Implement Frameworks the DevOps way

Developed by: Swati Jajoriya

Email: swatijajoriya@gmail.com

Github link- https://github.com/SwatiJajoriya/SportyShoe.git

Project: Make an E-commerce Website for Sporty Shoes - sportyshoes.com

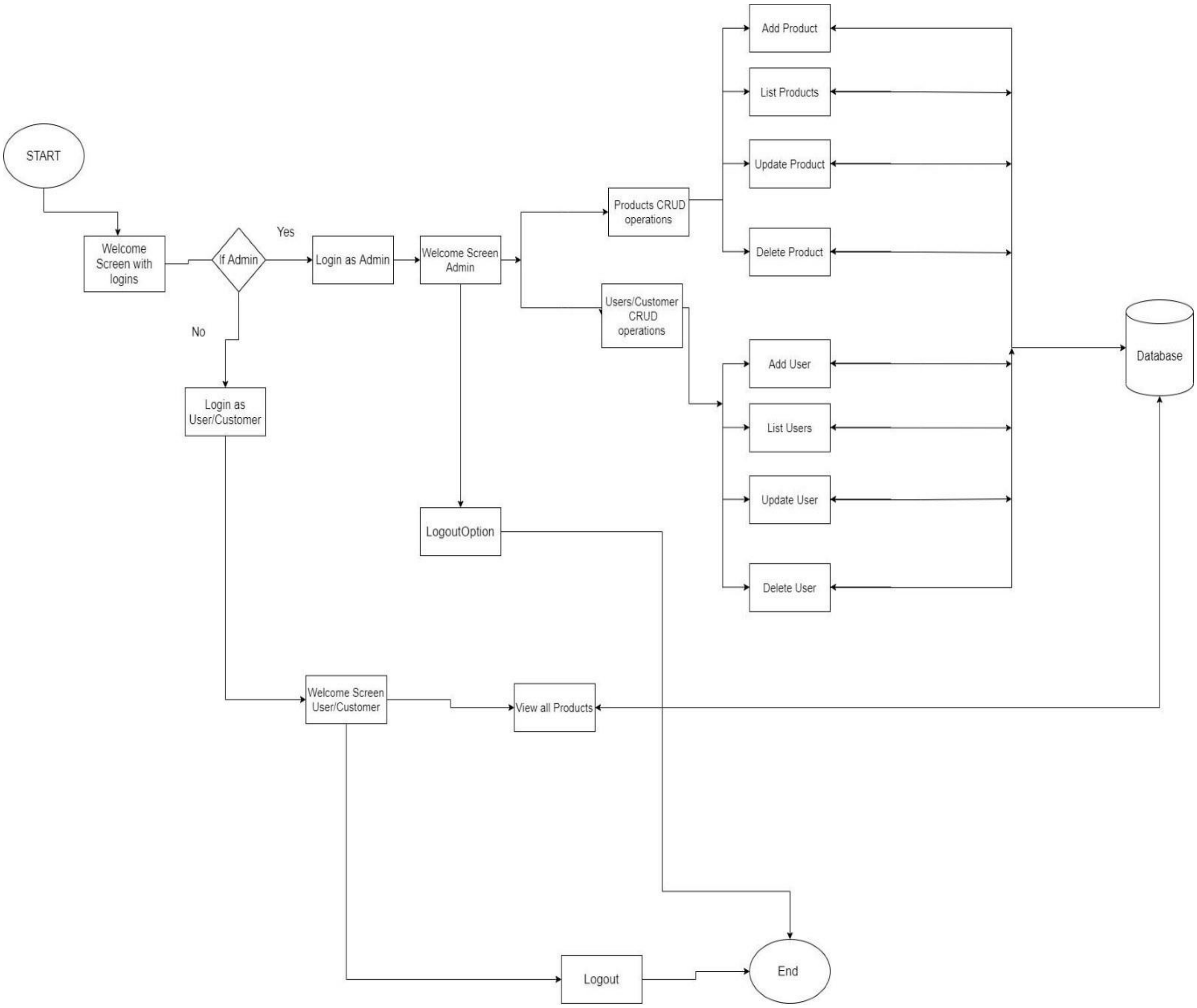
PROJECT DESCRIPTION :

This application is designed and developed as an e-commerce portal Sportyshoes.com for a walk-in store. This system allows Administrator to create (add), read (list), update and delete the customers of the application as well as the products of the application. This project helps the administrator to manage the products and the customers well. The customer can login into the system as well as view the products added in the portal by the authorized admin.

Java concepts used in the project

- OOPS Concepts such as Classes and Objects.
- JSP
- Maven
- Spring MVC Framework with DB
- Spring Restful
- Spring security
- JPQL - java persistence query language
- Annotations
- POSTMAN
- Exception Handling
- Lombok
- Collections such as List
- Loops such as For Loops
- SQL - MySQL database
- Tomcat Server

FLOW CHART OF THE APPLICATION



PROJECT STORIES : (Agile and Scrum)

- As a developer I have develop an application for E-commerce portal for walk-in shoe store.
- I have developed an application in which proper single administrator login is there.
- Developed an application in which multiple customers registered into the system can login.
- As a developer I have develop an application in which administrator is able to view the list of products, customers from a database.
- I have developed an application in which administrator is able to add, update and delete, retrieve the details of products, customers from a database.
- I have developed an application in which User is able see the details of the products and buy it as per their need.

SPRINTS

- Understanding the problem statement of the project.
- Creating the flow chart of the application.
- Creating Spring Starter Project.
- Adding necessary dependencies in pom.xml file.
- Creating Rest Controller, Service, Repository Interface as well as Entity Packages and Classes needed for Back End testing in POSTMAN.
- Creating Spring MVC Controller, Service, Repository Interface as well as Entity Packages needed for working with JSP Pages.
- Using Spring security for login of Administrator.
- Writing JPA-Hibernate template to connect with the database MySQL.
- Testing at each step for different user inputs.
- Initializing the git repository.
- Pushing the code to the GitHub
- Creating the Specification document for showing the Product’s capabilities, appearance and the user interactions.

Source Code of the Project:

```
package com.ecommerce.demo.POJO;

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



import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@Entity
@NoArgsConstructor
@AllArgsConstructor

public class ShoeData {
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
    private int shoeId;
private String shoeName;
private String shoePrice;
```

```
}
package com.ecommerce.demo.POJO;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@Entity
@NoArgsConstructor
@AllArgsConstructor
public class UserData {

@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private int id;
private String name;
private String email;
private String phono;
private String username;
private String password;
```

```
}
package com.ecommerce.demo.POJO;

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

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@Entity
@NoArgsConstructor
@AllArgsConstructor
public class BankData {
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
 private int bid;
private String bname;
private long amount;
}
```

```
}

package com.ecommerce.demo.DAO;

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

import com.ecommerce.demo.POJO.BankData;

import com.ecommerce.demo.Repo.BankRepo;

@Service
public class BankDAO {
@Autowired
BankRepo repo;


public BankData insertbank(BankData b) {

return repo.save(b);
}

public long getBal(String bname) {
return repo.getBal(bname);
}
```

```
}
package com.ecommerce.demo.DAO;

import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.ecommerce.demo.POJO.ShoeData;
import com.ecommerce.demo.Repo.ShoeRepo;

@Service
public class ShoeDAO {

@Autowired
ShoeRepo repo;

public ShoeData insert(ShoeData s) {

return repo.save(s);
}

public List<ShoeData> insertall(List<ShoeData> shoe){
return repo.saveAll(shoe);

}

public List<ShoeData> getall(){
```

```
return repo.findAll();
}

public void deletebyid(int id){
repo.deleteById(id);
}

public ShoeData updateshoe(ShoeData shoe) {
ShoeData shoedata =repo.findById(shoe.getShoeId()).orElse(null);
shoedata.setShoeName(shoe.getShoeName());
shoedata.setShoePrice(shoe.getShoePrice());
returnrepo.save(shoedata);

}

public void delete(ShoeData s) {
repo.delete(s);


}

// for search
public List<ShoeData> listAll(String keyword){
if(keyword!=null) {
return repo.findAll(keyword);
}
```

```
return null;
}
//service
//customize query
public String findbyname(String name,int id){
return repo.findByname(name,id);
}



}

package com.ecommerce.demo.DAO;

import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.ecommerce.demo.POJO.UserData;
import com.ecommerce.demo.Repo.UserRepo;

@Service
public class UserDAO {
@Autowired
UserRepo repo;
```

```
public UserData insertuser(UserData u) {

return repo.save(u);
}

public List<UserData> insertalluser(List<UserData> user){
return repo.saveAll(user);

}

public List<UserData> getalluser(){
return repo.findAll();
}

public void deleteuserbyid(int id){
repo.deleteById(id);
}

public UserData updateuser(UserData user) {
UserData userdata =repo.findById(user.getId()).orElse(null);
userdata.setName(user.getName());
userdata.setEmail(user.getEmail());
userdata.setPhono(user.getPhono());
userdata.setUsername(user.getUsername());
userdata.setPassword(user.getPassword());
returnrepo.save(userdata);
```

```
}

public void deleteuser(UserData u) {
repo.delete(u);


}

// for search
public List<UserData> listAlluser(String keyword){
if(keyword!=null) {
return repo.findAll(keyword);
}
return null;
}

//service
//customize query
public String finduserbyname(String name,int id){
return repo.findByname(name,id);
}


public UserData findByuserpwd(String username,String password) {
```

```
return repo.findByuserpwd(username, password);
}

public boolean checkEmail(String email) {

return repo.existsByEmail(email);
}



}
```

```
package com.ecommerce.demo.Repo;
import java.util.List;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import com.ecommerce.demo.POJO.UserData;
public interface UserRepo extends JpaRepository<UserData, Integer> {




@Query("select u from UserData u where u.username=?1 and u.password=?2")
public UserData findByuserpwd(String username,String password);

@Query("select u.name from UserData u where u.name=?1 and u.id=?2")
public String findByname(String name,int id);

//Search query
@Query("SELECT u from UserData u WHERE u.name LIKE %?1%"
    +"OR u.email LIKE %?1%")
public List<UserData> findAll(String Keyword);


//query to check email
public boolean existsByEmail(String email);
```

```
}
package com.ecommerce.demo.Repo;

import java.util.List;


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

import com.ecommerce.demo.POJO.ShoeData;



public interface ShoeRepo  extends JpaRepository<ShoeData, Integer> {

@Query("select s.shoeName from ShoeData s where s.shoeName=?1 and s.shoeId=?2")
public String findByname(String name,int id);

//Search query
@Query("SELECT s from ShoeData s WHERE s.shoeName LIKE %?1%")
public List<ShoeData> findAll(String Keyword);
```

```
}
package com.ecommerce.demo.Repo;

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

import com.ecommerce.demo.POJO.BankData;

public interface BankRepo extends JpaRepository<BankData, Integer>{
@Query("Select b.amount from BankData b where b.bname=?1")
public long getBal(String bname);
}
```

```
package com.ecommerce.demo.RestController;
import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
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.RestController;
import com.ecommerce.demo.DAO.UserDAO;
import com.ecommerce.demo.POJO.UserData;



@RestController
public class UserRestController {

@Autowired
UserDAO dao;



@PostMapping("/insertuser")
public UserData insertuser(@RequestBody UserData u) {
```

```
return dao.insertuser(u);
}

@PostMapping("/insertalluser")
public List<UserData> insertall(@RequestBody List<UserData> user){
return dao.insertalluser(user);
}

@GetMapping("/getalluser")
public List<UserData> getall(){
return dao.getalluser();
}

@DeleteMapping("deleteuserbyid/{id}")
public String deletebyid(@PathVariable("id") int id) {
dao.deleteuserbyid(id);
return "deleted the id value of "+ id;
}

@PutMapping("/updateuser")
public UserData updateByName(@RequestBody UserData user) {
return dao.updateuser(user);
}
```

```
@GetMapping("/getuserbyname/{name}/{id}")
public String findbyname(@PathVariable("name") String name,@PathVariable("id") int id) throws Exception{
if(dao.finduserbyname(name, id)!=null) {
return dao.finduserbyname(name, id);
}
else {
throw new Exception("the name and the id is not found");
}
}


}
package com.ecommerce.demo.RestController;

import java.io.IOException;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
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.RestController;
import com.ecommerce.demo.DAO.ShoeDAO;
import com.ecommerce.demo.POJO.ShoeData;

@RestController
public class ShoeRestController {

@Autowired
ShoeDAO dao;

@PostMapping("/insert")
public ShoeData insert(@RequestBody ShoeData s) {

return dao.insert(s);
}
//insert a list of objects
@PostMapping("/insertall")
public List<ShoeData> insertall(@RequestBody List<ShoeData> shoe){
return dao.insertall(shoe);
}

@GetMapping("/getall")
public List<ShoeData> getall(){
return dao.getall();
}
```

```
@DeleteMapping("deletebyid/{id}")
public String deletebyid(@PathVariable("id") int id) {
dao.deletebyid(id);
return "deleted the id value of "+ id;
}

@PutMapping("/update")
public ShoeData updateByName(@RequestBody ShoeData shoe) {
return dao.updateshoe(shoe);
}



@GetMapping("/getbyname/{name}/{id}")
public String findbyname(@PathVariable("name") String name,@PathVariable("id") int id) throws Exception{
if(dao.findbyname(name, id)!=null) {
return dao.findbyname(name, id);
}
else {
throw new Exception("the name and the id is not found");
}
}


}
```

```
package com.ecommerce.demo.RestController;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;

import com.ecommerce.demo.DAO.BankDAO;
import com.ecommerce.demo.POJO.BankData;

@RestController
public class BankRestController {
@Autowired
BankDAO dao;


@PostMapping("/insertbank")
public BankData insertbank(@RequestBody BankData b) {

return dao.insertbank(b);
}
}
```

```
package com.ecommerce.demo.Exception;
import java.util.Date;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class ErrorMsg {
private Date timestamp;
private String errormsg;
private String path;
}
package com.ecommerce.demo.Exception;

import java.util.Date;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.context.request.WebRequest;
```

```
@ControllerAdvice
public class GlobalExceptionHandler {
//general exception handler
@ExceptionHandler(Exception.class)
public  ResponseEntity<?> handleNotFoundException(Exception exception,WebRequest request){
ErrorMsg details=new ErrorMsg(new Date(),exception.getMessage(),request.getDescription(false) );
return new ResponseEntity<>(details,HttpStatus.NOT_FOUND);
}
}


package com.ecommerce.demo;

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);
}

}
package com.ecommerce.demo;

import org.springframework.boot.builder.SpringApplicationBuilder;
import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;

public class ServletInitializer extends SpringBootServletInitializer {

@Override
protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {
return application.sources(SportyShoesApplication.class);
}

}
package com.ecommerce.demo;

import org.springframework.context.annotation.Bean;
import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
import org.springframework.security.crypto.password.NoOpPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;

@EnableWebSecurity

public class SecurityConfig extends WebSecurityConfigurerAdapter{

@Override
protected void configure(AuthenticationManagerBuilder auth) throws Exception {
auth.inMemoryAuthentication()
.withUser("admin")
.password("admin123")
.roles("ADMIN")
.and()
.withUser("user")
.password("user")
.roles("USER");

}


//no password encoder used
@Bean
public PasswordEncoder getpassword() {
return NoOpPasswordEncoder.getInstance();
}
//authorization
@Override
protected void configure(HttpSecurity http) throws Exception {
http.authorizeRequests()
.antMatchers("/adminlogin").hasRole("ADMIN")
//        .antMatchers("/access2").hasRole("USER")
//        .antMatchers("/access3").hasAnyRole("ADMIN","USER")
.antMatchers("/").permitAll().and().formLogin();
}

}
```

Main Controller:

```
package com.ecommerce.demo;

import java.util.List;
import java.util.logging.Logger;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;

import com.ecommerce.demo.DAO.BankDAO;
import com.ecommerce.demo.DAO.ShoeDAO;
import com.ecommerce.demo.DAO.UserDAO;
import com.ecommerce.demo.POJO.ShoeData;
import com.ecommerce.demo.POJO.UserData;

@Controller
public class MainController {
@Autowired
```

```
ShoeDAO sdao;
@Autowired
    UserDAO udao;
@Autowired
BankDAO bdao;

Logger log=Logger.getAnonymousLogger();



@RequestMapping("/")
public ModelAndView displaypage(HttpServletRequest
request,HttpServletResponse response) {
ModelAndView mv=new ModelAndView();
mv.setViewName("websitepage.jsp");
return mv;
}



@RequestMapping("/adminlogin")
public ModelAndView adminloginpage(HttpServletRequest
request,HttpServletResponse response) {
ModelAndView mv=new ModelAndView();
String msg="Welcome Admin ";
mv.addObject("msg",msg);
mv.setViewName("adminhomepage.jsp");
return mv;
}
```

```
@RequestMapping("/insertshoe")
public ModelAndView shoeinsertControl(HttpServletRequest request,HttpServletResponse response) {
log.info("Entered into the insert method");
ModelAndView mv=new ModelAndView();
ShoeData s=new ShoeData();
s.setShoeName(request.getParameter("shoeName"));
s.setShoePrice(request.getParameter("shoePrice"));
ShoeData sd=sdao.insert(s);
if(sd!=null) {
mv.setViewName("success.jsp");
}
return mv;
}

@RequestMapping("/editshoe")
public ModelAndView shoeupdateControl(HttpServletRequest request,HttpServletResponse response) {
ModelAndView mv=new ModelAndView();
ShoeData s=new ShoeData();
s.setShoeId(Integer.parseInt(request.getParameter("shoeId")));
s.setShoeName(request.getParameter("shoeName"));
s.setShoePrice(request.getParameter("shoePrice"));
ShoeData sd=sdao.updateshoe(s);
if(sd!=null) {
mv.setViewName("success.jsp");
}
return mv;
}

@RequestMapping("/deleteshoe")
public ModelAndView shoedeleteControl(HttpServletRequest
request,HttpServletResponse response) {
ModelAndView mv=new ModelAndView();
ShoeData s=new ShoeData();
s.setShoeId(Integer.parseInt(request.getParameter("shoeId")));
    sdao.delete(s);
mv.setViewName("success.jsp");
return mv;
}

@RequestMapping("/shoelist")
public ModelAndView displayAll(HttpServletRequest
request,HttpServletResponse response) {
ModelAndView mv=new ModelAndView();
List<ShoeData> list=sdao.getall();
mv.setViewName("shoelist.jsp");
mv.addObject("list",list);
return mv;
}
```

```
@RequestMapping("/searchshoe")
public ModelAndView search (HttpServletRequest request,HttpServletResponse response,@Param("keyword")String keyword) {
log.info("Entered into the search method");
ModelAndView mv=new ModelAndView();
List<ShoeData> listAll=sdao.listAll(keyword);
mv.setViewName("shoelist.jsp");
mv.addObject("list",listAll);
return mv;

}


@RequestMapping("/insertuser")
public ModelAndView userinsertControl(HttpServletRequest request,HttpServletResponse response) {
log.info("Entered into the insert method");
ModelAndView mv=new ModelAndView();
UserData u=new UserData();
u.setName(request.getParameter("name"));
u.setEmail(request.getParameter("email"));
u.setPhono(request.getParameter("phono"));
u.setUsername(request.getParameter("username"));
u.setPassword(request.getParameter("password"));
UserData ud=udao.insertuser(u);
```

```
if(ud!=null) {
mv.setViewName("success.jsp");
}
return mv;
}

@RequestMapping("/userlist")
public ModelAndView displayAlluser(HttpServletRequest
request,HttpServletResponse response) {
ModelAndView mv=new ModelAndView();
List<UserData> list=udao.getalluser();
mv.setViewName("userlist.jsp");
mv.addObject("list",list);
return mv;
}

@RequestMapping("/searchuser")
public ModelAndView searchuser (HttpServletRequest
request,HttpServletResponse response,@Param("keyword")String keyword) {
log.info("Entered into the search method");
ModelAndView mv=new ModelAndView();
List<UserData> listAll=udao.listAlluser(keyword);
mv.setViewName("userlist.jsp");
mv.addObject("list",listAll);
return mv;
```

```
}
@RequestMapping("/edituser")
public ModelAndView userupdateControl(HttpServletRequest
request,HttpServletResponse response) {
ModelAndView mv=new ModelAndView();
UserData u=new UserData();
u.setId(Integer.parseInt(request.getParameter("id")));
u.setName(request.getParameter("name"));
u.setEmail(request.getParameter("email"));
u.setPhono(request.getParameter("phono"));
u.setUsername(request.getParameter("username"));
u.setPassword(request.getParameter("password"));
UserData ud=udao.updateuser(u);
if(ud!=null) {
mv.setViewName("success.jsp");
}
return mv;
}

@RequestMapping("/deleteuser")
public ModelAndView userdeleteControl(HttpServletRequest
request,HttpServletResponse response) {
ModelAndView mv=new ModelAndView();
UserData u=new UserData();
u.setId(Integer.parseInt(request.getParameter("id")));
    udao.deleteuser(u);
```

```
mv.setViewName("success.jsp");
return mv;
}

@RequestMapping("/userlogin")
public ModelAndView userloginpage(HttpServletRequest
request,HttpServletResponse response) {
ModelAndView mv=new ModelAndView();
String username=request.getParameter("username");
String password=request.getParameter("password");
UserData ud=udao.findByuserpwd(username, password);
if(ud!=null) {
String msg="Welcome "+username;
mv.addObject("msg",msg);
mv.setViewName("displaypageuser.jsp");
}
else {
mv.setViewName("fail.jsp");
}
return mv;

}

@RequestMapping("/registeruser")
public ModelAndView registeruserpage(HttpServletRequest
request,HttpServletResponse response) {
```

```
ModelAndView mv=new ModelAndView();
String email=request.getParameter("email");
boolean b =udao.checkEmail(email);
if(b){
String msg="Email already Exists";
mv.setViewName("registeruser1.jsp");
mv.addObject("msg",msg);

}
else {
UserData u=new UserData();
u.setName(request.getParameter("name"));
u.setEmail(request.getParameter("email"));
u.setPhono(request.getParameter("phono"));
u.setUsername(request.getParameter("username"));
u.setPassword(request.getParameter("password"));
UserData ud=udao.insertuser(u);
if(ud!=null) {
String msg="Registered Successfully";
String name=u.getName();
mv.addObject("name",name);
mv.setViewName("userlogin1.jsp");
mv.addObject("msg",msg);
}
}
```

```
return mv;

}



@RequestMapping("/usershoelist")

public ModelAndView usershoelistpage(HttpServletRequest
request,HttpServletResponse response) {

ModelAndView mv=new ModelAndView();

List<ShoeData> list=sdao.getall();

mv.setViewName("usershoelist.jsp");

mv.addObject("list",list);

return mv;


}


@RequestMapping("/buyshoe")

public ModelAndView butshoepage(HttpServletRequest
request,HttpServletResponse response) {

ModelAndView mv=new ModelAndView();

String shoeprice=request.getParameter("shoePrice");

int shprice=Integer.parseInt(shoeprice);

String bname=request.getParameter("bname");

Long bal=bdao.getBal(bname);


if(bal>=shprice) {
```

```
mv.setViewName("paysuccess.jsp");
}
else {
mv.setViewName("payfail.jsp");
}

return mv;}
}
```

application.properties file:

```
server.port=8082
#jpa-hibernate dialet,ddl-auto,show-sql
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.hibernate.dialect=org.hibernate.dialect.MySQLDialect

#datasource  driverClass,url,un,pwd
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/db9
spring.datasource.username=root
spring.datasource.password=123456
```

Pom.xml file:

```xml
<?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
https://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>2.7.7</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>com.ecommerce</groupId>
<artifactId>SportyShoes</artifactId>
<version>0.0.1</version>
<packaging>war</packaging>
<name>SportyShoes</name>
<description>Demo project for Spring Boot</description>
<properties>
<java.version>1.8</java.version>
</properties>
<dependencies>
<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</artifactId>
</dependency>
<!-- https://mvnrepository.com/artifact/org.apache.tomcat/tomcat-jasper -->
<dependency>
    <groupId>org.apache.tomcat</groupId>
    <artifactId>tomcat-jasper</artifactId>
    <version>9.0.69</version>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-devtools</artifactId>
<scope>runtime</scope>
<optional>true</optional>
</dependency>
<dependency>
<groupId>com.mysql</groupId>
<artifactId>mysql-connector-j</artifactId>
<scope>runtime</scope>
</dependency>


<dependency>
```

```
<groupId>org.projectlombok</groupId>
<artifactId>lombok</artifactId>
<optional>true</optional>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-tomcat</artifactId>
<scope>provided</scope>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>


<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-security</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.security</groupId>
<artifactId>spring-security-test</artifactId>
<scope>test</scope>
</dependency>
```

```
</dependencies>

<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
<configuration>
<excludes>
<exclude>
<groupId>org.projectlombok</groupId>
<artifactId>lombok</artifactId>
</exclude>
</excludes>
</configuration>
</plugin>
</plugins>
</build>

</project>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>SportyShoes Website</title>
</head>
<body>
<h1>Welcome to SportyShoes Website</h1>
<br>
<form action="adminlogin">
<input type="submit" value="Admin">
</form>
<br>
<br>
<form action="userlogin.jsp">
<input type="submit" value="User">
</form>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

      <%@page import="com.ecommerce.demo.POJO.*" %>
    <%@page import="java.util.*" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<a href="Logout.jsp">Logout</a>
<br><br>

<h1 align="center">List of Shoes</h1>
<%@include file="searchshoe.jsp" %><br><br>

<%List<ShoeData> s=(List<ShoeData>)request.getAttribute("list");%>
<table border="1">
<tr><th>ShoeId</th><th>Name</th><th>Price</th><th>Action</th></tr>
<%for(ShoeData sd:s){%>
<tr><td><%=sd.getShoeId() %></td><td><%=sd.getShoeName()
%></td><td><%=sd.getShoePrice()%></td><td><a
href="buyshoe.jsp?shoePrice=<%=sd.getShoePrice() %>
">BuyNow</a></td></tr>
<%}%>
</table>
</body>
</html>


<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
    <%@page import="com.ecommerce.demo.POJO.*" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

<h2><%="Welcome "+request.getAttribute("name") %></h2>

<h3><%=request.getAttribute("msg") %></h3>
<%@include file="userlogin.jsp" %>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

<h1>Welcome to SportyShoes Website</h1>
<br>
<form action="userlogin">



Username <input type="text" name="username"><br><br>

Password <input type="password" name="password"><br><br>

<input type="submit" value="Login">
<br><br>
<i>Click on the link for registration</i>
<a href="registeruser.jsp">Register here </a>
</form>
```

```
</body>
</html>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

      <%@page import="com.ecommerce.demo.POJO.*" %>
    <%@page import="java.util.*" %>

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<a href="adminhomepage.jsp">Home</a>&nbsp
<a href="Logout.jsp">Logout</a>
<br><br>
<h1 align="center"><a href="insertuser.jsp">Insert More Users</a></h1>

<%@include file="searchuser.jsp" %><br><br>

<%List<UserData> u=(List<UserData>)request.getAttribute("list");%>

<table border="1">
```

```
<tr><th>UserId</th><th>Name</th><th>Email</th><th>Phone
No.</th><th>Username</th><th>Password</th><th>Action</th></tr>
<%for(UserData ud:u){%>
<tr><td><%=ud.getId()%></td><td><%=ud.getName()%></td><td><%=ud.getE
mail()%></td><td><%=ud.getPhono()%></td><td><%=ud.getUsername()%></td
><td><%=ud.getPassword()%></td><td><a
href="edituser.jsp?id=<%=ud.getId()%>">Edit</a><br><a
href="deleteuser.jsp?id=<%=ud.getId()%>">Delete</a></td></tr>
<%}%>


</table>
</body>
</html>


<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1><i>Task Done successfully</i></h1>
<form action="shoelist">
<input type="submit" value="Shoelist">
</form>
```

```
<br>
<form action="userlist">
<input type="submit" value="Userlist">
</form>
</body>
</html>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

     <%@page import="com.ecommerce.demo.POJO.*" %>
    <%@page import="java.util.*" %>

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<a href="adminhomepage.jsp">Home</a>&nbsp
<a href="Logout.jsp">Logout</a>
<br><br>
<h1 align="center"><a href="insertshoe.jsp">Insert More Shoes</a></h1>

<%@include file="searchshoe.jsp" %><br><br>
```

```
<%List<ShoeData> s=(List<ShoeData>)request.getAttribute("list");%>


<table border="1">

<tr><th>ShoeId</th><th>Name</th><th>Price</th><th>Action</th></tr>

<%for(ShoeData sd:s){%>

<tr><td><%=sd.getShoeId() %></td><td><%=sd.getShoeName()
%></td><td><%=sd.getShoePrice()%></td><td><a
href="editshoe.jsp?shoeId=<%=sd.getShoeId()%>">Edit</a><br><a
href="deleteshoe.jsp?shoeId=<%=sd.getShoeId()%>">Delete</a></td></tr>

<%}%>

</table>


</body>

</html>


<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

    pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<form action="searchuser">

UserName or Email: <input type="text" name="keyword" required />&nbsp;
```

```
<input type="submit" value="Search"/>&nbsp;
<input type="submit" value="Clear"/>
</form><br> <br>
<form action="userlist">
<input type="submit" value="Userlist">
</form>

</body>
</html>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

<form action="searchshoe">
ShoeName: <input type="text" name="keyword" required />&nbsp;
<input type="submit" value="Search"/>&nbsp;
<input type="submit" value="Clear"/>
</form><br> <br>
<form action="shoelist">
```

```
<input type="submit" value="Shoelist">
</form>

</body>
</html>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1><%=request.getAttribute("msg") %></h1>
<%@include file="registeruser.jsp" %>
</body>
</html>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

    <%@page import="com.ecommerce.demo.POJO.*" %>
<!DOCTYPE html>
<html>
```

```
<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<a href="userlogin.jsp">Login</a>&nbsp

<a href="Logout.jsp">Logout</a><br>

<h1>Register-SignUp</h1>

<br>



<form action="registeruser">



Full Name <input type="text" name="name"><br><br>


Email     <input type="email" name="email"><br><br>


Phone No. <input type="text" name="phono"><br><br>


Username  <input type="text" name="username"><br><br>


Password  <input type="password" name="password"><br><br>


<input type="submit" value="SignUp">
```

```
</form>




</body>
</html>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

<a href="Logout.jsp">Logout</a><br>

<h1>Transaction Successful...Payment Done</h1>
<h1><i>Display the Shoes List</i></h1>
<form action="usershoelist">
<input type="submit" value="Shoelist">
</form>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

<a href="Logout.jsp">Logout</a><br>

<h1>Transaction Failed...Insufficient Balance</h1>
<h1><i>Display the Shoes List</i></h1>
<form action="usershoelist">
<input type="submit" value="Shoelist">
</form>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
```

```
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

<a href="websitepage.jsp">Login</a>&nbsp
<br>
<br>
<%="Logout Successfully" %>
<br>
<br>

</body>
</html>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<h1>Insert User Details</h1>
```

```
<body>

<%@include file="searchuser.jsp" %>
<br><br>

<form action="insertuser">

Name <input type="text" name="name"><br><br>

Email <input type="email" name="email"><br><br>

Phone No. <input type="text" name="phono"><br><br>

Username <input type="text" name="username"><br><br>

Password <input type="text" name="password"><br><br>

<input type="submit" value="Insert">

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

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<h1>Insert Shoes Details</h1>
<body>

<%@include file="searchshoe.jsp" %>
<br><br>

<form action="insertshoe">
ShoeName <input type="text" name="shoeName"><br><br>

Price <input type="text" name="shoePrice"><br><br>

<input type="submit" value="Insert">

</form>
</body>
</html>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
```

```
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h2><i>Invalid Username or Password!!</i></h2>
<%@include file="userlogin.jsp" %>
<br>
<i>Click on the link for registration</i>
<a href="registeruser.jsp">Register here </a>
</body>
</html>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
```

```
<h1><i>Edit the User</i></h1>
<form action="edituser">
<input type="hidden" name="id" value=<%=request.getParameter("id") %>>

Name <input type="text" name="name"><br><br>

Email <input type="email" name="email"><br><br>

Phone No. <input type="text" name="phono"><br><br>

Username <input type="text" name="username"><br><br>

Password <input type="text" name="password"><br><br>

<input type="submit" value="Edit">

</form>
</body>
</html>


<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
```

```
<title>Insert title here</title>
</head>
<body>

<h1><i>Edit the Shoes</i></h1>
<form action="editshoe">
<input type="hidden" name="shoeId" value=<%=request.getParameter("shoeId")
%>>

ShoeName <input type="text" name="shoeName"><br><br>

Price <input type="text" name="shoePrice"><br><br>

<input type="submit" value="Edit">

</form>
</body>
</html>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
```

```
</head>
<body>
<h2><%=request.getAttribute("msg") %></h2>
<h1><i>Display the Shoes List</i></h1>
<form action="usershoelist">
<input type="submit" value="Shoelist">
</form>
</body>
</html>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1><i>Delete the User</i></h1>
<form action="deleteuser">
<input type="hidden" name="id" value=<%=request.getParameter("id") %>>

<input type="submit" value="Delete">
</form>
</body>
```

```
</html>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

<h1><i>Delete the Shoes</i></h1>
<form action="deleteshoe">
<input type="hidden" name="shoeId" value=<%=request.getParameter("shoeId")
%>>

<input type="submit" value="Delete">
</form>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="buyshoe">
<input type="hidden" name="shoePrice"
value=<%=request.getParameter("shoePrice") %>>

Bank Name <input type="text" name="bname"><br><br>

<input type="submit" value="Proceed to Pay">


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

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
```

```
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h2><%=request.getAttribute("msg") %></h2>
<h1><i>Display the Lists</i></h1>
<form action="shoelist">
<input type="submit" value="Shoelist">
</form><br> <br>
<form action="userlist">
<input type="submit" value="Userlist">
</form>
</body>
</html>
```