**SCD Lab 12**

**Name:** Hafsa Salman

**Roll no.** 22K-5161

UserController.java

//Hafsa Salman  
//22K-5161  
  
package com.lab\_12.lab\_12.Controller;  
  
import com.lab\_12.lab\_12.Model.Category;  
import com.lab\_12.lab\_12.Model.Item;  
import com.lab\_12.lab\_12.Model.User;  
import com.lab\_12.lab\_12.Repository.CategoryRepository;  
import com.lab\_12.lab\_12.Repository.ItemRepository;  
import com.lab\_12.lab\_12.Repository.UserRepository;  
import com.lab\_12.lab\_12.Service.UserService;  
import jakarta.servlet.http.Cookie;  
import jakarta.servlet.http.HttpServletResponse;  
import jakarta.servlet.http.HttpSession;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.\*;  
import java.util.List;  
  
@Controller  
@RequestMapping("/users")  
public class UserController  
{  
 @Autowired  
 private UserService userService;  
 @Autowired  
 private UserRepository userRepository;  
 @Autowired  
 private ItemRepository itemRepository;  
 @Autowired  
 private CategoryRepository categoryRepository;  
  
 @GetMapping("/createAccount")  
 public String createAccount(Model model)  
 {  
 model.addAttribute("user", new User());  
 return "createAccount";  
 }  
  
 @PostMapping("/createAccount")  
 public String createAccount(@ModelAttribute("user") User user, Model model)  
 {  
 if (user.getRole() == null || user.getRole().isEmpty())  
 {  
 user.setRole("user");  
 }  
  
 System.*out*.println("Role: " + user.getRole());  
  
 userRepository.save(user);  
 model.addAttribute("user", user);  
 return "redirect:/users/login";  
 }  
  
  
 @GetMapping("/login")  
 public String login()  
 {  
 return "login";  
 }  
 @PostMapping("/login")  
 public String loginPage(Model model, @RequestParam String email, @RequestParam String password, HttpSession session, HttpServletResponse response)  
 {  
 User user=userService.validateUser(email, password);  
  
 if(user!=null)  
 {  
 session.setAttribute("user", user);  
 Cookie cookie = new Cookie("user\_id", String.*valueOf*(user.getId()));  
 response.addCookie(cookie);  
  
 if (user.getRole().equals("super admin") || user.getRole().equals("admin"))  
 {  
 return "redirect:/users/SuperAndAdminDashboard";  
 }  
  
 return "redirect:/users/UserDashboard";  
 }  
  
 else  
 {  
 model.addAttribute("error", "Invalid email or password");  
 return "login";  
 }  
 }  
  
 @GetMapping("/SuperAndAdminDashboard")  
 public String SuperAndAdminDashboard(HttpSession session,Model model)  
 {  
 boolean superAdmin=false;  
 User user=(User)session.getAttribute("user");  
 if(user==null)  
 {  
 return "redirect:/users/login";  
 }  
  
 if(user.getRole().equals("super admin"))  
 {  
 superAdmin=true;  
 }  
  
 model.addAttribute("superAdmin", superAdmin);  
 model.addAttribute("user", user);  
 return "SuperAndAdminDashboard";  
 }  
  
 @GetMapping("/UserDashboard")  
 public String UserDashboard(HttpSession session,Model model)  
 {  
 User user=(User)session.getAttribute("user");  
 if(user==null)  
 {  
 return "redirect:/users/login";  
 }  
  
 model.addAttribute("user", user);  
  
 return "UserDashboard";  
 }  
  
 @GetMapping("/manageUsers")  
 public String manageUsers(HttpSession session,Model model)  
 {  
 User user=(User)session.getAttribute("user");  
 List<User> users=userRepository.findByRole("user");  
  
 if(user==null)  
 {  
 return "redirect:/users/login";  
 }  
  
 model.addAttribute("users", users);  
 model.addAttribute("user", user);  
 return "manageUsers";  
 }  
  
 @GetMapping("/manageAdmins")  
 public String manageAdmins(HttpSession session,Model model)  
 {  
 User user=(User)session.getAttribute("user");  
 List<User> admins=userRepository.findByRole("admin");  
  
 if(user==null)  
 {  
 return "redirect:/users/login";  
 }  
  
 model.addAttribute("admins", admins);  
 model.addAttribute("user", user);  
 return "manageAdmins";  
 }  
  
 @GetMapping("/editUser/{id}")  
 public String editUser(HttpSession session,Model model, @PathVariable ("id") int id)  
 {  
 boolean superAdmin=false;  
 User editUser=userRepository.findById(id).orElseThrow(()->new IllegalArgumentException("Invalid User Id:" + id));  
 User user=(User)session.getAttribute("user");  
  
 if(user==null)  
 {  
 return "redirect:/users/login";  
 }  
  
 if(user.getRole().equals("super admin"))  
 {  
 superAdmin=true;  
 }  
  
 model.addAttribute("user", user);  
 model.addAttribute("superAdmin", superAdmin);  
 model.addAttribute("editUser", editUser);  
 return "editUser";  
 }  
  
 @PostMapping("/editUser/{id}")  
 public String editUser(@ModelAttribute("editUser") User user, HttpSession session, Model model,@PathVariable ("id") int id)  
 {  
 user.setId(id);  
 userRepository.save(user);  
 System.*out*.println(user.getRole());  
 if(user.getRole().equals("admin"))  
 {  
 return "redirect:/users/manageAdmins";  
 }  
  
 else  
 {  
 return "redirect:/users/manageUsers";  
 }  
 }  
  
 @GetMapping("/manageRestaurant")  
 public String manageRestaurant(HttpSession session,Model model)  
 {  
 User user=(User)session.getAttribute("user");  
 List<Item> items=itemRepository.findAll();  
 List<Category> categories=categoryRepository.findAll();  
  
 if(user==null)  
 {  
 return "redirect:/users/login";  
 }  
 model.addAttribute("user", user);  
 model.addAttribute("items", items);  
 model.addAttribute("categories", categories);  
  
 return "manageRestaurant";  
 }  
}

Cateory.java

//Hafsa Salman  
//22K-5161  
  
package com.lab\_12.lab\_12.Model;  
  
import jakarta.persistence.\*;  
  
@Entity  
@Table(name = "category")  
public class Category  
{  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private int id;  
  
 @Column(nullable = false)  
 private String name;  
  
 public int getId()  
 {  
 return id;  
 }  
  
 public void setId(int id)  
 {  
 this.id = id;  
 }  
  
 public String getName()  
 {  
 return name;  
 }  
  
 public void setName(String name)  
 {  
 this.name = name;  
 }  
}

Item.java

//Hafsa Salman  
//22K-5161  
  
package com.lab\_12.lab\_12.Model;  
  
import jakarta.persistence.\*;  
  
@Entity  
@Table(name = "item")  
public class Item  
{  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private int id;  
  
 @Column(nullable = false)  
 private String name;  
  
 @ManyToOne  
 @JoinColumn(name = "category\_id", nullable = false)  
 private Category category;  
  
 @Column(nullable = false)  
 private int price;  
  
 public int getId()  
 {  
 return id;  
 }  
  
 public void setId(int id)  
 {  
 this.id = id;  
 }  
  
 public String getName()  
 {  
 return name;  
 }  
  
 public void setName(String name)  
 {  
 this.name = name;  
 }  
  
 public Category getCategory()  
 {  
 return category;  
 }  
  
 public void setCategory(Category category)  
 {  
 this.category = category;  
 }  
  
 public int getPrice()  
 {  
 return price;  
 }  
  
 public void setPrice(int price)  
 {  
 this.price = price;  
 }  
}

User.java

//Hafsa Salman  
//22K-5161  
  
package com.lab\_12.lab\_12.Model;  
  
import jakarta.persistence.\*;  
import java.sql.Timestamp;  
import java.time.LocalDateTime;  
  
  
@Entity  
@Table(name = "users")  
public class User  
{  
 public User()  
 {  
 this.created\_at = new Timestamp(System.*currentTimeMillis*());  
 }  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private int id;  
  
 @Column(nullable = false)  
 private String name;  
  
 @Column(nullable = false)  
 private String email;  
  
 @Column(nullable = false)  
 private String password;  
  
 @Column(nullable = false)  
 private String role;  
  
 @Column(nullable = false)  
 private Timestamp created\_at;  
  
 public int getId()  
 {  
 return id;  
 }  
  
 public void setId(int id)  
 {  
 this.id = id;  
 }  
  
 public String getName()  
 {  
 return name;  
 }  
  
 public void setName(String name)  
 {  
 this.name = name;  
 }  
  
 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;  
 }  
  
 public String getRole()  
 {  
 return role;  
 }  
  
 public void setRole(String role)  
 {  
 this.role = role;  
 }  
  
 public Timestamp getCreated\_at()  
 {  
 return created\_at;  
 }  
  
 public void setCreated\_at(Timestamp created\_at)  
 {  
 this.created\_at = created\_at;  
 }  
}

CategoryRepository.java

//Hafsa Salman  
//22K-5161  
  
package com.lab\_12.lab\_12.Repository;  
  
import com.lab\_12.lab\_12.Model.Category;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface CategoryRepository extends JpaRepository<Category, Integer>  
{  
}

ItemRepository.java

//Hafsa Salman  
//22K-5161  
  
package com.lab\_12.lab\_12.Repository;  
  
import com.lab\_12.lab\_12.Model.Item;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface ItemRepository extends JpaRepository<Item, Integer>  
{  
}

UserRepository.java

//Hafsa Salman  
//22K-5161  
  
package com.lab\_12.lab\_12.Repository;  
  
import com.lab\_12.lab\_12.Model.User;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
import java.util.List;  
import java.util.Optional;  
  
public interface UserRepository extends JpaRepository<User, Integer>  
{  
 Optional<User> findByEmail(String email);  
 List<User> findByRole(String role);  
}

UserService.java

//Hafsa Salman  
//22K-5161  
  
package com.lab\_12.lab\_12.Service;  
  
import com.lab\_12.lab\_12.Model.User;  
import com.lab\_12.lab\_12.Repository.UserRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import java.util.Optional;  
  
@Service  
public class UserService  
{  
 @Autowired  
 private UserRepository userRepository;  
  
 public User validateUser(String email, String password)  
 {  
 Optional<User> user = userRepository.findByEmail(email);  
 return user.filter(u -> u.getPassword().equals(password)).orElse(null);  
 }  
}

//createAccount.html

<!--  
Hafsa Salman  
22K-5161  
SCD Lab 12  
-->  
  
<!DOCTYPE html>  
<html xmlns:th="www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>22K-5161 (SCD Lab 12)</title>  
</head>  
<body>  
<h2>Create Account</h2>  
<form th:action="@{/users/createAccount}" th:object="${user}" method="post">  
 <label for="name">Name:</label>  
 <input type="text" id="name" th:field="\*{name}" placeholder="Enter Name" required><br><br>  
 <label for="email">Email:</label>  
 <input type="email" id="email" th:field="\*{email}" placeholder="Enter Email" required><br><br>  
 <label for="password">Password:</label>  
 <input type="password" id="password" th:field="\*{password}" placeholder="Enter Password" required><br><br>  
 <input type="hidden" th:field="\*{role}" th:value="'user'" />  
 <button type="submit">Create Account</button>  
</form>  
  
</body>  
</html>

//editUser.html

<!--  
Hafsa Salman  
22K-5161  
SCD Lab 12  
-->  
  
<!DOCTYPE html>  
<html xmlns:th="http://www.thymeleaf.org">  
<head>  
 <title>22K-5161 (SCD Lab 12)</title>  
  
</head>  
<body>  
<h2>Edit User</h2>  
<form th:action="@{/users/editUser/{id}(id=${editUser.id})}" th:object="${editUser}" method="post">  
 <label for="id">ID:</label>  
 <input type="text" id="id" th:field="\*{id}" readonly><br><br>  
 <label for="name">Name:</label>  
 <input type="text" id="name" th:field="\*{name}" required><br><br>  
 <label for="email">Email:</label>  
 <input type="email" id="email" th:field="\*{email}" required><br><br>  
 <label for="password">Password:</label>  
 <input type="password" id="password" th:field="\*{password}" required><br><br>  
 <div th:if="${superAdmin}">  
 <label for="role">Role:</label>  
 <select id="role" th:field="\*{role}">  
 <option value="user" th:selected="${editUser.role == 'user'}">User</option>  
 <option value="admin" th:selected="${editUser.role == 'admin'}">Admin</option>  
 </select>  
 <br><br>  
 </div>  
 <button type="submit">Save Changes</button>  
</form>  
  
</body>  
</html>

//login.html

<!--  
Hafsa Salman  
22K-5161  
SCD Lab 12  
-->  
  
<!DOCTYPE html>  
<html xmlns:th="www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>22K-5161 (SCD Lab 12)</title>  
</head>  
<body>  
<h2>Log In</h2>  
<form th:action="@{/users/login}" method="post">  
 <label>Email:</label>  
 <input type="email" name="email" required>  
 <br>  
 <label>Password:</label>  
 <input type="password" name="password" required>  
 <br>  
 <button type="submit">Login</button>  
</form>  
</body>  
</html>

//manageAdmins.html

<!--  
Hafsa Salman  
22K-5161  
SCD Lab 12  
-->  
  
<!DOCTYPE html>  
<html xmlns:th="www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>22K-5161 (SCD Lab 12)</title>  
</head>  
<body>  
<h2>Manage Admins</h2>  
<table border="1">  
 <thead>  
 <tr>  
 <th>ID</th>  
 <th>Name</th>  
 <th>Email</th>  
 <th>Password</th>  
 <th>Role</th>  
 <th>Created At</th>  
 <th>Actions</th>  
 </tr>  
 </thead>  
 <tbody>  
 <tr th:each="user : ${admins}">  
 <td th:text="${user.id}"></td>  
 <td th:text="${user.name}"></td>  
 <td th:text="${user.email}"></td>  
 <td th:text="${user.password}"></td>  
 <td th:text="${user.role}"></td>  
 <td th:text="${user.created\_at}"></td>  
 <td>  
 <a th:href="@{/users/editUser/{id}(id=${user.id})}">Edit</a> |  
 <a th:href="@{/users/deleteUser/{id}(id=${user.id})}" onclick="return confirm('Are you sure you want to delete this employee?')">Delete</a>  
 </td>  
 </tr>  
 </tbody>  
</table>  
</body>  
</html>

//manageRestaurant.html

<!--  
Hafsa Salman  
22K-5161  
SCD Lab 12  
-->  
  
<!DOCTYPE html>  
<html xmlns:th="www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>22K-5161 (SCD Lab 12)</title>  
</head>  
<body>  
<h2>Manage Restaurant Items</h2>  
  
<h2>Items</h2>  
<table border="1">  
 <thead>  
 <tr>  
 <th>ID</th>  
 <th>Name</th>  
 <th>Category ID</th>  
 <th>Price</th>  
 <th>Actions</th>  
 </tr>  
 </thead>  
 <tbody>  
 <tr th:each="item : ${items}">  
 <td th:text="${item.id}"></td>  
 <td th:text="${item.name}"></td>  
 <td th:text="${item.category.id}"></td>  
 <td th:text="${item.price}"></td>  
 <td>  
 <a th:href="@{/users/editItem/{id}(id=${item.id})}">Edit</a> |  
 <a th:href="@{/users/deleteItem/{id}(id=${item.id})}" onclick="return confirm('Are you sure you want to delete this employee?')">Delete</a>  
 </td>  
 </tr>  
 </tbody>  
</table>  
<a href="/users/addItem">Add New Item</a><br><br>  
  
<h2>Categories</h2>  
<table border="1">  
 <thead>  
 <tr>  
 <th>ID</th>  
 <th>Name</th>  
 <th>Actions</th>  
 </tr>  
 </thead>  
 <tbody>  
 <tr th:each="category : ${categories}">  
 <td th:text="${category.id}"></td>  
 <td th:text="${category.name}"></td>  
  
 <td>  
 <a th:href="@{/users/editCategory/{id}(id=${category.id})}">Edit</a> |  
 <a th:href="@{/users/deleteCategory/{id}(id=${category.id})}" onclick="return confirm('Are you sure you want to delete this employee?')">Delete</a>  
 </td>  
 </tr>  
 </tbody>  
</table>  
<a href="/users/addCategory">Add New Category</a><br><br>  
  
</body>  
</html>

//SuperAdminDashboard.html

<!--  
Hafsa Salman  
22K-5161  
SCD Lab 12  
-->  
  
<!DOCTYPE html>  
<html xmlns:th="www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>22K-5161 (SCD Lab 12)</title>  
</head>  
<body>  
<h2>Welcome to <span th:text="${user.role}">[User Role]</span> Dashboard</h2>  
  
<h3>Options : </h3>  
<ul>  
 <li th:if="${superAdmin}"><a href="/users/manageAdmins">Manage Admins</a></li>  
 <li><a href="/users/manageUsers">Manage Users</a></li>  
 <li><a href="/users/manageRestaurant">Restaurant Items</a></li>  
</ul>  
</body>  
</html>

//UserDashboard.html

<!--  
Hafsa Salman  
22K-5161  
SCD Lab 12  
-->  
  
<!DOCTYPE html>  
<html xmlns:th="www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>22K-5161 (SCD Lab 12)</title>  
</head>  
<body>  
<h2>User Dashboard</h2>  
<h2> Welcome <span th:text="${user.name}">[User Name]</span> </h2>  
<p><strong>Email:</strong> <span th:text="${user.email}">[User Email]</span></p>  
<p><strong>User ID:</strong> <span th:text="${user.id}">[User ID]</span></p>  
</body>  
</html>

//manageUsers.html

<!--  
Hafsa Salman  
22K-5161  
SCD Lab 12  
-->  
  
<!DOCTYPE html>  
<html xmlns:th="www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>22K-5161 (SCD Lab 12)</title>  
</head>  
<body>  
<h2>Manage users</h2>  
<table border="1">  
 <thead>  
 <tr>  
 <th>ID</th>  
 <th>Name</th>  
 <th>Email</th>  
 <th>Password</th>  
 <th>Role</th>  
 <th>Created At</th>  
 <th>Actions</th>  
 </tr>  
 </thead>  
 <tbody>  
 <tr th:each="user : ${users}">  
 <td th:text="${user.id}"></td>  
 <td th:text="${user.name}"></td>  
 <td th:text="${user.email}"></td>  
 <td th:text="${user.password}"></td>  
 <td th:text="${user.role}"></td>  
 <td th:text="${user.created\_at}"></td>  
 <td>  
 <a th:href="@{/users/editUser/{id}(id=${user.id})}">Edit</a> |  
 <a th:href="@{/users/deleteUser/{id}(id=${user.id})}" onclick="return confirm('Are you sure you want to delete this employee?')">Delete</a>  
 </td>  
 </tr>  
 </tbody>  
</table>  
</body>  
</html>