

SPORTY SHOES CODE

Source_Code:

Open Pom.xml add Dependencies

```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-
4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.5.1</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.project</groupId>
    <artifactId>SportyShoes</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>SportyShoes</name>
    <description>phase3 project for Spring Boot</description>
    <properties>
        <java.version>11</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jdbc</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-thymeleaf</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-webflux</artifactId>
        </dependency>
        <dependency>
            <groupId>org.eclipse.persistence</groupId>
            <artifactId>javax.persistence</artifactId>
            <version>2.1.0</version>
        </dependency>
    </dependencies>
</project>
```

```

    <scope>compile</scope>
  </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-devtools</artifactId>
      <scope>runtime</scope>
      <optional>true</optional>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>io.projectreactor</groupId>
      <artifactId>reactor-test</artifactId>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-batch</artifactId>
    </dependency>

    <dependency>
      <groupId>org.projectlombok</groupId>
      <artifactId>lombok</artifactId>
      <optional>true</optional>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web-services</artifactId>
    </dependency>
    <dependency>
      <groupId>com.h2database</groupId>
      <artifactId>h2</artifactId>
      <scope>runtime</scope>
    </dependency>
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <scope>runtime</scope>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>

```

```
        </plugin>
    </plugins>
</build>

</project>
```

Src/main/java

By default Spring creates package com.project.SportyShoes

Create GlobalExceptionHandler.java

```
package com.project.SportyShoes;

import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;

@ControllerAdvice

public class GlobalExceptionHandler {

    @ExceptionHandler(Exception.class)

    public String handleException1(Exception e) {

        e.printStackTrace();

        return "Some error has occurred, please contact to administrator";

    }

}
```

Create SportyShoesApplication.java

```
package com.project.SportyShoes;

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

@SpringBootApplication

public class SportyShoesApplication {
```

```

        public static void main(String[] args) {

            SpringApplication.run(SportyShoesApplication.class, args);

        }

    }

```

Create package com.project.SportyShoes.controller

Create CategoriesController.java

```

package com.project.SportyShoes.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

import com.project.SportyShoes.entity.Categories;
import com.project.SportyShoes.service.CategoriesService;

@Controller

public class CategoriesController {

    @Autowired

    private CategoriesService catserv;

    @RequestMapping("/listcat")

    public String viewListPage(Model model)

    {

```

```

        List<Categories> listCategories= catserv.listAll();

        model.addAttribute("listCategories", listCategories);

        return "manage";
    }

    @RequestMapping("/new")
    public String addNewCatPage(Model model)
    {

        Categories categories= new Categories();

        model.addAttribute("categories", categories);

        return "new_category";
    }

    @RequestMapping(value = "/save", method = RequestMethod.POST)
    public String saveCategory(@ModelAttribute("categories") Categories categories, Model m)
    {

        catserv.save(categories);

        List<Categories> listCategories= catserv.listAll();

        m.addAttribute("listCategories", listCategories);

        return "manage";
    }

    @RequestMapping("/delete/{id}")
    public String deleteCategory(@PathVariable(name ="id") long id, Model m)
    {

        catserv.delete(id);

        List<Categories> listCategories= catserv.listAll();

        m.addAttribute("listCategories", listCategories);

        return "manage";
    }
}

```

Create LoginController.java

```
package com.project.SportyShoes.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;
import com.project.SportyShoes.entity.Login;
import com.project.SportyShoes.service.LoginService;

@Controller

public class LoginController {

    @Autowired

    private LoginService ls;

    @GetMapping("/")

    public ModelAndView login() {

        ModelAndView mav = new ModelAndView("login");

        mav.addObject("user", new Login());

        return mav;

    }

    @PostMapping("/adminlogin")

    public String signUp(@ModelAttribute("user") Login user ) {

        Login oauthUser = null;

        System.out.println("from login"+ user);

        if(user.getPassword() != "" ) {

            System.out.println("logging in");

            oauthUser = ls.signgin(user.getUsername(), user.getPassword());
```

```

        return "menu";
    } else {
        if (user.getNewPassword() != "" ) {
            System.out.println("Changing new password");
            oauthUser = ls.changePassword(user.getUsername(), user.getNewPassword());
            if(oauthUser == null ) {
                return "invalid_credentials";
            }
            return "menu";
        }
        return "invalid_credentials";
    }
}

//@GetMapping("/forgotPassword")
//public String update(@ModelAttribute("user") Login user){
//    //Login changepw = userService.updatePassword(user.);
//    //System.out.println(changepw);
//    //if(Objects.nonNull(changepw))
//    //{
//        //return "redirect:/";
//    //} else {
//        //return "redirect:/login";
//    //}
//}

//{
//}

//}

@RequestMapping("/logout")
public String logoutDo()

```

```

    {

        return "conform_logout";

    }

    //PostMapping("/reset")

    //public String resetPassword( @ModelAttribute("userchange") Login user) {

        //System.out.println("form reset");

        //Boolean changePassword = ls.changePassword(user.getUsername(),
user.getPassword());

        //if(!changePassword) {

            //        return "usernotfound";

            //}

            //return "menu";

        //}

    /*

    @GetMapping("/forgotpw")

    public String resetPassword1( @ModelAttribute("change") Login change ) {

        System.out.println("from login"+ change);

        Boolean oauthUser = ls.changePassword(change.getUsername(), change.getPassword());

        System.out.print(oauthUser);

        if(Objects.nonNull(oauthUser))

        {

            return "menu";

        } else {

            return "invalid_credentials";

        }

    }

    */

}

```

Create ProductController.java


```

package com.project.SportyShoes.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

import com.project.SportyShoes.entity.Categories;
import com.project.SportyShoes.entity.Product;
import com.project.SportyShoes.service.ProductService;

@Controller
public class ProductController {
    @Autowired
    public ProductService prodserv;

    @RequestMapping("/listprod")
    public String viewListProductPage(Model model)
    {
        List<Product> listProduct= prodserv.listAll();
        model.addAttribute("listProduct", listProduct);
        return "product_manage";
    }

    @RequestMapping("/newproduct")
    public String addNewProductPage(Model model)
    {
        Product product= new Product();
        model.addAttribute("product", product);

        return "new_product";
    }

    @RequestMapping(value = "/saveproduct", method = RequestMethod.POST)
    public String saveCategory(@ModelAttribute("product") Product product, Model md)
    {
        prodserv.save(product);
        List<Product> listProduct= prodserv.listAll();
        md.addAttribute("listProduct", listProduct);

        return "product_manage";
    }
}

```

Create UserDetailsController.java

```
package com.project.SportyShoes.controller;

import java.util.ArrayList;

import java.util.List;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

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

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

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

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

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

import org.springframework.web.servlet.ModelAndView;

import com.project.SportyShoes.entity.UserDetails;

import com.project.SportyShoes.service.UserDetailsService;
```

@Controller

```
public class UserDetailsController {

    @Autowired

    UserDetailsService userservice;

    //@GetMapping("/list")

    //public List<UserDetails> getUserDetails()

    //{

        //List<UserDetails> listuser = new ArrayList<UserDetails>();

        //listuser= userservice.getUsers();

        // return listuser;

    //}
```

```

    @GetMapping("/index")

    public String viewUserDetails(Model model)
    {

        List<UserDetails> listUsers= userservice.listAll();

        System.out.println("listUsers" + listUsers);

        model.addAttribute("listUsers",listUsers);

        return "/index";

    }

//    @GetMapping("/find")
//
//    public List<UserDetails> findusername(@RequestParam("username") String username)
//
//    {
//
//        List<UserDetails> searchuser =new ArrayList<UserDetails>();
//
//        searchuser = userservice.get(username);
//
//        System.out.println("user"+ searchuser);
//
//        return searchuser;
//
//    }

    @GetMapping("/user/{username}")

    public String findUserName(@PathVariable(name="username")String username)

    {

        System.out.println("i am here");

        ModelAndView mav = new ModelAndView("find");

        List<UserDetails> userdetails=userservice.get(username);

        mav.addObject("userdetails", userdetails);

        return "finduser";

    }

}

```

Create package com.project.SportyShoes.entity

Create BaseEntity.java

```
package com.project.SportyShoes.entity;

import lombok.Data;

import lombok.EqualsAndHashCode;

import javax.persistence.Column;

import javax.persistence.EntityListeners;

import javax.persistence.MappedSuperclass;

import java.util.Date;

@MappedSuperclass

@Data

@EqualsAndHashCode

@EntityListeners(EntityListener.class)

public abstract class BaseEntity {

    @Column(nullable = false, updatable = false)

    private Date createDate;

    @Column(insertable = false)

    private Date lastUpdateDate;

    public Date getCreateDate() {

        return createDate;

    }

    public void setCreateDate(Date createDate) {

        this.createDate = createDate;

    }

    public Date getLastUpdateDate() {

        return lastUpdateDate;

    }

    public void setLastUpdateDate(Date lastUpdateDate) {

        this.lastUpdateDate = lastUpdateDate;

    }

}
```

```
}
```

Create Categories.java

```
package com.project.SportyShoes.entity;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="categories")

public class Categories {

    @Id

    @GeneratedValue(strategy=GenerationType.IDENTITY)

    private long id;

    private String categoryName;


    public long getId() {

        return id;

    }

    public void setId(long id) {

        this.id = id;

    }

    public Categories() {

        super();

        // TODO Auto-generated constructor stub

    }

    public Categories(long id, String categoryName) {

        super();
```

```

        this.id = id;

        this.categoryName = categoryName;
    }

    @Override
    public String toString() {
        return "Categories [id=" + id + ", categoryName=" + categoryName + "]";
    }

    public String getCategoryName() {
        return categoryName;
    }

    public void setCategoryName(String categoryName) {
        this.categoryName = categoryName;
    }

    /*public String women;

    public String men;

    public String kids;

    public Categories() {
        super();

        // TODO Auto-generated constructor stub
    }

    public Categories(long id, String women, String men, String kids) {
        super();

        this.id = id;

        this.women = women;

        this.men = men;

        this.kids = kids;
    }

```

```
public long getId() {  
    return id;  
}  
  
public void setId(long id) {  
    this.id = id;  
}  
  
public String getWomen() {  
    return women;  
}  
  
public void setWomen(String women) {  
    this.women = women;  
}  
  
public String getMen() {  
    return men;  
}  
  
public void setMen(String men) {  
    this.men = men;  
}  
  
public String getKids() {  
    return kids;  
}  
  
public void setKids(String kids) {  
    this.kids = kids;  
}  
  
@Override  
public String toString() {  
    return "Categories [id=" + id + ", women=" + women + ", men=" + men + ", kids=" +  
kids + "];";  
}
```

```
    }*/  
  
}
```

Create EntityListener.java

```
package com.project.SportyShoes.entity;  
  
import javax.persistence.PrePersist;  
import javax.persistence.PreUpdate;  
  
import java.util.Date;  
  
public class EntityListener {  
    @PrePersist  
    public void prePersist(BaseEntity entity) {  
        Date now = new Date();  
        entity.setCreateDate(now);  
    }  
  
    @PreUpdate  
    public void preUpdate(BaseEntity entity) {  
        Date now = new Date();  
        entity.setLastUpdateDate(now);  
    }  
}
```

Create Login.java

```
package com.project.SportyShoes.entity;
```



```
import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;


@Entity

@Table(name="login")

public class Login {

    @Id

    @GeneratedValue(strategy= GenerationType.IDENTITY)

    private Long id;

    private String username;

    private String password;

    private String newPassword;


    public String getNewPassword() {

        return newPassword;

    }

    public void setNewPassword(String newPassword) {

        this.newPassword = newPassword;

    }

    public Login(Long id, String username, String password, String newPassword) {

        super();

        this.id = id;

        this.username = username;

        this.password = password;
```

```
        this.newPassword = newPassword;
    }

    public Login() {
        super();

        // TODO Auto-generated constructor stub
    }

    public Login(Long id, String username, String password) {
        super();

        this.id = id;

        this.username = username;

        this.password = password;
    }

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
```

```

        this.password = password;
    }

    @Override

    public String toString() {

        return "Login [id=" + id + ", username=" + username + ", password=" + password + ",
newPassword=" + newPassword+ "]";

    }

}

```

Create Product.java

```

package com.project.SportyShoes.entity;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="product")

public class Product {

    @Id

    @GeneratedValue(strategy=GenerationType.IDENTITY)

    private long id;

    private String productName;

    private String category;

    private long price;

    private Boolean status;

    private String size;

    public Product() {

```

```
        super();

        // TODO Auto-generated constructor stub
    }

    public long getId() {

        return id;

    }

    public void setId(long id) {

        this.id = id;

    }

    public String getProductName() {

        return productName;

    }

    public void setProductName(String productName) {

        this.productName = productName;

    }

    public String getCategory() {

        return category;

    }

    public void setCategory(String category) {

        this.category = category;

    }

    public long getPrice() {

        return price;

    }

    public void setPrice(long price) {

        this.price = price;

    }

    public Boolean getStatus() {
```

```

        return status;
    }

    public void setStatus(Boolean status) {

        this.status = status;

    }

    public String getSize() {

        return size;

    }

    public void setSize(String size) {

        this.size = size;

    }

    @Override

    public String toString() {

return "Product [id=" + id + ", productName=" + productName + ", category=" + category + ", price=" +
price+ ", status=" + status + ", size=" + size + "]";

    }

    public Product(long id, String productName, String category, long price, Boolean status, String
size) {

        super();

        this.id = id;

        this.productName = productName;

        this.category = category;

        this.price = price;

        this.status = status;

        this.size = size;

    }

}

```

Create UserDetails.java

```
package com.project.SportyShoes.entity;
```

```
import java.sql.Date;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.GenerationType;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.Table;
```

```
@Entity(name = "userdetails")
```

```
@Table(name="userdetails")
```

```
public class UserDetails extends BaseEntity {
```

```
    @Id
```

```
    @GeneratedValue(strategy= GenerationType.IDENTITY)
```

```
    public long userId;
```

```
    public String username;
```

```
    public String emailId;
```

```
    public String phone;
```

```
    public String productPerchased;
```

```
    public Date purchaseDate;
```

```
    public String category;
```

```
    public UserDetails() {
```

```
        super();
```

```
        // TODO Auto-generated constructor stub
```

```
    }
```

```
    public UserDetails(long userId, String username, String emailId, String phone, String  
productPerchased,
```

```
        Date purchaseDate, String category) {

    super();

    this.userId = userId;

    this.username = username;

    this.emailId = emailId;

    this.phone = phone;

    this.productPerchased = productPerchased;

    this.purchaseDate = purchaseDate;

    this.category = category;

}

public long getUserId() {

    return userId;

}

public void setUserId(long userId) {

    this.userId = userId;

}

public String getUsername() {

    return username;

}

public void setUsername(String username) {

    this.username = username;

}

public String getEmailId() {

    return emailId;

}

public void setEmailId(String emailId) {

    this.emailId = emailId;

}
```

```
public String getPhone() {  
    return phone;  
}  
  
public void setPhone(String phone) {  
    this.phone = phone;  
}  
  
public String getProductPerchased() {  
    return productPerchased;  
}  
  
public void setProductPerchased(String productPerchased) {  
    this.productPerchased = productPerchased;  
}  
  
public Date getPerchaseDate() {  
    return perchaseDate;  
}  
  
public void setPerchaseDate(Date perchaseDate) {  
    this.perchaseDate = perchaseDate;  
}  
  
public String getCategory() {  
    return category;  
}  
  
public void setCategory(String category) {  
    this.category = category;  
}  
  
@Override  
public String toString() {  
    return "UserDetails [userId=" + userId + ", username=" + username + ", emailId=" +  
emailId + ", phone=" + phone
```



```

        + ", productPerchased=" + productPerchased + ", purchaseDate=" +
purchaseDate + ", category=" + category
        + "]";
    }
}

```

Create package com.project.SportyShoes.jdbc

Create CategoriesRepository.java

```

package com.project.SportyShoes.jdbc;

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

import com.project.SportyShoes.entity.Categories;

public interface CategoriesRepository extends JpaRepository<Categories, Long> {

}

```

Create LoginRepository.java

```

package com.project.SportyShoes.jdbc;

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

import org.springframework.data.jpa.repository.Modifying;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Repository;

import com.project.SportyShoes.entity.Login;

import lombok.val;

@Repository

public interface LoginRepository extends JpaRepository<Login, Long> {

    Login findByUsernameAndPassword(String username, String password);
}

```

```

        Login findByUsername(String username);

        //Login findByEmail(String emailId);

        //Login findByResetPassword(String resetPassword);

        @Modifying

        @Query("update Login log set log.password = :password where log.id = :id")

        void changePassword(@Param(value="password") String password, @Param(value = "id") Long id);

    }

```

Create ProductRepository.java

```

package com.project.SportyShoes.jdbc;

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

import com.project.SportyShoes.entity.Product;

public interface ProductRepository extends JpaRepository<Product, Long>{

}

```

Create UserDetailsRepository.java

```

package com.project.SportyShoes.jdbc;

import java.util.List;

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

import org.springframework.stereotype.Repository;

import com.project.SportyShoes.entity.UserDetails;

@Repository

public interface UserDetailsRepository extends JpaRepository<UserDetails, Long> {

    public List<UserDetails> findByusername(String username);

}

```

Create package com.project.SportyShoes.service

Create CategoriesService.java

```

package com.project.SportyShoes.service;

import java.util.List;

```

```
import javax.transaction.Transactional;

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

import org.springframework.stereotype.Service;

import com.project.SportyShoes.entity.Categories;

import com.project.SportyShoes.jdbc.CategoriesRepository;
```

```
@Service
```

```
@Transactional
```

```
public class CategoriesService {

    @Autowired

    private CategoriesRepository catrepo;

    public List<Categories> listAll()

    {

        return catrepo.findAll();

    }

    public void save(Categories categories)

    {

        catrepo.save(categories);

    }

    public Categories get(long id)

    {

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

    }

    public void delete(long id)

    {

        catrepo.deleteById(id);

    }

}
```

Create LoginService.java

```
package com.project.SportyShoes.service;

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

import com.project.SportyShoes.entity.Login;
import com.project.SportyShoes.entity.UserDetails;
import com.project.SportyShoes.jdbc.LoginRepository;

@Service

public class LoginService{

    @Autowired

    private LoginRepository repo;

    public Login signin(String username, String password) {

        Login user = repo.findByUsernameAndPassword(username, password);

        System.out.println("from service" + user);

        return user;

    }

    public Login changePassword(String username, String newpassword) {

        // TODO Auto-generated method stub

        Login uplogin = null;

        Login user1 = repo.findByUsername(username);

        if(user1 !=null ) {

            System.out.println("user1" + user1);
```

```

        repo.changePassword(newpassword, user1.getId());

        uplogin = repo.findByUsername(username);

        System.out.println("Password Changed");

        return uplogin;
    }

    return null;
}

}

// Login login=repo.findByEmail(email);

// if(login != null)

// {

//     login.setResetPassword(resetPassword);

//     repo.save(login);

// }

// else {

//     System.out.println("couldnot find the emailid");

// }

// }

// catch (Exception e) {

//     e.printStackTrace(e.getMessage());

// }

// }

// public Login getByResetPassword(String resetPassword)

// {

//     return repo.findByResetPassword(resetPassword);

// }

// }

```

Create ProductService.java

```
package com.project.SportyShoes.service;

import java.util.List;

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

import com.project.SportyShoes.entity.Categories;
import com.project.SportyShoes.entity.Product;
import com.project.SportyShoes.jdbc.ProductRepository;
```

```
@Service
```

```
public class ProductService {

    @Autowired

    public ProductRepository prodrepo;

    public List<Product> listAll()
    {

        return prodrepo.findAll();

    }

    public void save(Product product)
    {

        prodrepo.save(product);

    }

}
```

Create UserDetailsService.java

```
package com.project.SportyShoes.service;
```

```

import java.util.ArrayList;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.project.SportyShoes.entity.UserDetails;

import com.project.SportyShoes.jdbc.UserDetailsRepository;

@Service

public class UserDetailsService {

    @Autowired

    UserDetailsRepository userRepo;

    //public List<UserDetails> listAll(){

        //List<UserDetails> ud=new ArrayList<UserDetails>();

        //ud= userRepo.findAll();

        //System.out.println("userdetails "+ud);

        //return ud;

    //}

    public List<UserDetails> listAll()

    {

        return userRepo.findAll();

    }

    public List<UserDetails> get(String username) {

        return userRepo.findByusername(username);

    }

    /*public List<UserDetails> searchUsers(String username) {

        List<UserDetails> su= new ArrayList<UserDetails>();

```

```

        su= userRepo.findByusername(username);

        return su;

    }*/

}

```

Src/main/resources

Open application.properties

```

spring.datasource.url=jdbc:mysql://localhost:3306/sairam213
spring.datasource.username=root
spring.datasource.password=password
#auto create tables fromspring boot
spring.jpa.generate-ddl=true
spring.jpa.hibernate.ddl-auto = update
server.port = 8080
#logging.level.root=WARN
logging.level.org.springframework.web: DEBUG
spring.jpa.open-in-view=true
spring.thymeleaf.cache=true
spring.thymeleaf.prefix=classpath:/templates/
spring.thymeleaf.suffix=.html
spring.jpa.show-sql=true
api.base.path = http://localhost:/mywork

```

Create Folder templates and

Create conform_logout.html.

```

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<a href="/">Conform Logout</a>

</body>
</html>

```


Create finduser.html

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<div align="center">
  <H3>Find User By UserName</H3>
  <br>
  <form action="#" th:action="@{/find{username}}" th:object="${userdetails}" method="post">
    <table border="0" cellpadding="10">
      <tr>
        <td>UserName:</td>
        <td><input type="text" th:field="*{username}"/>
        </td>
      </tr>
      <tr>
        <td colspan="2"><button type="submit">Search</button> </td>
      </tr>
    </table>
  </form>
</div>
</body>
</html>
```

Create Index.html

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="ISO-8859-1">
<title>List OF User Details</title>
</head>
<body>
<div align="center">
  <H1>List of All The Users</H1>
  <br>
  <table border="1" cellpadding="10">
    <thead>
      <tr>
        <th>UserId</th>
        <th>UserName</th>
        <th>email Id</th>
        <th>Phone</th>
        <th>Product Purchased</th>
        <th>Purchase Date</th>
        <th>Category</th>
        <th>Create Date</th>
        <th>Last Update Date</th>
      </tr>
    </thead>
```

```

<tbody>
  <tr th:each="userdetails : ${listUsers}">
    <td th:text="${userdetails.userId}">UserId</td>
    <td th:text="${userdetails.username}">UserName</td>
    <td th:text="${userdetails.emailId}">email Id</td>
    <td th:text="${userdetails.phone}">Phone</td>
    <td th:text="${userdetails.productPerchased}">Product Perchased</td>
    <td th:text="${userdetails.purchaseDate}">Perchase Date</td>
    <td th:text="${userdetails.category}">Category</td>
    <td th:text="${userdetails.createDate}">Create Date</td>
    <td th:text="${userdetails.lastUpdateDate}">Last Update Date</td>
  </tr>
</tbody>
</table>
<a href="/logout">LogOut</a>
</div>
</body>
</html>

```

Create invalid_credentials.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1>Invalid Credentials</h1><br>
<br>
<a href="/">try Again</a>

</body>
</html>

```

Create login.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<div align="center">
<h2>Wellcome To Sporty Shoes Web App!!!</h2>

<table width="100%" height="100%" border="0" cellpadding="0" align="center">
<tr>
<td align="center" valign="middle">
<table class="table-bordered" width="350" border="0" cellpadding="3" cellspacing="3" bgcolor="#ffffff">
<form action="#" th:action="@{/adminlogin}" th:object="${user}" method="post">
<tr>
<td height="25" colspan="2" align="left" valign="middle" bgcolor="#ffffff" class="style2">
<div align="center">

```

```

<strong>Admin Login</strong>
</div>
</td>
</tr>
<tr>
<td>Username</td>
<td><input type="text" th:field="*{username}"
class="form-control" /></td>
</tr>
<tr>
<td>Password</td>
<td><input type="password" th:field="*{password}"
class="form-control" /></td>
</tr>
<tr>
<td>New Password</td>
<td><input type="password" th:field="*{newPassword}"
class="form-control" /></td>
</tr>
<tr>
<td colspan="2"><button type="submit" class="btn btn-
info" align="right">login/change password</button></td>
</tr>
</form>
</table>
</td>
</tr>
</table>
</body>
</html>

```

Create manage.html

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="ISO-8859-1">
<title>Manage Category Unit</title>
</head>
<body>
<div align="center">
<h1>Categories List</h1><br>
<div align="center">
<a href="/new">Create New Category</a>
</div>
<table border="1" cellpadding="10"><thead><tr>
<th>id</th>
<th>Name</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr th:each="categories : ${listCategories}">
<td th:text="${categories.id}">id</td>
<td th:text="${categories.categoryName}">Name</td>
<td><a th:href="@{/delete/{id}(id=${categories.id})}">Delete</a></td>

```

```

</tr>
</tbody>
</table>
<a href="/logout">LogOut</a>
</div>
</body>
</html>

```

Create menu.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1>Admin Work System</h1>
<a href="/listcat">Click here to Manage Categories</a><br>
<a href="/index">Click here to view the list of users</a><br>
<a href="/listprod">Click here to view the products List</a><br>
<br>
<br>
<a href="/logout">LogOut</a>
</body>
</html>

```

Create new_category.html

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
__xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="ISO-8859-1">
<title>Create New Category</title>
</head>
<body>
<div align="center">
<h1>Create New Category</h1>
<br/>
<form action="#" th:action="@{/save}" th:object="${categories}"
method="post">

<table border="0" cellpadding="10">
<tr>
<td>Name:</td>
<td><input type="text" th:field="*{categoryName}" /></td>
</tr>
<tr>
<td colspan="2"><button type="submit">Save</button> </td>
</tr>
</table>
</form>
<a href="/logout">LogOut</a>
</div>

```

```
</body>
</html>
```

Create new_product.html

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="ISO-8859-1">
<title>Create New Category</title>
</head>
<body>
<div align="center">
<h1>Create New Product</h1>
<br/>
<form action="#" th:action="@{/saveproduct}" th:object="${product}"
      method="post">

<table border="0" cellpadding="10">
<tr>
<td>Product Name:</td>
<td><input type="text" th:field="*{productName}" /></td>
</tr>
<tr>
<td>Category:</td>
<td><input type="text" th:field="*{category}" /></td>
</tr>
<tr>
<td>Price:</td>
<td><input type="text" th:field="*{price}" /></td>
</tr>
<tr>
<td>Status:</td>
<td><input type="text" th:field="*{status}" /></td>
</tr>
<tr>
<td>size:</td>
<td><input type="text" th:field="*{size}" /></td>
</tr>
<tr>
<td colspan="2"><button type="submit">Save</button> </td>
</tr>
</table>
</form>

</div>
</body>
</html>
```

Create product_manage.html

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="ISO-8859-1">
<title>Manage Product Unit</title>
</head>
<body>
<div align="center">
<h1>Products List</h1><br>
<h2><b><a href="/newproduct">Create New ProductList</a></b></h2>
<table border="1" cellpadding="10">
<thead>
<tr>
<th>Product id</th>
<th>Product Name</th>
<th>Category</th>
<th>Price</th>
<th>Status</th>
<th>size</th>
</tr>
</thead>
<tbody>
<tr th:each="product : ${listProduct}">
<td th:text="${product.id}">Product id</td>
<td th:text="${product.productName}">Product Name</td>
<td th:text="${product.category}">Category</td>
<td th:text="${product.price}">Price</td>
<td th:text="${product.status}">Status</td>
<td th:text="${product.size}">size</td>
</tr></tbody>
</table>
<a href="/logout">LogOut</a>
</div>
</body>
</html>

```

Create relogin.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<div align="center">
<h2>Change Password!!!</h2>
<table width="100%" height="100%" border="0" cellpadding="0" align="center">
<tr>
<td align="center" valign="middle">
<table class="table-bordered" width="350" border="0" cellpadding="3" cellspacing="3" bgcolor="#ffffff">
<form action="#" th:action="@{/}" th:object="${userchange}" method="post">
<tr>
<td height="25" colspan="2" align="left" valign="middle" bgcolor="#ffffff" class="style2">
<div align="center">

```

```

<strong>Change Password</strong>
</div>
</td>
</tr>
<tr>
<td>Username</td>
<td><input type="text" th:field="*{username}" name="username" /></td>
</tr>
<tr>
<td>New Password</td>
<td><input type="password" th:field="*{password}" name="password" /></td>
</tr>
<tr>
<td colspan="2"><button type="submit" class="btn btn-info" align="right">Change
Password</button></td>
</tr>
</form>
</table>
</td>
</tr>
</table>
</body>
</html>

```

Create usernotfound.html

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="ISO-8859-1">
<title>User Not Found</title>
</head>
<body>
<h1>User not found !</h1>
</body>
</html>

```