

Make an E-commerce Website for Sporty Shoes

Document contains:

- Project and developer details
- Sprints planned and the tasks achieved in them
- Algorithms and flowcharts of the application
- List of core concepts used
- Links to the GitHub repository to verify the project completion
- Conclusions

Project and developer details

Project objective:

As a Full Stack Developer, complete the features of the application by planning the development and pushing the source code to the GitHub repository.

Background of the problem statement:

Sporty Shoes is a company that manufactures and sells sports shoes. They have a walk-in store, and now, they wish to launch their e-commerce portal sportyshoes.com.

You're asked to develop a prototype of the application. It will be then presented to the relevant stakeholders for budget approval. Your manager has set up a meeting where you're asked to do the following:

Presenting the specification document which has the product's capabilities, appearance, and user interactions

Setting up Git and GitHub account to store and track your enhancements of the prototype

Explaining the Java concepts used in the project

Discussing the generic features of the product:

There will be an admin to manage the website. An administrator login will be required to access the admin page.

The admin should be able to change his password if he wants, he should be able to:

Manage the products in the store including categorizing them

Browse the list of users who have signed up and be able to search users

See purchase reports filtered by date and category

Developer Details: S.V. SAI KUMAR

svsai97@gmail.com

Sprints planned and the tasks achieved in them

There are three sprints for this project

Sprint-1: Studied the Features of project and prepared the flow chart and created Git Repository.

Sprint-2: In Spring Boot created Jsp pages and written Controller ,repository, securityconfig, Dao, Pojo and All the features are being implemented.

Sprint-3: Final testing of website using different inputs and pushing it to GitHub.

Codes:

Users.java

```
package com.example.demo;  
  
import javax.persistence.Column;  
  
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;

@Entity
@Data
@AllArgsConstructor
@NoArgsConstructor

public class users {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int userid;
    private String username;
    private String password;
    private String role;
    private Boolean enabled;
    private String name;
    private String useremail;
    private String contactno;
    private String adress;
    private String pincode;
}
```

usersDao

```
package com.example.demo;

import java.util.List;
import java.util.Optional;

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

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

```
@Service
```

```
public class usersDao {
```

```
    @Autowired
```

```
    usersrepository repo;
```

```
    public users insert(users u) {
```

```
        return repo.save(u);
```

```
    }
```

```
    //retrieve
```

```
    public List<users> getusers() {
```

```
        return repo.findusers();
```

```
    }
```

```
}
```

```
usersrepository
```

```
package com.example.demo;
```

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

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

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

```
public interface usersrepository extends JpaRepository<users,Integer> {
```

```
    @Query("select users from users users where role='USER'")
```

```
    public List<users> findusers();
```

```
}
```

Product.java

```
package com.example.demo;
```

```
import javax.persistence.Column;
```

```
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;
```

```
@Entity
```

```
@Data
```

```
@AllArgsConstructor
```

```
@NoArgsConstructor
```

```
public class product {
```

```
    @Id
```

```
    private Long productid;
```

```
    private String productcategory;
```

```
    private String productname;
```

```
    private String productprice;
```

```
    private String productdescription;
```

```
}
```

productDao.java

```
package com.example.demo;
```

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

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

```
@Service
```

```
public class productDao {
```

```
    @Autowired
```

```
    productrepository repo;
```

```
    public product insert(product p) {
```

```
        return repo.save(p);
```

```
    }
```

```
    //retrieve
```

```
    public List<product> getproduct() {
```

```
        return repo.findAll();
```

```
    }
```

```
    public product delete(long id) {
```

```
        repo.deleteById(id);
```

```
        return null;
```

```
    }
```

```
    public List<product> getproductbycategory(String name) {
```

```
        return repo.findproductbycategory(name);
```

```
    }
```

```
}
```

Productrepositry.java

```
package com.example.demo;
```

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

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

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

```
public interface productrepositry extends JpaRepository<product,Long> {
```

```
    @Query("select product from product product where productcategory=?1")
```

```
    public List<product> findproductbycategory(String name);
```

```
}
```

Cart.java

```
package com.example.demo;
```

```
import java.util.Date;
```

```
import javax.persistence.Column;
```

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

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

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

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

```
import javax.persistence.UniqueConstraint;
```

```
import org.springframework.format.annotation.DateTimeFormat;
```

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

```
@Entity
```

```
@Data
```

```
@AllArgsConstructor
```

```
@NoArgsConstructor
```

```
public class cart {
    @Id
    @GeneratedValue

    private long cartid;
    private String username;
    private Long productid;
    private String productcategory;
    private String productname;
    private String productprice;

    private String orderno;
    @DateTimeFormat(pattern="dd-MM-yyyy")
    private Date dateofpurchase;
    private String payment;
}
```

cartDao.java

```
package com.example.demo;

import java.util.Date;
import java.util.List;

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



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

```
@Service
```

```
public class cartDao {
```

```
    @Autowired
```

```
    cartrepository repo;
```

```
    public cart insert(cart c) {  
        return repo.save(c);
```

```
    }
```

```
    public cart deletebyid(long id) {  
        repo.deleteById(id);  
        return null;
```

```
    }
```

```
    public List<cart> getcartbyusername(String name) {  
        return repo.findcartbyusername(name);  
    }
```

```
    public List<cart> getcartbypayment() {  
        return repo.findcartbypayment();  
    }
```

```
    public void updatebyuser(Date Date,String ordeno,String payment ,String username) {  
        repo.updatebyuser(Date, ordeno, payment, username);
```

```
}
```

```
}
```

Cartrepository.java

```
package com.example.demo;
```

```
import java.util.Date;
```

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

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

```
import org.springframework.data.jpa.repository.Modifying;
```

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

```
import org.springframework.transaction.annotation.Transactional;
```

```
public interface cartrepository extends JpaRepository<cart,Long> {
```

```
    @Query("select cart from cart cart where username=?1 and payment='notpaid'")
```

```
    public List<cart> findcartbyusername(String name);
```

```
    @Transactional
```

```
    @Modifying
```

```
    @Query("update cart set dateofpurchase=?1,orderno=?2,payment=?3 where username=?4 and  
payment='notpaid'")
```

```
    public void updatebyuser(Date date,String orderno,String payment ,String username);
```

```
    @Query("select cart from cart cart where payment='paid'")
```

```
    public List<cart> findcartbypayment();
```

```
}
```

mainController.java

```
package com.example.demo;

import java.io.IOException;
import java.io.PrintWriter;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.Optional;

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

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.Authentication;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.servlet.ModelAndView;

@Controller
@RequestMapping("/")
```

```
public class mainController {

    @Autowired
    usersDao udao;

    @Autowired
    productDao pdao;

    @Autowired
    cartDao cdao;

    public String amount;
    public String id;

    @RequestMapping("/")
    public ModelAndView displaydefaultpage(HttpServletRequest request,HttpServletResponse
response)
    {

        ModelAndView mv=new ModelAndView();
        mv.setViewName("home.jsp");
        return mv;
    }

    @RequestMapping("/loginhome")
    public ModelAndView adminhome(HttpServletRequest request,HttpServletResponse response)
    {

        ModelAndView mv=new ModelAndView();
        Authentication authentication = SecurityContextHolder.getContext().getAuthentication();
        String role=authentication.getAuthorities().stream().findAny().get().getAuthority();
        System.out.println(role);
    }
}
```

```
        if (role.equals("ADMIN")) {

            mv.setViewName("adminHome.jsp");

        }

        if (role.equals("USER")) {

            mv.setViewName("userHome.jsp");

        }

        return mv;

    }
}
```

`@RequestMapping("/updatepassword")`

`public ModelAndView updatepassword(users u ,HttpServletRequest request,HttpServletResponse response) throws IOException {`

```
    ModelAndView mv=new ModelAndView();

    u.setUserid(1);

    u.setUsername("admin");

    u.setEnabled(true);

    u.setRole("ADMIN");

    u.setPassword(request.getParameter("password"));

    udao.insert(u);

    mv.setViewName("updatesucesshome.jsp");

    return mv;

}
}
```

`@RequestMapping("/userRegistration")`

`public ModelAndView registeruserdetails(users u ,HttpServletRequest request,HttpServletResponse response) throws IOException {`

```
    ModelAndView mv=new ModelAndView();

    u.setUsername(request.getParameter("username"));

    u.setPassword(request.getParameter("password"));

    u.setName(request.getParameter("name"));

}
```

```
        u.setContactno(request.getParameter("contactno"));
        u.setPincode(request.getParameter("pincode"));
        u.setName(request.getParameter("useremail"));
        u.setName(request.getParameter("adress"));
        u.setEnabled(true);
        u.setRole("USER");

        udao.insert(u);

        mv.setViewName("updatesucesshome.jsp");

        return mv;
    }
}
```

```
@RequestMapping("/listusers")
```

```
public ModelAndView getalluser(HttpServletRequest request, HttpServletResponse response) {
    ModelAndView mv=new ModelAndView();
    List<users> list=udao.getusers();
    mv.setViewName("userlist.jsp");
    mv.addObject("list", list);
    return mv;
}
}
```

```
@RequestMapping("/searchuser")
```

```
public ModelAndView getuser(users u, HttpServletRequest request, HttpServletResponse response) {
    ModelAndView mv=new ModelAndView();
    List<users> list=udao.getusers();
    mv.setViewName("userlistsearch.jsp");
    mv.addObject("list", list);
    u.setUsername(request.getParameter("username"));
    return mv;
}
```

```
}
```

```
@RequestMapping("/addproduct")
```

```
public ModelAndView addproductdetails(product p ,HttpServletRequest  
request,HttpServletResponse response) throws IOException {
```

```
    ModelAndView mv=new ModelAndView();
```

```
    String pid=request.getParameter("productid");
```

```
    if (pid=="") {
```

```
        pid = "0";
```

```
    }
```

```
    Long id=Long.parseLong(pid);
```

```
    String pcategory=request.getParameter("productcategory");
```

```
    String pname=request.getParameter("productname");
```

```
    String pprice=request.getParameter("productprice");
```

```
    if (pprice=="") {
```

```
        pprice= "0";
```

```
    }
```

```
    Double price=Double.parseDouble(pprice);
```

```
    String pdesc=request.getParameter("productdescription");
```

```
    if(id!=0 && pcategory!="" && pname!="" && price!=0 &&pdesc!="") {
```

```
        p.setProductid(Long.parseLong(request.getParameter("productid")));
```

```
        p.setProductcategory(request.getParameter("productcategory"));
```

```
        p.setProductname(request.getParameter("productname"));
```

```
        p.setProductprice(request.getParameter("productprice"));
```

```
        p.setProductdescription(request.getParameter("productdescription"));
```

```
        pdao.insert(p);
```

```
        mv.setViewName("listproduct");
```

```
    }
```

```

        if(id==0 || pcategory==" || pname==" || price==0 || pdesc=="") {

            mv.setViewName("addProduct.jsp");

        }

        return mv;
    }

```

```

@RequestMapping("/updateproduct")

```

```

    public ModelAndView updateproductdetails(product p ,HttpServletRequest
    request,HttpServletResponse response) throws IOException {

```

```

        ModelAndView mv=new ModelAndView();

```

```

        String pcategory=request.getParameter("productcategory");

```

```

        String pname=request.getParameter("productname");

```

```

        String pprice=request.getParameter("productprice");

```

```

        if (pprice=="") {

```

```

            pprice= "0";

```

```

        }

```

```

        Double price=Double.parseDouble(pprice);

```

```

        String pdesc=request.getParameter("productdescription");

```

```

        if( pcategory!=" " && pname!=" " && price!=0 &&pdesc!=" ") {

```

```

            p.setProductid(Long.parseLong(request.getParameter("productid")));

```

```

            p.setProductcategory(request.getParameter("productcategory"));

```

```

            p.setProductname(request.getParameter("productname"));

```

```

            p.setProductprice(request.getParameter("productprice"));

```

```

            p.setProductdescription(request.getParameter("productdescription"));

```

```

            pdao.insert(p);

```



```

        mv.setViewName("listproduct");
    }

    if(pcategory==" " || pname==" " || price==0 || pdesc==" ") {

        mv.setViewName("updateProduct.jsp");

    }
    return mv;
}

```

```

@RequestMapping("/deleteproduct")

```

```

    public ModelAndView deleteproductdetails(HttpServletRequest request,HttpServletResponse
response) throws IOException {

```

```

        ModelAndView mv=new ModelAndView();

        long id =Long.parseLong(request.getParameter("productid"));

        pdao.delete(id);

        mv.setViewName("listproduct");

        return mv;
    }

```

```

@RequestMapping("/listproduct")

```

```

    public ModelAndView getallproduct(HttpServletRequest request,HttpServletResponse response) {

        ModelAndView mv=new ModelAndView();

        List<product> list=pdao.getproduct();

        mv.setViewName("productmodify.jsp");

        mv.addObject("list", list);

        return mv;
    }

```

```

@RequestMapping("/purchasereport")

```

```

    public ModelAndView showcartbypayment(cart c, HttpServletRequest request, HttpServletResponse
response) {

        ModelAndView mv=new ModelAndView();

        List<cart> list=cdao.getcartbypayment();

        mv.addObject("list", list);

        mv.setViewName("purchaseReports.jsp");

        return mv;

    }

```

```

////////////////////////////////////

```

```

@RequestMapping("/productbycategory")

```

```

    public ModelAndView getproductbycategory(HttpServletRequest request,HttpServletResponse
response) {

        ModelAndView mv=new ModelAndView();

        List<product> list=pdao.getProductbycategory(request.getParameter("productcategory"));

        mv.setViewName("productbyCategory.jsp");

        mv.addObject("list", list);

        return mv;

    }

```

```

@RequestMapping("/addtocart")

```

```

    public ModelAndView addcart(cart c, HttpServletRequest request,HttpServletResponse response) {

        ModelAndView mv=new ModelAndView();

        Authentication auth = SecurityContextHolder.getContext().getAuthentication();

        String username = auth.getName();

        Long productid=Long.parseLong(request.getParameter("productid"));

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

        String productPrice=request.getParameter("productprice");

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

        c.setUsername(username);
    }

```

```
        c.setProductid(productid);
        c.setProductname(productname);
        c.setProductcategory(productcategory);
        c.setProductprice(productPrice);
        c.setPayment("notpaid");
        cdao.insert(c);
        mv.setViewName("addcartsucess.jsp");
        return mv;
    }
}
```

```
@RequestMapping("/showCart")
```

```
{
    public ModelAndView showcart(cart c, HttpServletRequest request,HttpServletResponse response)

        ModelAndView mv=new ModelAndView();

        Authentication auth = SecurityContextHolder.getContext().getAuthentication();
        String username = auth.getName();
        List<cart> list=cdao.getcartbyusername( username);
        mv.addObject("list", list);

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

```
@RequestMapping("/failpayment")
```

```
{
    public ModelAndView showcart2(cart c, HttpServletRequest request,HttpServletResponse response)

        ModelAndView mv=new ModelAndView();

        Authentication auth = SecurityContextHolder.getContext().getAuthentication();
        String username = auth.getName();
        List<cart> list=cdao.getcartbyusername( username);
        mv.addObject("list", list);

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

```
@RequestMapping("/removefromcart")
```

```
public ModelAndView removefromcart(cart c, HttpServletRequest request, HttpServletResponse response) {
```

```
    ModelAndView mv=new ModelAndView();
```

```
    long id=Long.parseLong(request.getParameter("cartid"));
```

```
    cdao.deletebyid(id);
```

```
    mv.setViewName("showCart");
```

```
    return mv;
```

```
}
```

```
@RequestMapping("/paymentcheck")
```

```
public ModelAndView paymentcheck( HttpServletRequest request,HttpServletResponse response)  
throws ParseException {
```

```
    ModelAndView mv=new ModelAndView();
```

```
    String cardnum=request.getParameter("cardno");
```

```
    String month=request.getParameter("month");
```

```
    String pswrd=request.getParameter("psd");
```

```
    System.out.println("check");
```

```
    if (cardnum=="") {
```

```
        cardnum = "0";
```

```
    }
```

```
    Long cardno=Long.parseLong(cardnum);
```

```
    if(cardno!=0 && month!="" && pswrd!="") {
```

```
        amount=request.getParameter("amt");
```

```
        id=request.getParameter("order");
```

```
        mv.setViewName("paid.jsp");
```

```
        Authentication auth = SecurityContextHolder.getContext().getAuthentication();
```

```
        String username = auth.getName();
```

```
        Date dNow = new Date( );
```

```
        SimpleDateFormat ft =new SimpleDateFormat ("dd-MM-yyyy");
```

```

        String dt1=ft.format(dNow);

        Date date=new SimpleDateFormat("dd-MM-yyyy").parse(dt1);

        String ordeno=request.getParameter("order");

        String payment="paid";


        cdao.updatebyuser(date, ordeno, payment,username);


    }

    else {

        mv.setViewName("failpayment");

    }

    return mv;

}

}

```

SecurityConfig.java

```

package com.example.demo;


import javax.sql.DataSource;


import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import
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 {

```

@Autowired

DataSource dataSource;

@Override

protected void configure(AuthenticationManagerBuilder auth) throws Exception {

auth.jdbcAuthentication()

.dataSource(dataSource)

.usersByUsernameQuery("select username, password, enabled from users where username=?")

.authoritiesByUsernameQuery("select username, role from users where username=?");

}

@Bean

public PasswordEncoder getPassword() {

return NoOpPasswordEncoder.getInstance();

}

//authorize

@Override

protected void configure(HttpSecurity http) throws Exception {

http.authorizeRequests()

.antMatchers("/updatepassword").hasAuthority("ADMIN")

.antMatchers("/listusers").hasAuthority("ADMIN")

.antMatchers("/searchuser").hasAuthority("ADMIN")

.antMatchers("/addproduct").hasAuthority("ADMIN")

.antMatchers("/updateproduct").hasAuthority("ADMIN")

.antMatchers("/deleteproduct").hasAuthority("ADMIN")

.antMatchers("/listproduct").hasAuthority("ADMIN")

.antMatchers("/purchasereport").hasAuthority("ADMIN")

```

        .antMatchers("/productbycategory").hasAuthority("USER")
        .antMatchers("/addtocart").hasAuthority("USER")
        .antMatchers("/showCart").hasAuthority("USER")
        .antMatchers("/failpayment").hasAuthority("USER")
        .antMatchers("/removefromcart").hasAuthority("USER")
        .antMatchers("/paymentcheck").hasAuthority("USER")

        .antMatchers("/loginhome").hasAnyAuthority("ADMIN","USER")

        .antMatchers("/").permitAll()
        .antMatchers("/userRegistration").permitAll()

        .and().formLogin().defaultSuccessUrl("/loginhome")
        .and().logout().logoutSuccessUrl("/");

    }

}

```

SportyShoesApplication.java

```

package com.example.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);

    }

}

```

Jsp

Home.jsp

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Home</title>
</head>
<body style="background:url(picer.jpg) no-repeat;background-size:90% auto; ">
    <div align="center">

        <i><h1>SportyShoes</h1></i>
        <h4 align="right"> - devloped by S.V. SAI KUMAR</h4>

        <br><br>

        <h1>Welcome, Shop According to Your Sports</h1><br>

        <b> <input style=font-size:24px type="button"
onclick="window.location.href='login';" value="Admin Signin" /></b><br><br><br>

        <b> <input style=font-size:24px type="button"
onclick="window.location.href='login';" value=" User Signin " /></b><br><br>
        <h3><a href='userRegistration.jsp' style="text-decoration: none">new here?
Create Account</a></h3>

        <u></form>

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

```

userRegistration.jsp

```

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

```



```

<head>
<meta charset="ISO-8859-1">
<title>Register</title>
</head>
<body style="background:url(picer.jpg) no-repeat;background-size:90% auto; ">
    <div align="center">

        <i><h1>SportyShoes</h1></i>
        <h4 align="right"> - devloped by S.V. SAI KUMAR</h4><br>
        <br><br>
        <h3>Enter Details</h3>
        <form action="userRegistration"><br>

            <table>
            <tr>
            <td><b>Name</b></td>
            <td><input type ="text" name="name"  ></td>
            </tr>
            <tr>
            <td><b>Email</b></td>
            <td><input type ="email" name="useremail"  ></td>
            </tr>
            <tr>
            <td><b>Phone no</b></td>
            <td><input type ="number" name="contactno"  ></td>
            </tr>
            <tr>
            <td><b>Address</b></td>
            <td><input type ="text" name="adress"  ></td>
            </tr>
            <tr>
            <td><b>Pincode</b></td>
            <td><input type ="number" name="pincode"  ></td>
            </tr>
            <tr>
            <td><b>Username</b></td>
            <td><input type ="text" name="username"  ></td>
            </tr>
            <tr>
            <td><b>Password</b></td>
            <td><input type ="password" name="password"  ></td>
            </tr>

            </table>

            <br><input type ="submit" value="Register">
        </form>

    </div>

</body>
</html>

```

adminHome.jsp

```

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

```


userlistSearch.jsp

[illegible]

purchaseReports.jsp

```
<%@page import="com.example.demo.*"%>
<%@page import="java.util.*"%>
<%@ page import="com.example.*"%>

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



```

        <table>
        <tr>
        <td><b>Product Id</b></td>
        <td><input type = "number" name = "productid"  ></td>
        </tr>
        <tr>
        <td><b>Product Category</b></td>
        <td><select name = "productcategory">
        <option value = "Running shoes">Running shoes</option>
        <option value = "Minimalist shoes">Minimalist shoes</option>
        <option value = "Walking shoes">Walking shoes</option>
        <option value = "Tennis shoes">Tennis shoes</option>
        <option value = "Versatile training shoes">Versatile training
shoes</option>
        <option value = "Trail running shoes">Trail running shoes</option>
        <option value = "Basketball shoes">Basketball shoes</option>
        <option value = "Football boots">Football boots</option>

        </select></td>
        </tr>
        <tr>
        <td><b>Product Name</b></td>
        <td><input type = "text" name = "productname"  ></td>
        </tr>
        <tr>
        <td><b>Product Price Rs</b></td>
        <td><input type = "number" name = "productprice"  >/-</td>
        </tr>
        <tr>
        <td><b>Product Description</b></td>
        <td><input type = "text" name = "productdescription"  ></td>
        </tr>

        </table>

        <br><input type = "submit" value = "Add Product">
</form>

</div>

</body>
</html>

```

productmodify.jsp

```

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

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>product modify</title>
</head>
<body style="background:url(picer.jpg) no-repeat;background-size:90% auto; ">
    <%
        List<product> list=(List<product>)request.getAttribute("list");
    %>

```



```

        <tr>
        <td><b>Product Category</b></td>
            <td><select name="productcategory">
                <option value="Running shoes">Running shoes</option>
                <option value="Minimalist shoes">Minimalist shoes</option>
                <option value="Walking shoes">Walking shoes</option>
                <option value="Tennis shoes">Tennis shoes</option>
                <option value="Versatile training shoes">Versatile training
shoes</option>
                <option value="Trail running shoes">Trail running shoes</option>
                <option value="Basketball shoes">Basketball shoes</option>
                <option value="Football boots">Football boots</option>

            </select></td>

        </tr>
        <tr>
        <td><b>Product Name</b></td>
        <td><input type="text" name="productname" ></td>
        </tr>
        <tr>
        <td><b>Product Price Rs</b></td>
        <td><input type="number" name="productprice" >/-</td>
        </tr>
        <tr>
        <td><b>Product Description</b></td>
        <td><input type="text" name="productdescription" ></td>
        </tr>

    </table>
    <input type="hidden" name="productid"
value=<%=request.getParameter("productid") %> >

    <br><input type="submit" value="Update Product">
</form>

</div>

</body>
</html>

```

updatesucesshome.jsp

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Home</title>
</head>
<body style="background:url(picer.jpg) no-repeat;background-size:90% auto; ">

    <jsp:include page="home.jsp"></jsp:include>

    <span style="color:green"><h4 align="center">updated sucessfully</h4></span>
</body>
</html>

```


productbyCategory.jsp

```
<%@page import="com.example.demo.*"%>
<%@page import="java.util.*"%>
<%@ page import="com.example.*"%>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>product category</title>
</head>
<body style="background:url(picer.jpg) no-repeat;background-size:90% auto;" >

        List<product> list=(List<product>)request.getAttribute("list");

    %>


    <div align="center">

            <i><h1>SportyShoes</h1></i>
            <h4 align="right"> - developed by S.V. SAI KUMAR</h4><br>

                <h3 align="right"><a href='userHome.jsp' style="text-decoration: none">Home</a>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
                <a href='showCart' style="text-decoration: none">Cart</a>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
&nbsp;&nbsp;&nbsp;&nbsp;&~
&nbsp;&nbsp;&~
                <a href='Logout' style="text-decoration: none">Logout</a></h3>

                <br><br>

        <table cellpadding = "5" cellspacing = "5" style=font-size:25px >
            <tr><th>Product Id</th><th>Product Name</th><th>Product Description</th><th>Product price</th><th>Select Size</th></tr>
            <%for(product p:list){ %>
                <tr style="text-align: center"><td><%=p.getProductid()%></td><td><%=p.getProductname() %></td><td><%=p.getProductdescription()%></td><td><%=p.getProductprice()%></td>
                <td><select name="size">
                    <option value="5">5</option>
                    <option value="6">6</option>
                    <option value="7">7</option>
                    <option value="8">8</option>
                    <option value="9">9</option>
                    <option value="10">10</option>
                    <option value="11">11</option>
                    <option value="12">12</option>

                        </select></td>
                    <td><form action ="addtocart">
                            <input type="hidden" id="productid" name="productid" value=<%=p.getProductid() %>><input type="hidden" id="productcategory" name="productcategory" value=<%=p.getProductcategory()%>>
                                <input type="hidden" id="productname" name="productname" value=<%=p.getProductname() %>><input type="hidden" id="productprice" name="productprice" value=<%=p.getProductprice()%>>
                                    <input type="hidden" id="productsized" name="productsized" value=<%=request.getParameter("size") %>>
                                        <input type="submit" name="add" value="Add to Cart" ></form></td>
                        </tr>
                    <%}%>
                </table>

    </div>
```

```
<%@page import="com.example.demo.*"%>
<%@page import="com.example.demo.cartrepository"%>
<%@page import="com.example.demo.product"%>
<%@page import="com.example.demo.mainController"%>
<%@page import="org.springframework.security.core.context.SecurityContextHolder"%>
<%@page import="org.springframework.security.core.Authentication"%>

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

[illegible]

</body>

</html>

failpayshowcart.jsp

```
<%@page import="com.example.demo.*"%>
<%@page import="com.example.demo.cartrepositry"%>
<%@page import="com.example.demo.product"%>
<%@page import="com.example.demo.mainController"%>
<%@page import="org.springframework.security.core.context.SecurityContextHolder"%>
<%@page import="org.springframework.security.core.Authentication"%>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@page import="java.util.*"%>
<%@ page import="com.example.*"%>

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>cart</title>
</head>
<body style="background:url(picer.jpg) no-repeat;background-size:90% auto;" >

        Authentication auth = 
SecurityContextHolder.getContext().getAuthentication();
String username = auth.getName();

List<cart> list=(List<cart>)request.getAttribute("list");

double totalprice =0;

%>



<i><h1>SportyShoes</h1></i>
        <h4 align="right"> - developed by S.V. SAI KUMAR</h4><br>

        <h3 align="right"><a href='userHome.jsp' style="text-decoration: none">Home</a>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
        <a href='logout' style="text-decoration: none">Logout</a></h3>
        <h3 align="left">Hai <%=username %></h3>
        <h2 align="center">Your Cart</h2>


        <table cellpadding ="5" cellspacing ="5" style=font-size:20px >
            <tr><th>Product Id</th><th>Product Name</th><th>Select Size</th><th>Product price</th></tr>
            <%for(cart c:list){ %>
                <tr style="text-align: center"><td><%=c.getProductid() %></td><td><%=c.getProductname() %></td>
                    <td><select name="size">
                        <option value="5">5</option>
                        <option value="6">6</option>
                        <option value="7">7</option>
                        <option value="8">8</option>
                        <option value="9">9</option>
                        <option value="10">10</option>
                        <option value="11">11</option>
                        <option value="12">12</option>

                    </select></td>
                    <td><%=c.getProductprice()%></td>
                    <td><form action = "removefromcart"><input type="hidden" id="cartid" name="cartid" value=<%=c.getCartid() %>><input type="submit" value="Remove from Cart" ></form></td></tr>
                    <%totalprice+=Double.parseDouble(c.getProductprice()); %>

                <%}>

            </table><br>
            <h3>Total Price = <%=totalprice %></h3>
            <form action="payment.jsp"><input type="hidden" id="totalprice" name="totalprice" value=<%=totalprice %>><input style=font-size:20px type="submit" value=" Buy " ></form><br><br><br>


```

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page import="java.util.*" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Payment GateWay</title>
</head>
<body style="background:url(picer.jpg) no-repeat;background-size:90% auto; ">
    <%
```

[illegible]

Application.properties

server.port=8085

#jpa-hibernate

spring.jpa.hibernate.dialect=org.hibernate.dialect.MySQLDialect

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

#data source

spring.datasource.driver-class-name=com.mysql.jdbc.Driver

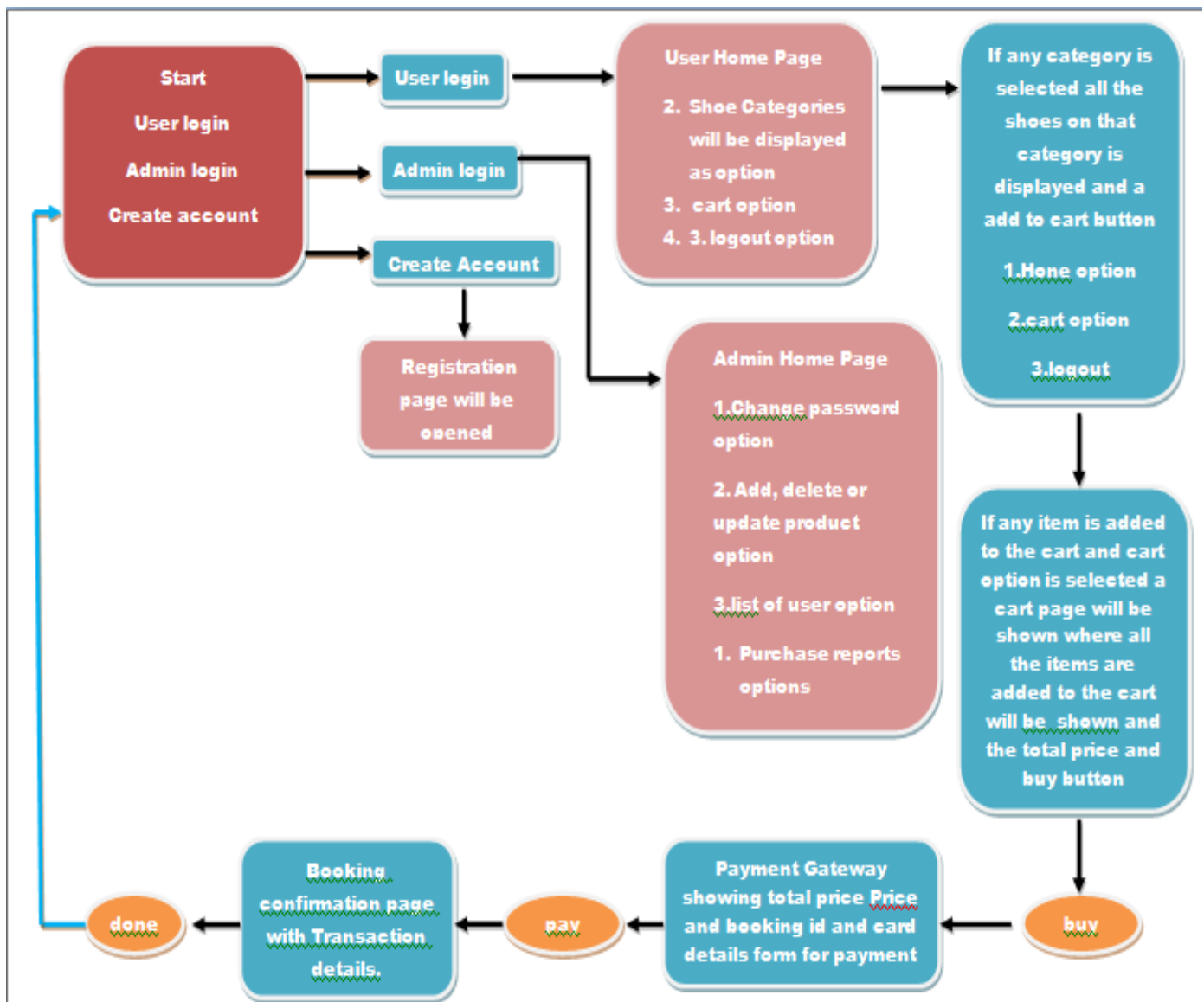
spring.datasource.url=jdbc:mysql://localhost:3306/db7

spring.datasource.username=root

spring.datasource.password=root

Algorithms and flowcharts of the application

Flow chart:



1. Home Page



SportyShoes

- 1

Welcome, Shop According to Your Sports



Admin Signin

User Signin

[new here? Create Account](#)



Activate
Go to Settir

2. On clicking create account user registration form will open.



SportyShoes

- devlop

Enter Details



Name	R Bharath
Email	bharath@gmail.com
Phone no	8985843407
Address	11-1-13, srikakulam
Pincode	532001
Username	Bharath
Password

Register



Activate Win
Go to Settings to

3. It shows confirm page as updated successfully.



SportyShoes

-1

Welcome, Shop According to Your Sports



Admin Signin

User Signin

[new here? Create Account](#)

updated sucessfully



4. In database Mysql password has been changed

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use db7;

Database changed

mysql> show tables;

```
+-----+
| Tables_in_db7 |
+-----+
| cart           |
| hibernate_sequence |
| product       |
| users         |
+-----+
```

4 rows in set (0.02 sec)

mysql> select * from users;

userid	adress	contactno	enabled	name	password	pincode	role	useremail	username
1	NULL	NULL	0x01	NULL	admin	NULL	ADMIN	NULL	admin
2	srikurmam	9502646206	0x01	srikurmam	kumar	532404	USER	ppraveen62.ppk@gmail.com	praveen06
3	Hyderabad	999577922	0x01	Hyderabad	12345	54332	USER	rk@gmail.com	ravi98

3 rows in set (0.00 sec)

mysql> select * from users;

userid	adress	contactno	enabled	name	password	pincode	role	useremail	username
1	NULL	NULL	0x01	NULL	admin	NULL	ADMIN	NULL	admin
2	srikurmam	9502646206	0x01	srikurmam	kumar	532404	USER	ppraveen62.ppk@gmail.com	praveen06
3	Hyderabad	999577922	0x01	Hyderabad	12345	54332	USER	rk@gmail.com	ravi98
4	11-1-13,srikakulam	8985843407	0x01	11-1-13,srikakulam	1432	532001	USER	bharath@gmail.com	Bharath
5	11-1-13,srikakulam	8985843407	0x01	11-1-13,srikakulam	1432	532001	USER	bharath@gmail.com	Bharath

5 rows in set (0.00 sec)

Activate Windows

Go to Settings to activate Windows

5. Trying signin again with changed password.

Please sign in

Sign in

Activate Windows

6.Admin home page is opened.

[Cart](#) [Logout](#)

Welcome to Sporty Shoes



Shop According to your Sprort

Running shoes

Minimalist Shoes

Walking Shoes

Tennis Shoes

Versatile training shoes



Activate Windows
Go to Settings to activate Windows.

**7.clicked on running shoes all the running shoes are displayed,
selected shoes are added to the cart**

Product Id	Product Name	Product Description	Product price	Select Size	
12131	Adidas Men?s Glowrun Reflective M	This shoe has high-density printing with a gradient effect which increases its glamor.	2200	5 ▼	Add to Cart
124355	Adidas Men?s Dasher M Running Shoe	The great cushioning, antibacterial comfort, smoothness, and superb grip makes it a must buying shoe.	2589	5 ▼	Add to Cart
13241	Asics Men?s Road Running Shoe	Asics Men?s Road Running Shoe is a perfect match for those who want light shoes as it weighs 160 gm only	1819	5 ▼	Add to Cart
13234	Puma Men?s Zeta Running Shoes	performance, good material then Puma Men?s Zeta Running Shoes is the best option for you	2469	5 ▼	Add to Cart

8.if clicked and added to cart ,it displays confirmation of adding

Item added to cart Successfully

Shop According to your Sport



Running shoes

Minimalist Shoes

Walking Shoes

Tennis Shoes

Versatile training shoes

Trail running shoes



Activate Windows
Go to Settings to activate Windows.

Product Id	Product Name	Product Description	Product price	Select Size	
12131	Adidas Men?s Glowrun Reflective M	This shoe has high-density printing with a gradient effect which increases its glamor.	2200	5 ▼	Add to Cart
124355	Adidas Men?s Dasher M Running Shoe	The great cushioning, antibacterial comfort, smoothness, and superb grip makes it a must buying shoe.	2589	5 ▼	Add to Cart
13241	Asics Men?s Road Running Shoe	Asics Men?s Road Running Shoe is a perfect match for those who want light shoes as it weighs 160 gm only	1819	10 ▼	Add to Cart
13234	Puma Men?s Zeta Running Shoes	performance, good material then Puma Men?s Zeta Running Shoes is the best option for you	2469	5 ▼	Add to Cart

Item added to cart Successfully

Shop According to your Sprort



Running shoes

Minimalist Shoes

Walking Shoes

Tennis Shoes

Versatile training shoes

Trail running shoes

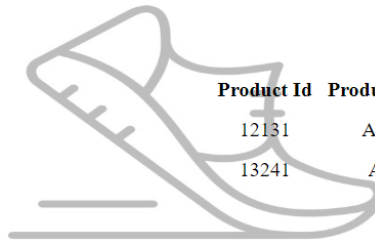


Activate Windows
Go to Settings to activate Windows.

9. Clicking on cart

Hai Bharath

Your Cart



Product Id	Product Name	Select Size	Product price
------------	--------------	-------------	---------------

12131	Adidas	10	2200
-------	--------	----	------

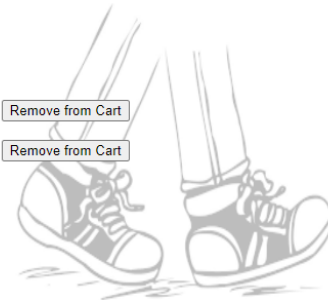
13241	Asics	10	1819
-------	-------	----	------

Total Price = 4019.0

Buy

Remove from Cart

Remove from Cart

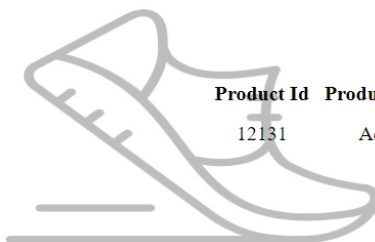


Activate Windows
Go to Settings to activate Windows.

10. removed a item ,using remove from cart button

Hai Bharath

Your Cart



Product Id	Product Name	Select Size	Product price
------------	--------------	-------------	---------------

12131	Adidas	5	2200
-------	--------	---	------

Total Price = 2200.0

Buy

Remove from Cart




Activate Windows
Go to Settings to activate Windows.

11. on clicking buy option ,payment page will be displayed

[Home](#) [Cart](#) [Logout](#)

SportyShoes.Pay

(A Secure Payment Gateway.)




Company SportyShoes
Order Number 120275
Total Amount Rs 2200.0 /-

Card Payment

Card Number

Expiry Date

Password



Activate Windows
Go to Settings to activate Windows.

[Home](#) [Cart](#) [Logout](#)

SportyShoes.Pay

(A Secure Payment Gateway.)



Company SportyShoes
Order Number 120275
Total Amount Rs 2200.0 /-

Card Payment

Card Number


Expiry Date


Password



Activate Windows
Go to Settings to activate Windows.

12. On giving details and clicking on pay, successful transaction page will be shown





 **Payment Successful !**

Thank you! Your order has been placed.

Booking ID : 120275 | Transaction ID : 724910448 | Total Amount Paid : 2200.0

Your Order details have been sent to your mail.





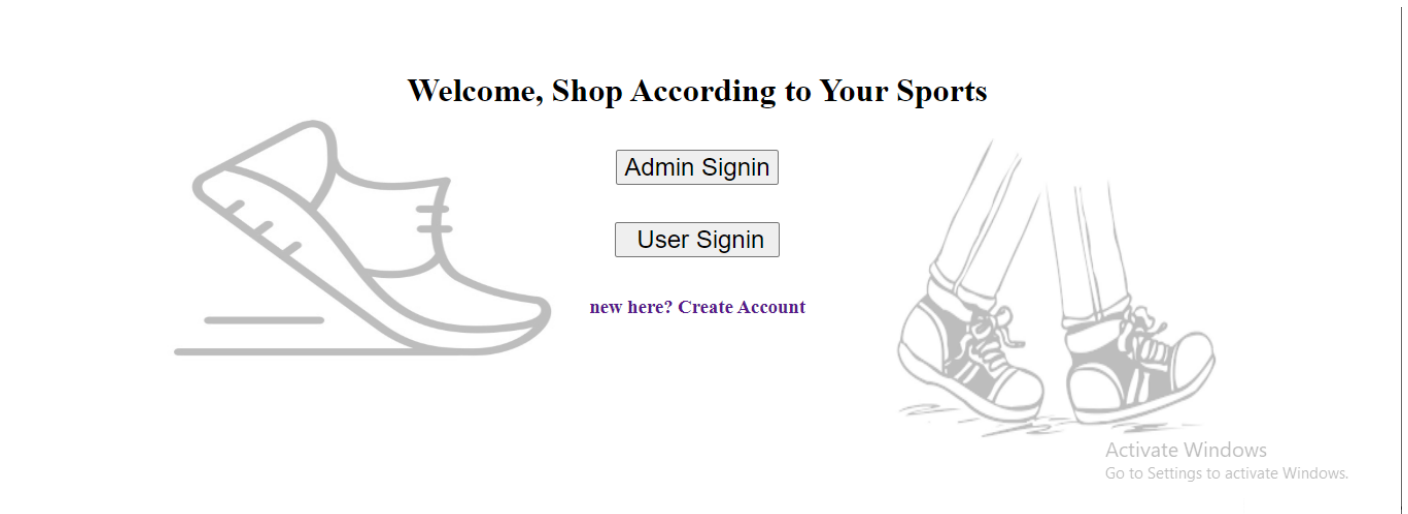
Activate Windows
Go to Settings to activate Windows.

13.Lets check admin part ,

Use username-admin

Password-admin

As it was updated on database by me.



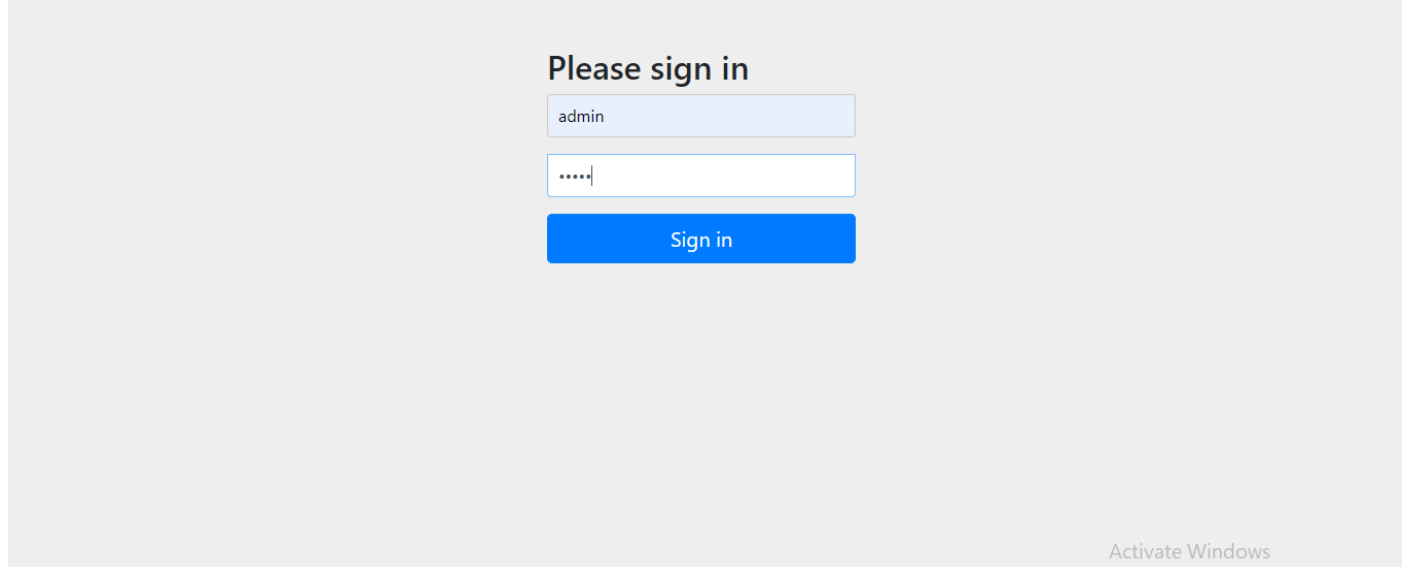
```
mysql> select * from users;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 4 | 11-1-13,srikakulam | 8985843407 | 0x01 | 11-1-13,srikakulam | 1432 | 532001 | USER | bharath@gmail.com | Bharath |
| 5 | 11-1-13,srikakulam | 8985843407 | 0x01 | 11-1-13,srikakulam | 1432 | 532001 | USER | bharath@gmail.com | Bharath |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> delete from users where userid=5;
Query OK, 1 row affected (0.09 sec)

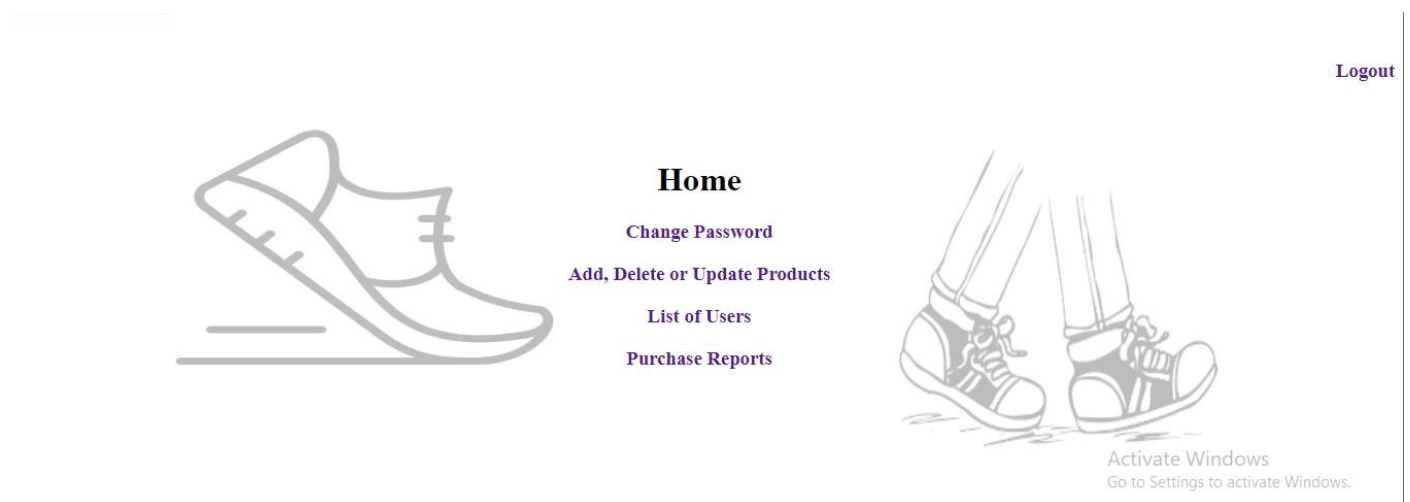
mysql> select * from users;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | NULL | NULL | 0x01 | NULL | admin | NULL | ADMIN | NULL | admin |
| 2 | srikurmam | 9502646206 | 0x01 | srikurmam | kumar | 532404 | USER | ppraveen62.ppk@gmail.com | prav |
| 3 | Hyderabad | 999577922 | 0x01 | Hyderabad | 12345 | 54332 | USER | rk@gmail.com | ravi |
| 4 | 11-1-13,srikakulam | 8985843407 | 0x01 | 11-1-13,srikakulam | 1432 | 532001 | USER | bharath@gmail.com | Bharath |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from cart;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| cartid | dateofpurchase | orderno | payment | productcategory | productid | productname | productprice | username |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

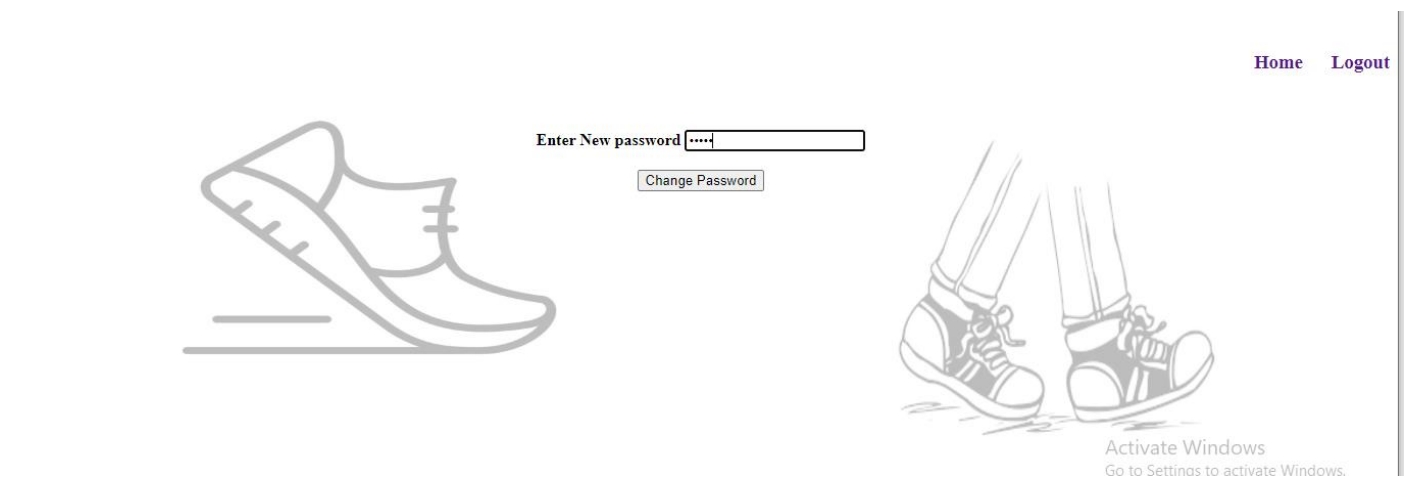
```



14. On successful sign in admin home page will be displayed.



15. on clicking change password page will be displayed and new password is given



16. Confirmation page will be displayed as updated successfully



As in database will be changed

```
mysql> select * from users;
```

userid	adress	contactno	enabled	name	password	pincode	role
1	NULL	NULL	0x01	NULL	12345	NULL	ADMIN
2	srikurmam	9502646206	0x01	srikurmam	kumar	532404	USER
3	Hyderabad	999577922	0x01	Hyderabad	12345	54332	USER
4	11-1-13,srikakulam	8985843407	0x01	11-1-13,srikakulam	1432	532001	USER

```
4 rows in set (0.00 sec)

mysql>
```

17.particular searched username details will be displayed

[Home](#)

User Id	Name	UserName	Email	Address	Contact No	Pincode
4	11-1-13,srikakulam	Bharath	bharath@gmail.com	11-1-13,srikakulam	8985843407	532001

18.coming to home page and clicking on purchase reports

Home I

Purchased Date	Order Id	product Id	Product Category	Product price	Customer Username
2022-10-23 00:00:00.0	121384	1233	Minimalist	2000	2000
2022-10-23 00:00:00.0	121384	442112	Versatile	3422	3422
2022-10-23 00:00:00.0	121036	12344	Walking	3009	3009
2022-10-24 00:00:00.0	120275	12131	Running	2200	2200

19.on home page clicking on add ,delete or update

Home Logout

Add Product

Product Id	Product Category	Product Name	Product price	Product Description		
12131	Running shoes	Adidas Men's Glowrun Reflective M	2200	This shoe has high-density printing with a gradient effect which increases its glamor.	Edit	Delete
124355	Running shoes	Adidas Men's Dasher M Running Shoe	2589	The great cushioning, antibacterial comfort, smoothness, and superb grip makes it a must buying shoe.	Edit	Delete
12344	Walking shoes	Nike Men's Flex Experience RN 8 Running Shoes	3009	With its easy lace-up fastening, you can easily achieve the desires snug fit.	Edit	Delete
13241	Running shoes	Asics Men's Road Running Shoe	1819	Asics Men's Road Running Shoe is a perfect match for those who want light shoes as it weighs 160 gm only	Edit	Delete
13234	Running shoes	Puma Men's Zeta Running Shoes	2469	performance, good material then Puma Men's Zeta Running Shoes is the best option for you	Edit	Delete

20.on clicking on add product giving product details and giving add option

[Home](#) [Logout](#)



Enter Details

Product Id

10122

Product Category

Walking shoes

Product Name

Nike test 2

Product Price Rs

2010

Product Description

hard souls and strong grip up

Add Product




Activate Windows
Go to Settings to activate Windows.

It will be added to list

112352	Trail running shoes	Saucony Xodus Ultra	3000	4.5mm lugs and rockplate Roomy toebox Gaiter compatible	Edit	Delete
35436	Basketball shoes	Nike Cosmic Unity 2	3000	lighter weight, better grip, and more overall response.	Edit	Delete
23425	Basketball shoes	Li-Ning Way Of Wade 10	2900	amazing cushioning and combines with a full-length carbon fiber plate for best-in-class energy return.	Edit	Delete
42324	Basketball shoes	adidas Trae Young 1	4000	The multi-directional ridged outsole sticks you to the hardwood while the ergonomic midsole offers support and comfort where you need it.	Edit	Delete
43231	Football boots	adidas COPA Sense	3500	this new colour-up was certainly eye catching	Edit	Delete
24533	Football boots	adidas Predator Edge	3600	hard souls and strong grip	Edit	Delete
10122	Walking shoes	Nike test 2	2010	hard souls and strong grip updated	Edit	Delete

21.on clicking edit option a update form will be opened .

[Home](#) [Logout](#)



Enter Details

Product Category

Running shoes

Product Name

Nike test 2


Product Price Rs

3000

Product Description

hard souls and strong grips

Update Product



Activate Windows
Go to Settings to activate Windows.

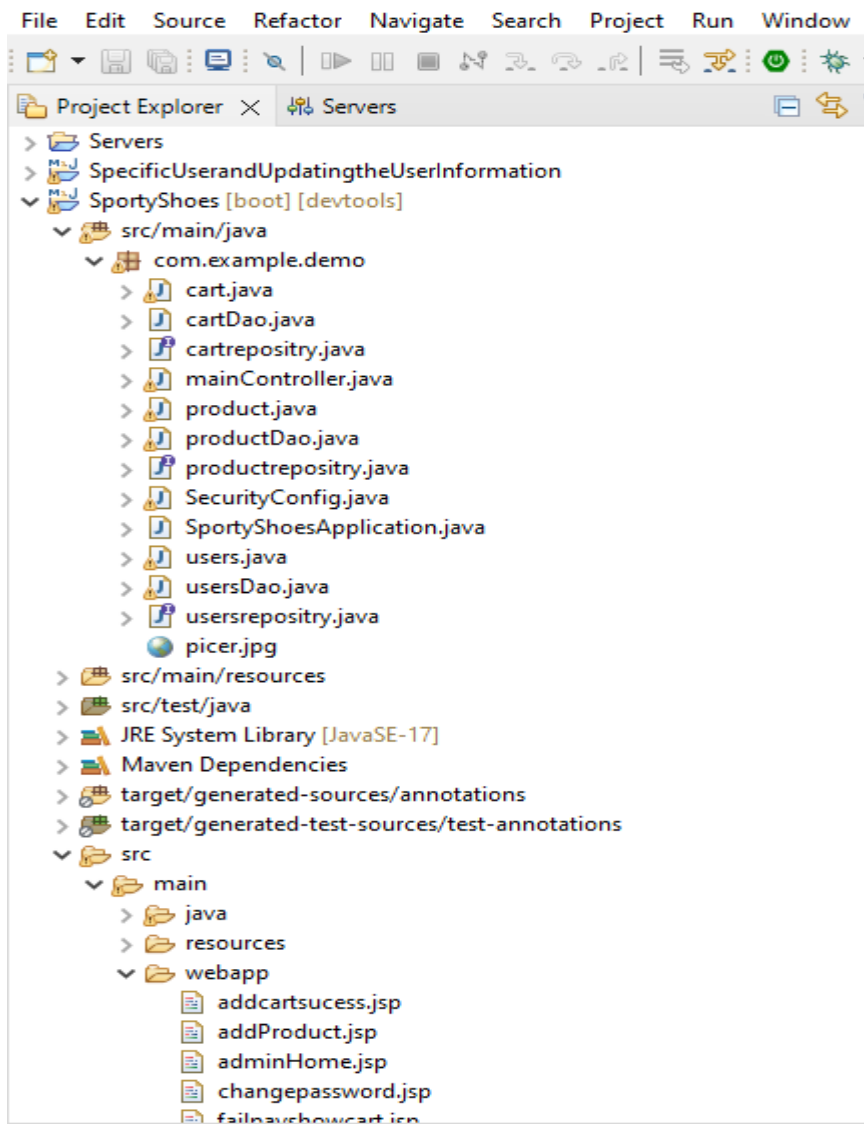
Where I changed the price only it was updated.


































112352	Trail running shoes	Saucony Xodus Ultra	3000	4.5mm lugs and rockplate Roomy toebox Gaiter compatible	Edit	Delete
35436	Basketball shoes	Nike Cosmic Unity 2	3000	lighter weight, better grip, and more overall response.	Edit	Delete
23425	Basketball shoes	Li-Ning Way Of Wade 10	2900	amazing cushioning and combines with a full-length carbon fiber plate for best-in-class energy return.	Edit	Delete
42324	Basketball shoes	adidas Trae Young 1	4000	The multi-directional ridged outsole sticks you to the hardwood while the ergonomic midsole offers support and comfort where you need it.	Edit	Delete
43231	Football boots	adidas COPA Sense	3500	this new colour-up was certainly eye catching	Edit	Delete
24533	Football boots	adidas Predator Edge	3600	hard souls and strong grip	Edit	Delete
10122	Running shoes	Nike test 2	3000	hard souls and strong grips	Edit	Delete

22. then I clicked on delete button of the updated item ,it will be deleted from the list

2434	shoes	Nike Metcon 7	3009	True to size Stable Awesome grip Very comfy	Edit	Delete
11344	Versatile training shoes	Nano X1	3400	More versatile, less crossfit-specific Upper hugs the foot	Edit	Delete
442112	Versatile training shoes	Adidas Powerlift 4	3422	Helps to improve lifting form Versatile for other WOD exercises	Edit	Delete
723332	Versatile training shoes	Nike's Metcons	2500	Excellent for agility workouts Lighter than other Metcons	Edit	Delete
4365	Trail running shoes	Salomon Pulsar Trail	3900	Excellent traction on both wet and dry trails	Edit	Delete
2134	Trail running shoes	Saucony Peregrine 12	2300	Saucony's Peregrine line has long been a speedy trail shoe that works well for many types of runners in a wide variety of conditions.	Edit	Delete
112352	Trail running shoes	Saucony Xodus Ultra	3000	4.5mm lugs and rockplate Roomy toebox Gaiter compatible	Edit	Delete
35436	Basketball shoes	Nike Cosmic Unity 2	3000	lighter weight, better grip, and more overall response.	Edit	Delete
23425	Basketball shoes	Li-Ning Way Of Wade 10	2900	amazing cushioning and combines with a full-length carbon fiber plate for best-in-class energy return.	Edit	Delete
42324	Basketball shoes	adidas Trae Young 1	4000	The multi-directional ridged outsole sticks you to the hardwood while the ergonomic midsole offers support and comfort where you need it.	Edit	Delete
43231	Football boots	adidas COPA Sense	3500	this new colour-up was certainly eye catching	Edit	Delete
24533	Football boots	adidas Predator Edge	3600	hard souls and strong grip	Edit	Delete

Files



- >  Maven Dependencies
- >  target/generated-sources/annotations
- >  target/generated-test-sources/test-annotations
- ▼  src
 - ▼  main
 - >  java
 - >  resources
 - ▼  webapp
 -  addcartsucess.jsp
 -  addProduct.jsp
 -  adminHome.jsp
 -  changepassword.jsp
 -  failpayshowcart.jsp
 -  home.jsp
 -  paid.jsp
 -  payment.jsp
 -  picer.jpg
 -  productbyCategory.jsp
 -  productmodify.jsp
 -  purchaseReports.jsp
 -  showcart.jsp
 -  updateProduct.jsp
 -  updatesucesshome.jsp
 -  userHome.jsp
 -  userlist.jsp
 -  userlistsearch.jsp
 -  userRegistration.jsp
 - >  test
- >  target
-  HELP.md
-  mvnw
-  mvnw.cmd
-  pom.xml

List of core concept used:

Spring Boot , Spring Security, Java , Jsp, Jpa, Jdbc, Hibernate, MySQL database, Collections, and sorting techniques .

Links to the GitHub repository to verify the project completion

https://github.com/Svsai97/SportyShoes_Project3.git

Conclusion:

The E-commerce Website for Sporty Shoes is prepared according to the provided flow and features of the websites and everything followed according to the instructions.