jakarta.persistence.CascadeType;

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;

@Entity

public class Medicine {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Long id;

private String name;

private Double price;

private String imageName;

private String category;

public Medicine() {

// TODO Auto-generated constructor stub

}

@ManyToOne(cascade = CascadeType.ALL)

@JoinColumn(name="user\_id" )

private User user;

public User getUser() {

return user;

}

public void setUser(User user) {

this.user = user;

}

@OneToOne

@JoinColumn(name = "medicine\_id")

@JsonIgnore

private Cart cart;

public Cart getCart() {

return cart;

}

public void setCart(Cart cart) {

this.cart = cart;

}

public Medicine(String name, Double price, String imageName, String category) {

super();

this.name = name;

this.price = price;

this.imageName = imageName;

this.category = category;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Double getPrice() {

return price;

}

public void setPrice(Double price) {

this.price = price;

}

public String getImageName() {

return imageName;

}

public void setImageName(String imageName) {

this.imageName = imageName;

}

public String getCategory() {

return category;

}

public void setCategory(String category) {

this.category = category;

}

* UserEntity

package org.capston.medicare.entity;

import java.util.ArrayList;

import java.util.List;

import com.fasterxml.jackson.annotation.JsonIgnore;

import jakarta.persistence.CascadeType;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.FetchType;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.OneToMany;

import jakarta.persistence.OneToOne;

@Entity

public class User {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

@Column(nullable = false, name="user\_id")

private Long id;

private String fullname;

private String email;

private String password;

public User() {

// TODO Auto-generated constructor stub

}

public List<Medicine> getMedicine() {

return medicine;

}

public void setMedicine(List<Medicine> medicine) {

this.medicine = medicine;

}

@OneToMany(mappedBy = "user" ,fetch = FetchType.EAGER)

private List<Medicine> medicine=new ArrayList<>();

@OneToOne(mappedBy ="user", cascade = CascadeType.ALL)

@JsonIgnore

private Cart cart;

public Cart getCart() {

return cart;

}

public void setCart(Cart cart) {

this.cart = cart;

}

public User(long id,String fullname, String email, String password) {

super();

this.id=id;

this.fullname = fullname;

this.email = email;

this.password = password;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getFullname() {

return fullname;

}

public void setFullname(String fullname) {

this.fullname = fullname;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

}

* AdminRepo

package org.capston.medicare.repository;

import org.capston.medicare.entity.Admin;

import org.springframework.data.jpa.repository.JpaRepository;

public interface AdminRepository extends JpaRepository<Admin, Long>{

}

* UserRepo

package org.capston.medicare.repository;

import java.util.List;

import org.capston.medicare.entity.Cart;

import org.springframework.data.jpa.repository.JpaRepository;

public interface CartRepository extends JpaRepository<Cart, Long>{

List<Cart> findAllByUserId(Long id);

}

* CartRepo
* package org.capston.medicare.repository;
* import org.capston.medicare.entity.User;
* import org.springframework.data.jpa.repository.JpaRepository;
* public interface UserRepository extends JpaRepository<User, Long>{
* }MedicineRepo

package org.capston.medicare.repository;

import org.capston.medicare.entity.Medicine;

import org.springframework.data.jpa.repository.JpaRepository;

public interface MedicineRepository extends JpaRepository<Medicine, Long>{

}

* AdminService:

package org.capston.medicare.service;

import java.util.List;

import org.capston.medicare.entity.Admin;

import org.capston.medicare.repository.AdminRepository;

import org.springframework.stereotype.Service;

@Service

public class AdminService {

private AdminRepository adminRepository;

public AdminService(AdminRepository adminRepository) {

super();

this.adminRepository = adminRepository;

}

public String addAdmin(Admin admin) {

Admin admin2 = new Admin();

admin2.setEmail(admin.getEmail());

admin2.setPassword(admin.getPassword());

adminRepository.save(admin2);

return "New Admin Saved";

}

public List<Admin> getAllAdmin() {

return adminRepository.findAll();

}

}

* CartService:

package org.capston.medicare.service;

import java.util.List;

import org.capston.medicare.entity.Cart;

import org.capston.medicare.repository.CartRepository;

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

import org.springframework.stereotype.Service;

@Service

public class CartService {

@Autowired

private CartRepository cartRepository;

public Cart addToCart(Cart cart) {

return cartRepository.save(cart);

}

public List<Cart> getAllByUserCart(long id) {

return cartRepository.findAllByUserId(id);

}

public void updateCart(Cart cart, long id) {

Cart cartUpdate = cartRepository.findById(id).get();

cartUpdate = cart;

cartRepository.save(cartUpdate);

}

public void deleteCart(long id) {

cartRepository.deleteById(id);

}

* MedicineService:

package org.capston.medicare.service;

import java.util.List;

import org.capston.medicare.entity.Medicine;

import org.capston.medicare.exception.MedicineNotFoundException;

import org.capston.medicare.repository.MedicineRepository;

import org.springframework.data.domain.Sort;

import org.springframework.stereotype.Service;

@Service

public class MedicineService {

private MedicineRepository medicineRepository;

public MedicineService(MedicineRepository medicineRepository) {

super();

this.medicineRepository = medicineRepository;

}

public Medicine insertMedicine(Medicine m) {

Medicine medicine=new Medicine();

medicine.setName(m.getName());

medicine.setPrice(m.getPrice());

medicine.setImageName(m.getImageName());

medicine.setCategory(m.getCategory());

return medicineRepository.save(medicine);

}

public List<Medicine> getMedicines() {

return medicineRepository.findAll();

}

public List<Medicine> getSortedMedicine(){

return medicineRepository.findAll(Sort.by("category"));

}

public Medicine getMedicineById(Long id) throws MedicineNotFoundException {

return medicineRepository.findById(id).get();

}

public String deleteMedicine(Long id) {

medicineRepository.deleteById(id);

return "Medicine Deleted by Id ";

}

public String updateMedicine(Medicine m) throws MedicineNotFoundException {

Medicine me=new Medicine();

me.setId(m.getId());

me.setName(m.getName());

me.setPrice(m.getPrice());

me.setImageName(m.getImageName());

me.setCategory(m.getCategory());

medicineRepository.save(me);

return "Data Updated";

}

}

* UserService:

package org.capston.medicare.service;

import java.util.List;

import org.capston.medicare.entity.User;

import org.capston.medicare.repository.UserRepository;

import org.springframework.stereotype.Service;

@Service

public class UserService {

private UserRepository userRepository;

public UserService(UserRepository userRepository) {

super();

this.userRepository = userRepository;

}

public User addUser(User user) {

User users = new User();

users.setId(user.getId());

users.setFullname(user.getFullname());

users.setEmail(user.getEmail());

users.setPassword(user.getPassword());

return userRepository.save(users);

}

public List<User> getAllUsers() {

return userRepository.findAll();

}

public User getUserById(long id) {

return userRepository.findById(id).get();

}

public User updateUser(User user) {

User user1 = new User();

user1.setId(user.getId());

user1.setFullname(user.getFullname());

user1.setEmail(user.getEmail());

user1.setPassword(user.getPassword());

return userRepository.save(user1);

}

public String deleteUser(Long id) {

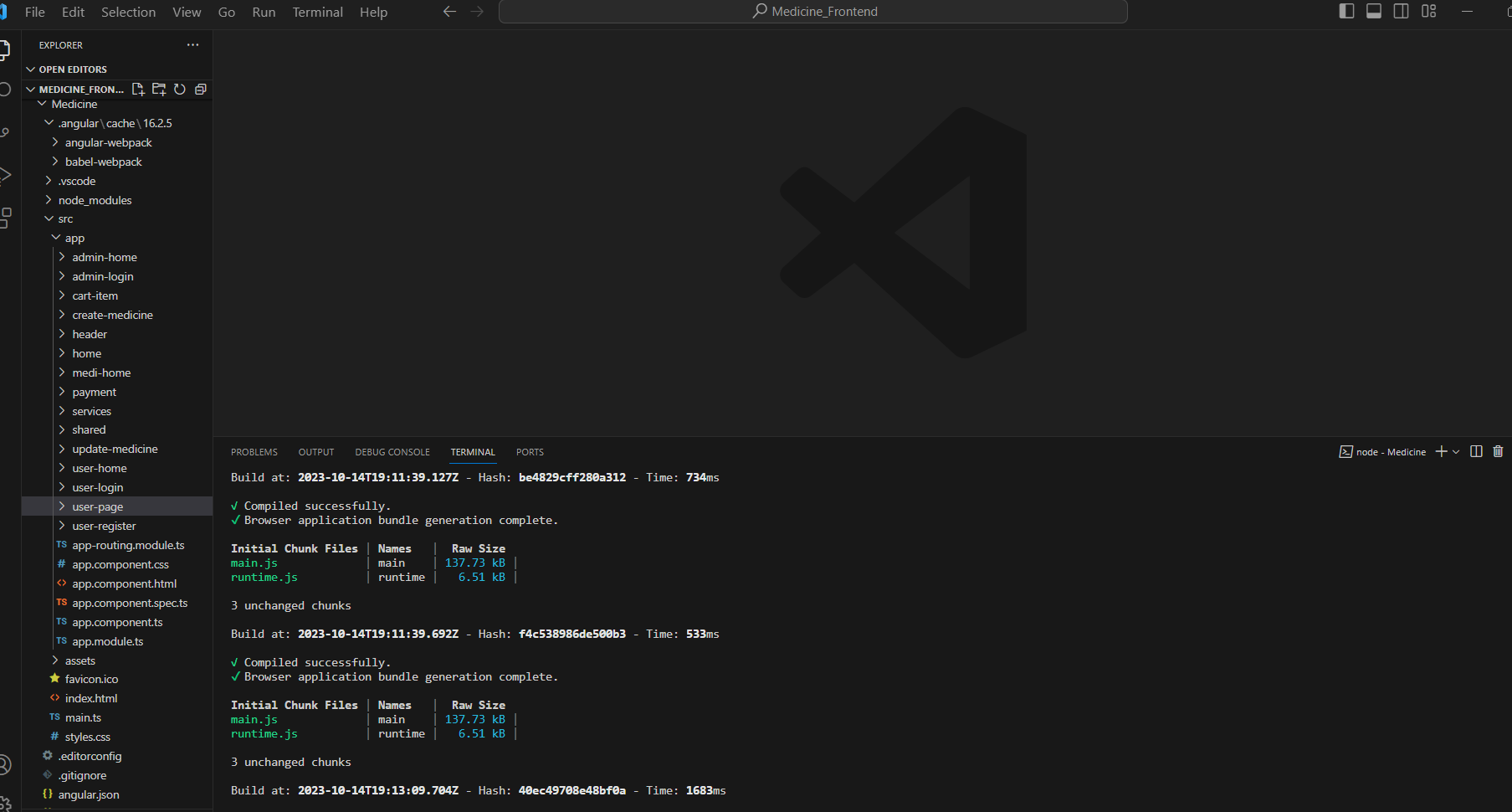
userRepository.deleteById(id);

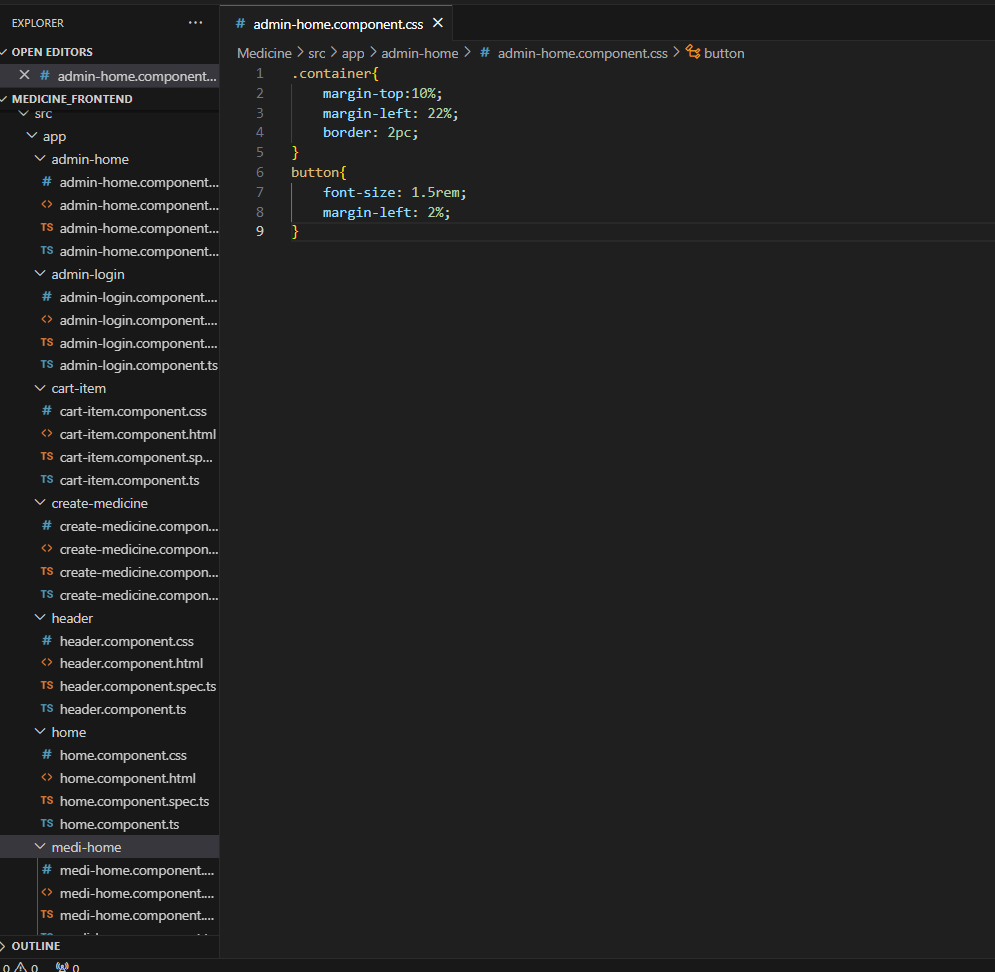
return "User Deleted Successfully";

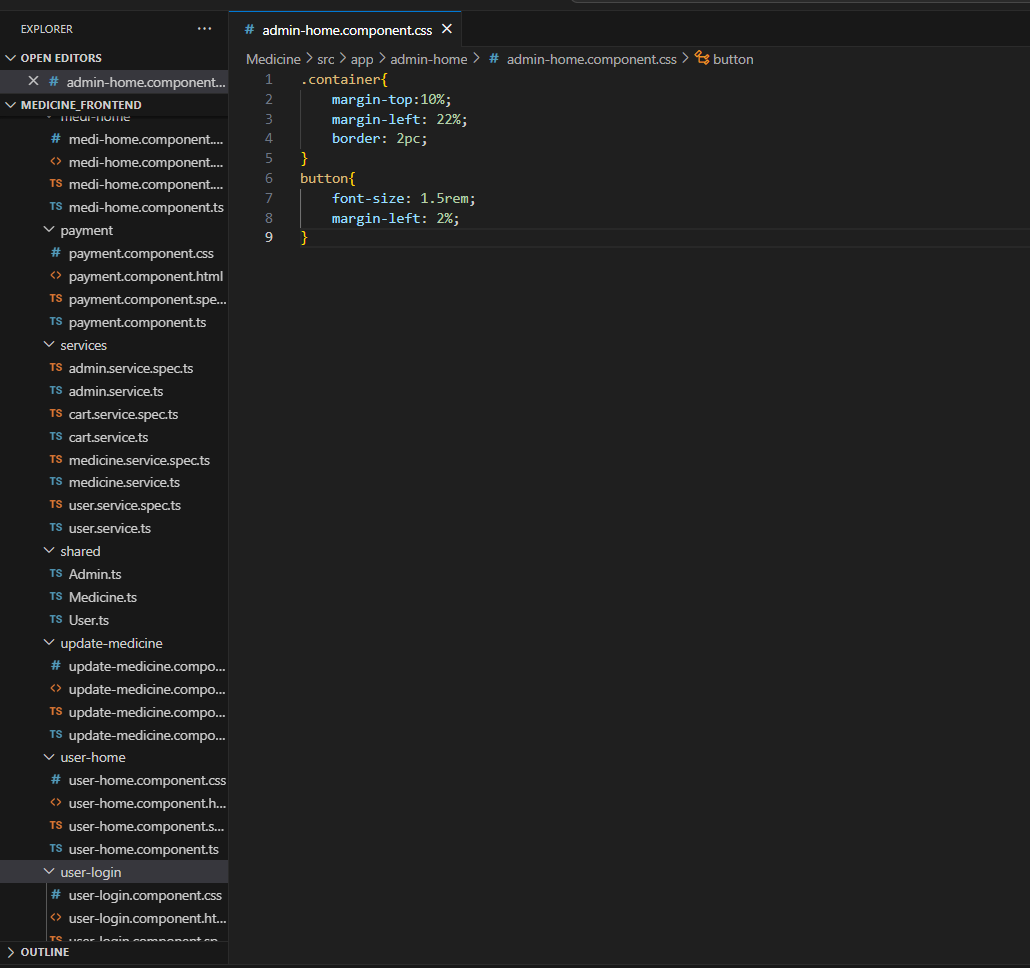
}

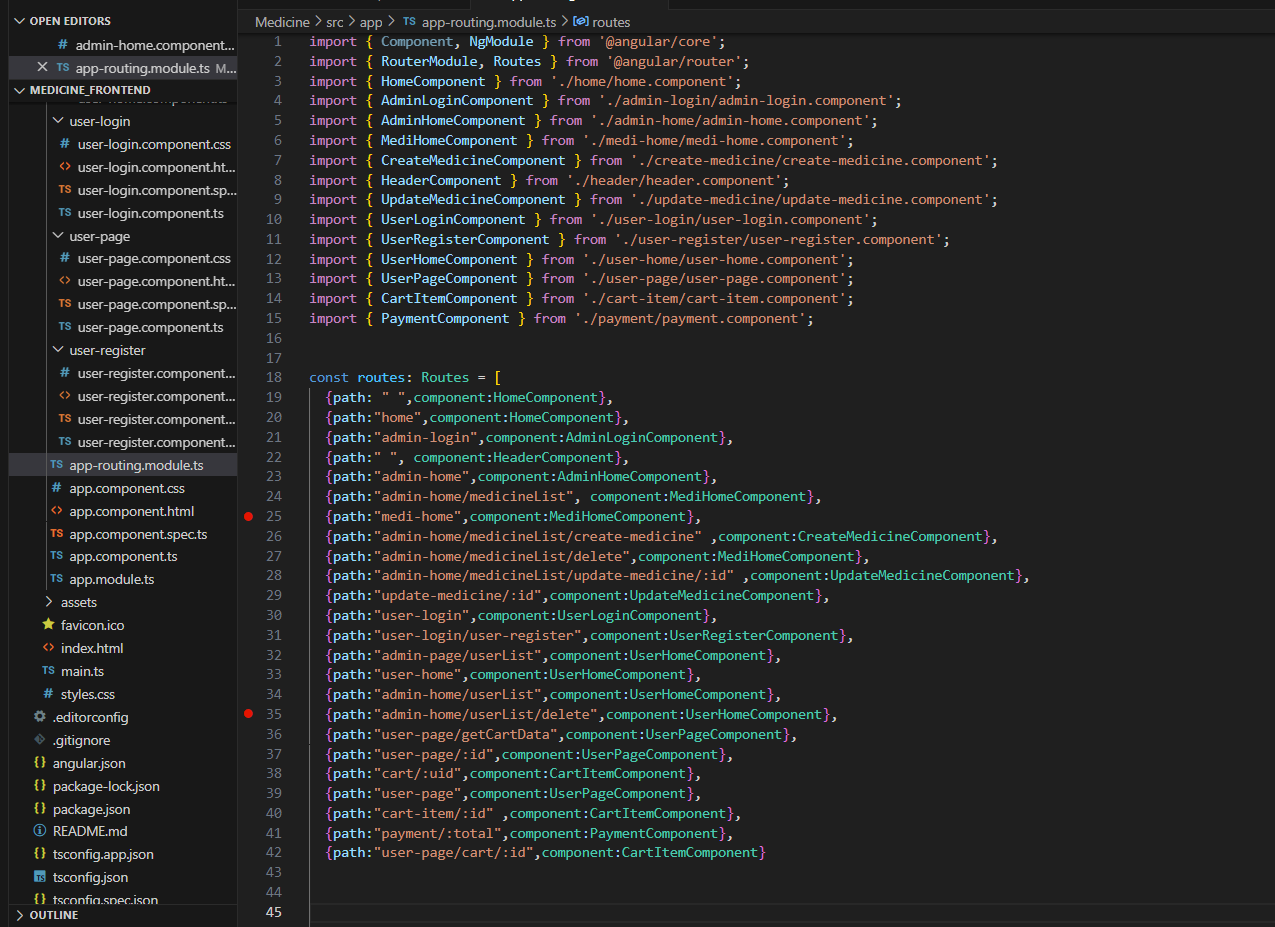
}

* **FrontEnd**

****

****

****

****