Welcome to Sporty Shoes Portal

Sporty shoes Ecommerce portal

Developer:Sayra Bano

Sporty Shoes Portal

Project Specification and Sprint Work

Version History

Author	**Sayra Bano**
Date	**08-Nov-2021**
Version	**1.0**
Purpose	**Project Specification and Sprint Work**

Content

1.GitHub Link

• Application Code and Version can be tracked in:

https://github.com/sayrabano/ProjectPhase3

2.Modules in the Project

- Module of the Projects
  I. First Module for Admin • The first option is Homepage for the Administration Login.
  II. After Login Page will redirect to AdminDashboard where admin can.
  III. Manage the products in the store including categorizing them.
  IV. Browse the list of users who have signed up and able to search users.
  V. Able to See purchase reports filtered by date and category.
  VI. Admin can Change his password.

3.Sprint Planning

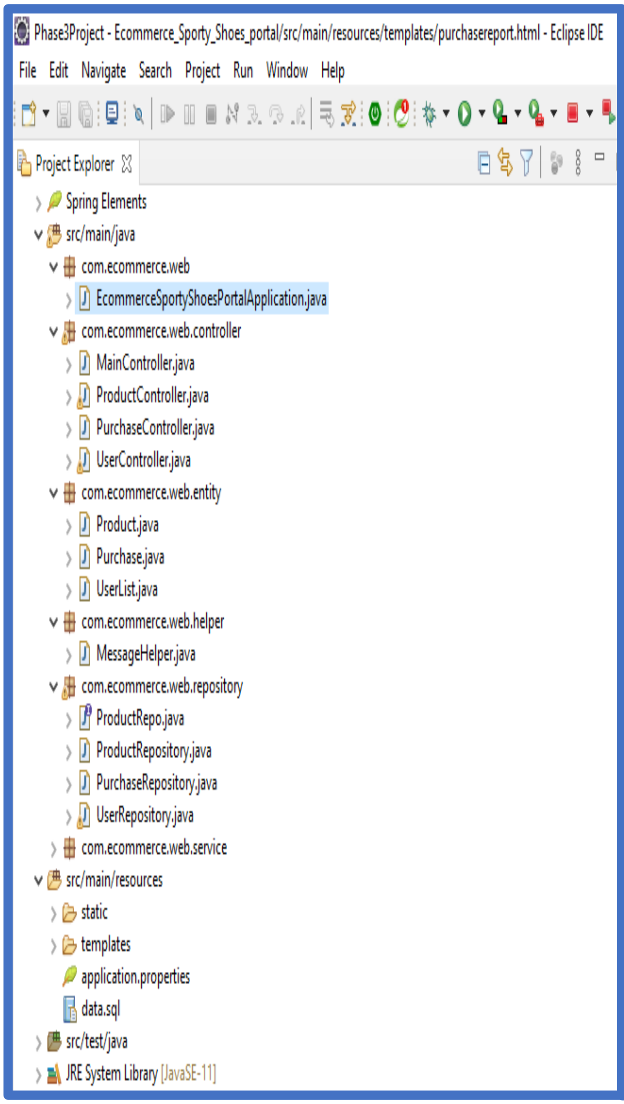
S. No	Sprint Number	Modules Covered
1.	• Sprint 1	– Home page /Module – Admin Login page / Module – Admin Dashboard / Module – Change Password page/ Module – Logout page / Module
2.	• Sprint 2	– Manage Product Category -wise/ Module – Delete Specific Product/Module – Add Product/ Module – Update Product / Module – List of users and search users/ Module – List of Purchase Report and search / Module

4. Technologies and Concepts

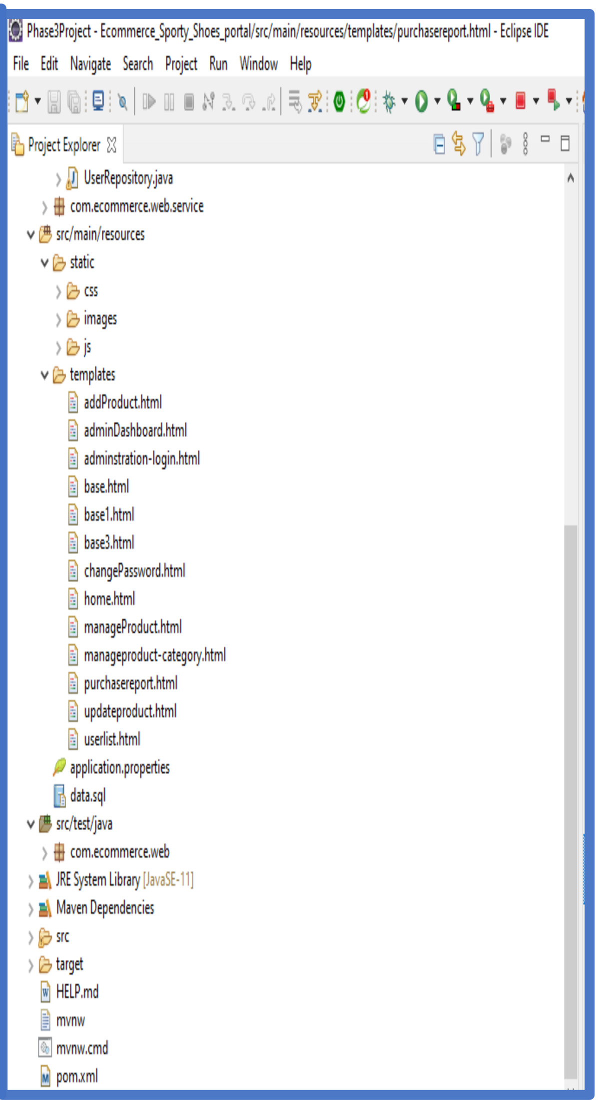
1. Eclipse Ide
2. Spring Boot
3. Core Java
4. Conditional Statement
5. Modularity
6. Naming Standard
7. Exception Handling
8. HTML
9. CSS
10. Bootstrap
11. Control Structure
12. Object Orientation Programming
13. Thyme leaf

5.Project Code and Programming Details

5.1. Folder Structure


Phase3Project - Ecommerce_Sporty_Shoes_portal/src/main/resources/templates/purchasereport.html - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Project Explorer

```
> Spring Elements
v src/main/java
  v com.ecommerce.web
    > EcommerceSportyShoesPortalApplication.java
  v com.ecommerce.web.controller
    > MainController.java
    > ProductController.java
    > PurchaseController.java
    > UserController.java
  v com.ecommerce.web.entity
    > Product.java
    > Purchase.java
    > UserList.java
  v com.ecommerce.web.helper
    > MessageHelper.java
  v com.ecommerce.web.repository
    > ProductRepo.java
    > ProductRepository.java
    > PurchaseRepository.java
    > UserRepository.java
  > com.ecommerce.web.service
v src/main/resources
  > static
  > templates
    application.properties
    data.sql
> src/test/java
> JRE System Library [JavaSE-11]
```


Phase3Project - Ecommerce_Sporty_Shoes_portal/src/main/resources/templates/purchasereport.html - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Project Explorer

```
    > UserRepository.java
  > com.ecommerce.web.service
v src/main/resources
  v static
    > css
    > images
    > js
  v templates
      addProduct.html
      adminDashboard.html
      adminstration-login.html
      base.html
      base1.html
      base3.html
      changePassword.html
      home.html
      manageProduct.html
      manageproduct-category.html
      purchasereport.html
      updateproduct.html
      userlist.html
    application.properties
    data.sql
v src/test/java
  > com.ecommerce.web
> JRE System Library [JavaSE-11]
> Maven Dependencies
> src
> target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

5.2. Creating new project in eclipse

There are many ways of Creating Project in eclipse. You can create project using with Spring boot Starter project and Maven.

Steps for creating Spring Boot Starter project in eclipse: -

- Open Eclipse.
- Click on File choose Spring Boot Starter and add dependencies.
- [Right click] on src/main/java folder and create a package with name Controller, Repository, Entity and required one.
- [Right click] on src/main/java resources folder than choose template folder and create html pages
- [Right click] on src/main/java resources folder than choose static folder and create CSS, JS and images folder.
- [Right click] on src/main/java resources folder and create data.sql file for storing data in database.
- Use application. Properties for configuration database requirements.

5.3. com.ecommerce.web

5.3.1. EcommerceSportyShoesPortalApplication

```
package com.ecommerce.web;


import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

import com.ecommerce.web.entity.Product;
import com.ecommerce.web.repository.ProductRepository;
```

```
@SpringBootApplication
public class EcommerceSportyShoesPortalApplication implements
CommandLineRunner {


public static void main(String[] args) {

SpringApplication.run(EcommerceSportyShoesPortalApplication.class, args);
}

//Injecting Dependency------------------>
@Autowired
ProductRepository<Product> repo;


//inserting product into database---------------------->


@Override
public void run(String... args) throws Exception {
// TODO Auto-generated method stub

repo.addProduct(new Product(1,"Soccer Shoes","Nike",3000,6,"Male"));
repo.addProduct(new Product(2,"Soccer Shoes","Adidas",2000,5,"Female"));
repo.addProduct(new Product(3,"Soccer Shoes","Reebok",3000,7,"Male"));

repo.addProduct(new Product(4,"Running Shoes","Nike",1000,6,"Female"));
repo.addProduct(new Product(5,"Running
Shoes","Adidas",2000,6,"Female"));
repo.addProduct(new Product(6,"Running Shoes","Reebok",3000,7,"Male"));

repo.addProduct(new Product(7,"Golf Shoes","Nike",2000,6,"Female"));
repo.addProduct(new Product(8,"Golf Shoes","Puma",2000,4,"Female"));
repo.addProduct(new Product(9,"Golf Shoes","Reebok",3000,7,"Male"));


repo.addProduct(new Product(10,"High Top
Shoes","Puma",1000,6,"Female"));
repo.addProduct(new Product(11,"High Top Shoes","Adidas",1000,4,"Male"));
repo.addProduct(new Product(12,"High Top Shoes","Red
Tap",2500,7,"Male"));


repo.addProduct(new Product(13,"Hiking Boots Shoes","Red
Tap",1000,4,"Male"));
repo.addProduct(new Product(14,"Hiking Boots
Shoes","Adidas",3000,5,"Female"));
repo.addProduct(new Product(15,"Hiking Boots
Shoes","Reebok",3000,7,"Male"));
}


}
```

5.4. com.ecommerce.web. controller

5.4.1. Main Controller

```
package com.ecommerce.web. controller;

import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;

import com.ecommerce.web.helper.MessageHelper;
import com.ecommerce.web.service.LoginService;

@Controller

public class MainController {

//Dependency Injection------------>
@Autowired
LoginService service;



//Home handler--------------->
@RequestMapping("/")
public String home(Model model) {
model.addAttribute("title", "Home - SportyShoes.com");
return "home";
}
```

```
//Home handler--------------->
@RequestMapping("/home")
public String homePage(Model model) {
model.addAttribute("title", "Home - SportyShoes.com");
return "home";
}


//Admin Login handler--------------->
@RequestMapping("/admin-login")
public String administrationLogin(Model model ) {
model.addAttribute("title", "AdminLogin - SportyShoes.com");
return "adminstration-login";
}


//AdminDashboard handler--------------->
@PostMapping("/adminDashboard")
public String dashboard(ModelMap model,@RequestParam String
email,@RequestParam String password,HttpSession s) {




model.put("title", "AdminDashboard - SportyShoes.com");

boolean isvalidAdmin = service.validateAdmin(email, password);


if(!isvalidAdmin) {

s.setAttribute("message", new MessageHelper("Oops Invalid Credential
!","error"));
return "adminstration-login";
}
else {
model.put("email",email);
model.put("password",password);


System.out.println("Admin password : "+ password);
System.out.println("Admin Email Address : " +email);

model.put("email","admin@sportyshoes.com");
return "adminDashboard";
}



}

//change password handler

@RequestMapping(value="/changePassword")
public String changePassword(Model model) {
model.addAttribute("title", "Change Password - SportyShoes.com");
return "changePassword";
}

//Process ChangePassword handler--------------->
@PostMapping("/change")
public String passwordChange(@RequestParam("npass") String
npass,HttpSession session) {


if(npass.equals("")) {
session.setAttribute("message", new MessageHelper("Password can not be
empty !","success"));



return "changePassword";
}
LoginService.password=npass;
```

```
return "redirect:home";




}

//Logout handler--------------->
@PostMapping(value="/logout")
public String logout() {
return"home";
}




}
```

5.4.2. ProductController

```
package com.ecommerce.web.controller;

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

import javax.servlet.http.HttpSession;

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

import com.ecommerce.web.entity.Product;
import com.ecommerce.web.helper.MessageHelper;
import com.ecommerce.web.repository.ProductRepo;
import com.ecommerce.web.repository.ProductRepository;
@Controller
public class ProductController {

//Dependency Injection-------------->
@Autowired
ProductRepository<Product> repo;


//AdminDashboard handler--------------->
@RequestMapping(value="adminDashboard")
public String dashboardt(Model model) {
model.addAttribute("title", "AdminDashboard - SportyShoes.com");
return "adminDashboard";


}

//Manage Product handler--------------->
@RequestMapping("/manageProduct")
public String manageProduct(Model model) {
model.addAttribute("title", "ManageProduct - SportyShoes.com");
return "manageProduct";

}

//Add Product handler--------------->
@RequestMapping("addproduct")
public String addProduct(Model model) {
model.addAttribute("title", "AddProduct - SportyShoes.com");
return "addProduct";


}

//Process add Product handler--------------->
@PostMapping("/processAddproduct")
public String processAddProduct(@ModelAttribute Product
product,HttpSession session) {

try {


System.out.println("Addedd" +product);
repo.addProduct(product);
session.setAttribute("message", new MessageHelper("Product added
Successfully !","success"));
}catch(Exception e) {
System.out.println("Something is wrong" +e.getMessage());
e.printStackTrace();
//session.setAttribute("message", new MessageHelper("Something went
wrong try again","danger"));
}

return "addproduct";

}

//Managing Product Category wise handler--------------->
@RequestMapping("/manage-cat")
public String manageCategory(Model model) {

return "manageproduct-category";
}

//SoccerShoes category handler--------------->
@RequestMapping(value="/soccer")
```

```
public String getProductSoccer(Model model) {
model.addAttribute("title", "Soccer - SportyShoes.com");
List<Product> product = null;

product = repo.findSoccerShoes();
model.addAttribute("product", product);
System.out.println("List of soccer shoes: " +product);
return "manageproduct-category";



}


//GolfShoes category handler--------------->
@RequestMapping(value="/golf")
public String getProductGolf(Model model) {
model.addAttribute("title", "Golf - SportyShoes.com");
List<Product> product = null;

product= repo.findGolfShoes();
model.addAttribute("product", product);
System.out.println("List of golf shoes:" +product);
return "manageproduct-category";


}

//RunningShoes category handler--------------->
@RequestMapping(value="/running")
public String getProductRunning(Model model) {
model.addAttribute("title", "Running - SportyShoes.com");
List<Product> product = null;

product = repo.findRunningShoes();
model.addAttribute("product", product);
System.out.println( "List of running shoes: "+ product);
return "manageproduct-category";


}

//HighTop Shoes category handler--------------->
@RequestMapping(value="/hightop")
public String getProductHighTop(Model model) {
model.addAttribute("title", "HighTop- SportyShoes.com");
List<Product> product = null;

product = repo.findHighTopShoes();
model.addAttribute("product", product);
System.out.println( "List of high top shoes: "+ product);
return "manageproduct-category";


}
//Boots Shoes category handler--------------->
@RequestMapping(value="/boots")
public String getProductHikingBoots(Model model) {
model.addAttribute("title", "HikingBoots - SportyShoes.com");
List<Product> product = null;

product = repo.findHikingBootsShoes();
model.addAttribute("product", product);
System.out.println("List of hiking boots shoes: "+ product);
return "manageproduct-category";


}


//Dependency Injection---------------->
@Autowired
ProductRepo repository;
```

```
//Manage Product handler--------------->
@GetMapping("/manageProduct/{product_id}")
public String deleteProduct(@PathVariable("product_id") Integer product_id)
{

Optional<Product> productoptional= repository.findById(product_id);
Product product = productoptional.get();
this.repository.delete(product);


return "redirect:/manageProduct/";

}

//Update Product  handler--------------->
@PostMapping("/updateproduct/{product_id}")
public String updateProduct(@PathVariable ("product_id") Integer
product_id,Model model) {
model.addAttribute("title", "Update Product - SportyShoes.com");
Product product  =repository.findById(product_id).get();

model.addAttribute("product",product);

return "updateproduct";
}

//Process Update Product handler--------------->
@PostMapping("/process")
public String proccesUpdateProduct(@ModelAttribute Product
product,HttpSession ss) {
try {

this.repository.save(product);
System.out.println("Product Category: " +product.getProduct_category());
System.out.println("Product Id: " +product.getProduct_id());
ss.setAttribute("message", new MessageHelper("Product Updated
Successfully","success"));
}catch(Exception e) {
e.printStackTrace();

}

return "updateproduct";
}



}
```

5.4.3. PurchaseController

```
package com.ecommerce.web.controller;

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

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.format.annotation.DateTimeFormat;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;

import com.ecommerce.web.entity.Purchase;
import com.ecommerce.web.repository.PurchaseRepository;

@Controller
public class PurchaseController {

//Dependency Injection--------->
@Autowired
PurchaseRepository repo;
//PurchaseReport handler--------------->
@RequestMapping("/purchase")
public String getPurchaseReport(Model model) {
model.addAttribute("title", "Purchase Report - SportyShoes.com");
List<Purchase>purchase= repo.findAll();
model.addAttribute("purchase",purchase);
System.out.println("Purchase Report " +purchase);
return "purchasereport";
}


//PurchaseReport Search handler--------------->
@PostMapping("/search-result")
public String searchPurchaseHistory(Model model,@RequestParam String
category,@RequestParam @DateTimeFormat(pattern="yyyy-MM-dd") Date
date) {

model.addAttribute("title", "SearchResult - SportyShoes.com");
List<Purchase> purchase = repo.findByFilter(category, date);
model.addAttribute("purchase", purchase);
System.out.println("Search result " +purchase );



return "purchasereport";
}
}
```

5.4.4. UserController

```
package com.ecommerce.web.controller;
```

```
import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;

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

import com.ecommerce.web.entity.UserList;
import com.ecommerce.web.repository.UserRepository;

@Controller
public class UserController {

//Dependency Injection ------->
@Autowired
UserRepository repo;



//UserList handler------------>
@RequestMapping(value="/userlist")
public String getUserList(Model model) {
model.addAttribute("title", "UserList - SportyShoes.com");
List<UserList> list = repo.findAll();
model.addAttribute("list",list);
System.out.println("User data"+list);
return "userlist";

}

//User Search handler--------------->
@PostMapping("/user")
public String searchUser(@RequestParam String search,Model model ) {
List<UserList> user = null;
user= repo.findByEmail(search);
model.addAttribute("list",user);

return "userlist";

}

}
```

5.5. com.ecommerce.web.entity

5.5.1. Product

```
package com.ecommerce.web.entity;

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

@Entity
public class Product {
@Id
@GeneratedValue(strategy = GenerationType.AUTO)
private int product_id;

private String product_category;
private String product_brand;

private int product_price;
private int product_size;
private String gender;
public int getProduct_id() {
return product_id;
}
public void setProduct_id(int product_id) {
this.product_id = product_id;
}
public String getProduct_category() {
return product_category;
}
public void setProduct_category(String product_category) {
this.product_category = product_category;
}
public String getProduct_brand() {
```

```
return product_brand;
}
public void setProduct_brand(String product_brand) {
this.product_brand = product_brand;
}
public int getProduct_price() {
return product_price;
}
public void setProduct_price(int product_price) {
this.product_price = product_price;
}
public int getProduct_size() {
return product_size;
}
public void setProduct_size(int product_size) {
this.product_size = product_size;
}
public String getGender() {
return gender;
}
public void setGender(String gender) {
this.gender = gender;
}
public Product(int product_id, String product_category, String product_brand,
int product_price, int product_size,
String gender) {
super();
this.product_id = product_id;
this.product_category = product_category;
this.product_brand = product_brand;
this.product_price = product_price;
this.product_size = product_size;
this.gender = gender;
}
public Product() {
super();
// TODO Auto-generated constructor stub
```

```
}
@Override
public String toString() {
return "Product [product_id=" + product_id + ", product_category=" +
product_category + ", product_brand="
+ product_brand + ", product_price=" + product_price + ", product_size=" +
product_size + ", gender="
+ gender + "]";
}

}
```

5.5.2. Purchase

```
package com.ecommerce.web.entity;

import java.util.Date;

public class Purchase {
```

```
private int product_id;

private String product_category;
private String product_brand;



private int product_price;
private int product_size;

private Date purchase_date;

@Override
public String toString() {
return "Purchase [product_id=" + product_id + ", product_category=" +
product_category + ", product_brand="
+ product_brand +  ", product_price=" + product_price + ", product_size="
+ product_size + ", purchase_date=" + purchase_date + "]";
}

public Purchase(int product_id, String product_category, String
product_brand, int product_price, int product_size,
Date purchase_date) {
super();
this.product_id = product_id;
this.product_category = product_category;
this.product_brand = product_brand;
this.product_price = product_price;
this.product_size = product_size;
this.purchase_date = purchase_date;
}

public Purchase() {
super();
// TODO Auto-generated constructor stub
}

public int getProduct_id() {
```

```
return product_id;
}

public void setProduct_id(int product_id) {
this.product_id = product_id;
}

public String getProduct_category() {
return product_category;
}

public void setProduct_category(String product_category) {
this.product_category = product_category;
}

public String getProduct_brand() {
return product_brand;
}

public void setProduct_brand(String product_brand) {
this.product_brand = product_brand;
}

public int getProduct_price() {
return product_price;
}

public void setProduct_price(int product_price) {
this.product_price = product_price;
}

public int getProduct_size() {
return product_size;
}

public void setProduct_size(int product_size) {
this.product_size = product_size;
```

```
}

public Date getPurchase_date() {
return purchase_date;
}

public void setPurchase_date(Date purchase_date) {
this.purchase_date = purchase_date;
}

}
```

5.5.3. UserList

```
package com.ecommerce.web.entity;



public class UserList {


private int user_id;
private String user_name;

private String user_email;
private String user_location;
public UserList(int user_id, String user_name, String user_email, String
user_location) {
super();
this.user_id = user_id;
this.user_name = user_name;
this.user_email = user_email;
this.user_location = user_location;
}
public UserList() {

}
@Override
public String toString() {
return "UserList [user_id=" + user_id + ", user_name=" + user_name + ",
user_email=" + user_email
+ ", user_location=" + user_location + "]";
}
public int getUser_id() {
return user_id;
}
public void setUser_id(int user_id) {
this.user_id = user_id;
}
public String getUser_name() {
return user_name;
}
public void setUser_name(String user_name) {
this.user_name = user_name;
}
public String getUser_email() {
return user_email;
}
public void setUser_email(String user_email) {
this.user_email = user_email;
}
public String getUser_location() {
return user_location;
}
public void setUser_location(String user_location) {
this.user_location = user_location;
}


}
```

5.6. com.ecommerce.web.repository

5.6.1. ProductRepo

```
package com.ecommerce.web.repository;
//Interface for crud operations in product------>
import org.springframework.data.repository.CrudRepository;

import com.ecommerce.web.entity.Product;

public interface ProductRepo extends CrudRepository<Product, Integer> {

}
```

5.6.2. ProductRepository

```
package com.ecommerce.web.repository;

import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;


import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;

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

import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.jdbc.core.RowMapper;
import org.springframework.stereotype.Repository;
import org.springframework.transaction.annotation.Transactional;

import com.ecommerce.web.entity.Product;



@Repository
@Transactional
public class ProductRepository<product_category> {

//Dependency Injection------->
@Autowired
JdbcTemplate template;


//JdbcTemplate method-------->
public void setTemplate(JdbcTemplate template)
{
this.template = template;
}

class ProductRowMapper implements RowMapper<Product>{


@Override
public Product mapRow(ResultSet rs, int row) throws SQLException
{
Product product = new Product();
product.setProduct_id(rs.getInt(1));
product.setGender(rs.getString(2));
product.setProduct_brand(rs.getString(3));
product.setProduct_category(rs.getString(4));
product.setProduct_price(rs.getInt(5));
product.setProduct_size(rs.getInt(6));
return  product;
}
}

@PersistenceContext
EntityManager entityManager;

public Product addProduct(Product product) {
return entityManager.merge(product);

}

public void save(Product product) {
entityManager.merge(product);
}




public List<Product> findSoccerShoes(){
return template.query("select * from product where
product_category='Soccer Shoes' ", new ProductRowMapper());


}
public List<Product> findRunningShoes(){
return template.query("select * from product where
product_category='Running Shoes' ", new ProductRowMapper());


}
public List<Product> findGolfShoes(){
return template.query("select * from product where product_category='Golf
Shoes' ", new ProductRowMapper());
```

```
}
public List<Product> findHighTopShoes(){
return template.query("select * from product where product_category='High Top Shoes' ", new ProductRowMapper());


}
public List<Product> findHikingBootsShoes(){
return template.query("select * from product where product_category='Hiking Boots Shoes' ", new ProductRowMapper());


}




}
```

5.6.3. PurchaseRepository

```
package com.ecommerce.web.repository;

import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.jdbc.core.RowMapper;
import org.springframework.stereotype.Repository;

import com.ecommerce.web.entity.Purchase;
```

```
@Repository
public class PurchaseRepository {
@Autowired
JdbcTemplate template;



public void setTemplate(JdbcTemplate template)
{
this.template = template;
}


class PurchaseRowMapper implements RowMapper<Purchase>{
@Override
public Purchase mapRow(ResultSet rs, int row) throws SQLException
{
Purchase purchase = new Purchase();
purchase.setProduct_id(rs.getInt(1));
purchase.setProduct_category(rs.getString(2));
purchase.setProduct_brand(rs.getString(3));
purchase.setProduct_price(rs.getInt(4));
purchase.setProduct_size(rs.getInt(5));
purchase.setPurchase_date(rs.getDate(6));


return purchase;
}}
public List<Purchase> findAll(){

return template.query("select * from purchase", new PurchaseRowMapper()
{

});
}



public List<Purchase> findByFilter(String product_category,Date
purchase_date){
SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
String NDate= formatter.format(purchase_date);


return template.query("select * from purchase where
product_category='"+product_category+"'"+"and purchase_date='"+NDate+"'
", new PurchaseRowMapper() {



});
}


}
```

5.6.4. UserRepository

```
package com.ecommerce.web.repository;

import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.jdbc.core.RowMapper;
import org.springframework.stereotype.Repository;


import com.ecommerce.web.entity.UserList;

@Repository

public class UserRepository
{
@Autowired
JdbcTemplate template;



public void setTemplate(JdbcTemplate template)
{
this.template = template;
}

public List<UserList> findAll()
{
return template.query("select * from userlist", new RowMapper<UserList>()
{
public UserList mapRow(ResultSet rs, int row) throws SQLException
{
UserList user = new UserList();
user.setUser_id(rs.getInt(1));
user.setUser_name(rs.getString(2));
user.setUser_email(rs.getString(3));
user.setUser_location(rs.getString(4));

return user;
}
});
}
public List<UserList> findByEmail(String user_email)

{

return template.query("select * from userlist where user_email
='"+user_email+"'", new RowMapper<UserList>()
{
public UserList mapRow(ResultSet rs, int row) throws SQLException
{
UserList user = new UserList();
user.setUser_id(rs.getInt(1));
user.setUser_name(rs.getString(2));
user.setUser_email(rs.getString(3));
user.setUser_location(rs.getString(4));

return user;


}
});





}}
```

5.7. package com.ecommerce.web.service

5.7.1 LoginService

```
package com.ecommerce.web.service;

import org.springframework.stereotype.Service;

@Service
public class LoginService {
//Credentials for admin login-------->
public static String password= "admin";
public static String email= "admin@sportyshoes.com";
public boolean validateAdmin(String email,String password) {

return
email.equals("admin@sportyshoes.com")&&(password.equals(LoginService.password));

}

}
```

5.8.com.ecommerce.web.helper

```
package com.ecommerce.web.helper;
```

5.8. MessageHelper

```
//Helper class for message------->
public class MessageHelper {
        private String content;
        private String type;
        public String getContent() {
                return content;
        }
        public void setContent(String content) {
                this.content = content;
        }
        public String getType() {
                return type;
        }
        public void setType(String type) {
                this.type = type;
        }
        public MessageHelper() {
                super();
                // TODO Auto-generated constructor stub
        }
        public MessageHelper(String content, String type) {
                super();
                this.content = content;
                this.type = type;
        }




}
```

5.9 HTML Pages

5.9.1. Home

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
th:replace="base::layout(~{::section})">
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<style>


</style>
<body>
<section class="banner d-flex justify-content-center align-items-center">


<div class="text-center">
<h1 style="color:white;" class="font-weight-bold">SportyShoes.com</h1>
<h4 style="color:white;">A Company that manufactures and sells sports
Shoes.</h4>

<a th:href="@{/admin-login/}" ><button class="btn btn-primary btn font-
weight-bold"><i class="fas fa-sign-in-alt"></i>Administration
Login</button></a>
</div></section>

</body>
</html>
```

5.9.2. Administration Login

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
th:replace="base::layout(~{::section})">
<head>
<meta charset="ISO-8859-1">
<title>Admin-login : SportyShoes.com</title>
</head>
<body>
<section>
<div th:if="${session.message}" class="container">
<div th:classappend="${'alert-'+session.message.type}" class ="alert text-center" role="alert">
<span th:text="${session.message.content}"></span>
</div>
<th:block th:text="${#session.removeAttribute('message')}"></th:block>
</div>
<div class="container">
<div class="row">
<div class="col-md-6 offset-md-3">
<div class="my-card">
<div class="container text-center">
<img style="width:80px;" alt="" src="" th:src="@{/images/blogger.png}">
</div>
<h1 class="text-center">

Administration Login !!

</h1>



<!-- Form field  -->
<form action="" th:action="@{/adminDashboard}" method="post">



<div class="form-group">
```

```
<label for="exampleInputEmail1">Email address</label>
<input name="email" type="email" class="form-control"
id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Enter
Admin Email" required="required">

</div>
<div class="form-group">
<label for="exampleInputPassword1">Password</label>
<input name="password" type="password" class="form-control"
id="exampleInputPassword1" placeholder=" Enter Admin Password"
required="required" >
</div>

<button type="submit" class="btn btn-primary btn-block">Login</button>


<!--form field  -->



</form>





</div>


</div>


</div>



</div>
```

```
</section>

</body>
</html>
```

5.9.3. Admin Dashboard

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
th:replace="base1::layout(~{::section})">
<head>
<meta charset="ISO-8859-1">
<title>AdminDashboard : SportyShoes.com</title>
</head>
<body>
<section >
<div class="container text-center"><img style="width:100px;" alt="" src=""
th:src="@{/images/backend.png}">
<h2>Welcome to Admin Dashboard</h2>
<h3>You are LoggedIn as: Administration !</h3>
<h3>Your Credential is : <span th:text="${email}"></span> </h3>



</div>



</section>

</body>
</html>
```

5.9.4. Manage Product Category

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
th:replace="base3::layout(~{::section})">

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


<div class="row">

<div class="column">
<div class="card">
<a th:href="@{/soccer}">
<img style="width: 200px;" alt="" src=""
th:src="@{/images/soccer-shoe-150x150.png}">
<h3>Soccer Shoes</h3>
</a>
</div>
</div>
<div class="column">
<div class="card">
<a th:href="@{/golf}">
<img style="width: 200px;" alt="" src=""
th:src="@{/images/golf-shoes-150x150.png}">
<h3>Golf Shoes</h3>
</a>
</div>
</div>0.00
```

```
<div class="column">
<div class="card">
<a href="/running">
<img style="width: 200px;" alt="" src=""
th:src="@{/images/running-shoes-150x150.png}">
<h3>Running Shoes</h3>
</a>
</div>
</div>

<div class="column">
<div class="card">
<a href="/boots">
<img style="width: 200px;" alt="" src=""
th:src="@{/images/hiking-shoes-150x150.png}">
<h3>Hiking Boots Shoes</h3>
</a>
</div>
</div>
<div class="column">
<div class="card">
<a href="/hightop">
<img style="width: 200px;" alt="" src=""
th:src="@{/images/high-tops-150x150.png}">
<h3>High-Top Shoes</h3>
</a>
</div>
</div>

<div class="column">

<div class="card"><a th:href="@{/addproduct}"><button type="submit"
class="btn btn-primary btn-block text-
center">AddProduct</button></a></div>
</div>

</div>
```

```
</section>

</body>
</html>
```

5.9.5. Manage Product

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
th:replace="base3::layout(~{::section})">
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>


</head>
<body>

<section>

<br>
<form method="post" th:action="@{/manage-cat}">
</form><br>
<div class="card">

<h1 class="text-center"></h1>

<table class="table">
<thead class="thead-dark">
```

```
<tr>
<th>Product ID</th>
<th>Product Category</th>
<th>Product Brand</th>
<th>Product Size</th>
<th>Product Price</th>
<th>Gender</th>
<th>Action</th>


</tr>
</thead>
<tbody>




<tr th:each="data : ${product}">
<td class="id"><span th:text="'SSPProduct'
+${data.product_id}">1</span></td>

<td th:text="${data.product_category}">Product Category</td>
<td th:text="${data.product_brand}">Product Brand</td>
<td th:text="${data.product_size}">Product Size</td>
<td th:text="${data.product_price}">Product Price</td>
<td th:text="${data.gender}">Gender</td>
<td><a href="#" th:onclick="|deleteProduct(${data.product_id})|"><button
class="btn btn-danger"><i class="fas fa-trash"></i> Delete</button></a>

<form method="post"
th:action="@{'/updateproduct/'+${data.product_id}}"><br><a><button
class="btn btn-primary"><i class="far fa-
edit"></i>Update</button></a></form>

</td>
```

```
</tr>

</tbody>
</table>
</div>

</section>

</body>
</html>
```

5.9.6 .UserList

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
th:replace="base3::layout(~{::section})">
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

<section>
<br>
<form method="post" th:action="@{/user}">
<input type="text" name="search" id="search" class="search text-center"
placeholder="Enter User Email">
<button type="submit" class="btn btn-primary">Search</button>
</form><br>
<div class="card">

<h1 class="text-center">List of User</h1>

<table class="table">
<thead class="thead-dark">
<tr>
```

```
<th>User ID</th>
<th>User Name</th>
<th>User Email</th>
<th>User Location</th>

</tr>
</thead>
<tbody>
<tr th:each="data : ${list}">
<td class="id"><span th:text="'SSPUSER' +${data.user_id}">1</span></td>

<td><span th:text="${data.user_name}"> User Name </span></td>
<td><span th:text="${data.user_email}"> User Email </span></td>
<td><span th:text="${data.user_location}">User Location </span></td>
</tr>

</tbody>
</table>
</div>

</section>

</body>
</html>
```

5.9.7. PurchaseReport List

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
th:replace="base3::layout(~{::section})">
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

<section>
```

```
<br>
<form method="post" th:action="@{/search-result}">

<label class="lable" for="category">Product Category:</label>
<select id="category" name="category">
<option value="Soccer Shoes">Soccer Shoes</option>
<option value="Golf Shoes">Golf Shoes</option>
<option value="Running Shoes">Running Shoes</option>
<option value="High Top Shoes">High Top Shoes</option>
<option value="Hiking Boots Shoes">Hiking Boots Shoes</option>
</select>
&nbsp;&nbsp;&nbsp;<label for="date">Purchase Date:</label>
<input type="date" name="date" id="date" required>


<button type="submit" class="btn btn-primary">Search</button>
</form>

<br>
<div class="card">

<h1 class="text-center">Purchase Report</h1>

<table class="table">
<thead class="thead-dark">
<tr>
<th>Product ID</th>
<th>Product Category</th>
<th>Product Brand</th>
<th>Product Price</th>
<th>Product Size</th>

<th>Purchase Date</th>

</tr>
</thead>
<tbody>
```

```
<tr th:each="data : ${purchase}">
<td class="id"><span th:text="'SSPProduct'
+${data.product_id}">1</span></td>

<td><span th:text="${data.product_category}"> Product Category</span></td>
<td><span th:text="${data.product_brand}">Product Brand</span></td>
<td><span th:text="${data.product_size}">Product Size </span></td>
<td><span th:text="${data.product_price}">Product Price </span></td>

<td><span th:text="${data.purchase_date}">Purchase Date </span></td>
</tr>

</tbody>
</table>
</div>

</section>

</body>
</html>
```

5.9.8. AddProduct

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
th:replace="base3::layout(~{::section})">
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<section>


<div th:if="${session.message}" class="container">
```

```
<div th:classappend="${'alert-'+session.message.type}" class ="alert text-
center" role="alert">
<span th:text="${session.message.content}"></span>
</div>
<th:block th:text="${#session.removeAttribute('message')}"></th:block>
</div>

<div class="container">
<div class="row">
<div class="col-md-4 offset-md-4">
<div class="my-card">

<h4 class="text-center">Add Product!!</h4>



<!-- Form field  -->
<form action="" th:action="@{/processAddproduct}" method="post">





<div class="form-group">
<label for="category">Product Category</label> <select class="form-control"

id="cat" name="product_category">

<option value="Soccer Shoes">Soccer shoes</option>
<option value="Golf Shoes">Golf shoes</option>
<option value="Running Shoes">Running shoes</option>
<option value="High-Top Shoes">High-Top Shoes</option>
<option value="Hiking Boots">Hiking Boots Shoes</option>

</select>
</div>
<div class="form-group">
<label for="brand">Product Brand</label> <select
class="form-control" id="brand" name="product_brand">
```

```
<option value="puma">Puma</option>
<option value="Nike">Nike</option>
<option value="Addidas">Adidas</option>
<option value="RedTap">Red Tap</option>
<option value="Reebok">Reebok</option>


</select>
</div>
<div>
<label for="size">Product Size</label>
<input type="number"  required="required" class="form-control" id="size"
name="product_size">

</div>
<div>
<label for="gender">Select Gender</label> <select
class="form-control" id="gender" name="gender" required="required" >
<option value="Male">Male</option>
<option value="Female">Female</option>


</select>
</div>
<div class="form-group">
<label for="exampleFormControlInput1">Product Price</label> <input
type="number" name="product_price" required="required" class="form-
control" id="exampleFormControlInput1">
</div>


<div><button type="submit" class="btn btn-primary btn-
block">Add</button></div>


<!--form field  -->
```

```
</form>

</div>

</div>

</div>

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

5.9.9 .Update Product

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
th:replace="base3::layout(~{::section})">
```

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

<div th:if="${session.message}" class="container">
<div th:classappend="${'alert-'+session.message.type}" class ="alert text-center" role="alert">
<span th:text="${session.message.content}"></span>
</div>
<th:block th:text="${#session.removeAttribute('message')}"></th:block>

</div>
<div class="container">
<div class="row">
<div class="col-md-4 offset-md-4">
<div class="my-card">

<h3 class="text-center">Update Product!!</h3>


<!-- Form field  -->
<form action="" th:action="@{/process}" method="post">

<input type="hidden" th:value="${product.product_id}" name="product_id">


<div class="form-group">
<label for="category">Product Category</label>


<input  type="text"
th:value="${product.product_category}"required="required" class="form-control" id="cat" name="product_category">
</div>
```

```
<div class="form-group">
<label for="brand">Product Brand</label>
<input type="text"  th:value="${product.product_brand}" required="required"
class="form-control" id="brand" name="product_brand">

</div>
<div>
<label for="size">Product Size</label>
<input type="number"
th:value="${product.product_size}"required="required" class="form-control"
id="size" name="product_size">

</div>
<div>
<label for="gender">Select Gender</label>


<input type="text"  th:value="${product.gender}"required="required"
class="form-control" id="gender" name="gender">

</div>
<div class="form-group">
<label for="exampleFormControlInput1">Product Price</label> <input
type="number" th:value="${product.product_price}" name="product_price"
required="required" class="form-control" id="exampleFormControlInput1">
</div>


<div><button type="submit" class="btn btn-primary btn-
block">Update</button></div>


<!--form field  -->




</form>
```

```
</div>

</div>

</div>

</div>

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

5.9.10 .Change Password

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
th:replace="base3::layout(~{::section})">
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
```

```html
<body>
<section>
<div th:if="${session.message}" class="container">
<div th:classappend="${'alert-'+session.message.type}" class ="alert text-center" role="alert">
<span th:text="${session.message.content}"></span>
<th:block th:text="${#session.removeAttribute('message')}"></th:block>
</div>
</div>
<div class="container">
<div class="row">
<div class="col-md-6 offset-md-3">
<div class="my-card">

<h1 class="text-center">

Change Password !!

</h1>



<!-- Form field  -->



<form th:action="@{/change}" method="post">

<div class="form-group">
<label for="newpass">Enter New Password</label>
<input name="npass" type="text" class="form-control" id="newpass" aria-describedby="passHelp" placeholder="Enter New Password">

</div>
<div class="container text-center"><button type="submit" class="btn btn-success">Change</button></div>



</form>
```

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

5.9.11 .Base

```
<!doctype html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
th:fragment="layout(content)">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport"
content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-
beta2/css/all.min.css"
integrity="sha512-
YWzhKL2whUzgiheMoBFwW8CKV4qpHQAEuvilg9FAn5VJUDwKZZxkJNuGM4XkWuk94WCrrwslk8yWNGmY1EduTA=="
crossorigin="anonymous" referrerpolicy="no-referrer" />
<link rel="stylesheet" th:href="@{/css/style.css}" />

<title th:text="${title}"></title>
```

```
</head>
<body>


<!-- this is navbar  -->


<nav
class="navbar navbar-expand-lg navbar-dark bg-custom text-uppercase"
id="navbar">





<h2>SPortyShoes Portal</h2>


<div class="collapse navbar-collapse" id="navbarSupportedContent">
<ul class="navbar-nav ml-auto">

<li class="nav-item"><a th:href="@{/}" href="#"
class="nav-link"><i class="fa fa-home"></i>Home</a></li>


<li class="nav-item"><a th:href="@{/admin-login/}" href="#"
class="nav-link"><i class="fas fa-sign-in-alt"></i>
Administration Login</a></li>




</ul>
```

```
</div>

</nav>

<div th:replace="${content}"></div>

<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>

</body>
</html>
```

5.9.12 .Base1

```
<!doctype html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
th:fragment="layout(content)">
<head>
<!-- Required meta tags -->
```

```
<meta charset="utf-8">
<meta name="viewport"
content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">

<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta2/css/all.min.css"
integrity="sha512-YWzhKL2whUzgiheMoBFwW8CKV4qpHQAEuvilg9FAn5VJUDwKZZxkJNuGM4XkWuk94WCrrwslk8yWNGmY1EduTA=="
crossorigin="anonymous" referrerpolicy="no-referrer" />
<link rel="stylesheet" th:href="@{/css/style.css}" />

<title th:text="${title}"></title>
</head>
<body>


<!-- this is navbar  -->


<nav class="navbar navbar-expand-lg navbar-dark bg-custom" id="navbar">






<h2>SPortyShoes Portal</h2>
```

```
<div class="collapse navbar-collapse" id="navbarSupportedContent">
<ul class="navbar-nav ml-auto">


<li class="nav-item"><a th:href="@{/}" href="#"
class="nav-link"><i class="fa fa-home"></i>Home</a></li>
<li class="nav-item"><a th:href="@{/userlist/}" href="#"
class="nav-link"><i class="fa fa-user"></i>User List</a></li>
<li class="nav-item"><a th:href="@{/purchase}" href="#"
class="nav-link"><i class="fas fa-shopping-cart"></i>Purchase Report</a></li>
<li class="nav-item"><a th:href="@{/manageProduct}" href="#"
class="nav-link"><i class="fa fa-gear"></i>Manage Product</a></li>
<li class="nav-item"><a th:href="@{/changePassword}" href="#"
class="nav-link"><i class="fa fa-key"></i>Change Password</a></li>
<li class="nav-item"><a  th:href="@{/}"
class="nav-link"><i class="fas fa-sign-out-alt"></i>Logout</a></li>



</ul>




</div>

</nav>


<!-- End bar  -->

<!-- sidebar -->

<div class="sidebar">
```

```
<a href="#" th:href="@{/home}"class="item"><i class="fa fa-
home"></i>Home</a>
<a href="#"  th:href="@{/userlist}"class="item"><i class="fa fa-
user"></i>UserList</a> <a href="#" th:href="@{/purchase}" class="item"><i
class="fas fa-shopping-cart"></i>Purchase Report</a>
<a href="#" th:href="@{/manageProduct}" class="item"><i class="fa fa-
gear"></i>Manage Product</a> <a href="#"
th:href="@{/changePassword}"class="item"><i class="fa fa-key"></i>Change
Password</a> <a href="#" th:href="@{/}"class="item"><i class="fas fa-sign-
out-alt"></i>Logout</a>
<div class="divider"></div>

</div>
<!-- content area -->
<div class="content">

<i onclick="toggleSidebar()" class="fas fa-bars m-3"></i>
<div th:replace="${content}"></div>



</div>

<!--  End of content -->

<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5
KkN"
crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min
.js"
```

```
integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4
Q"
crossorigin="anonymous"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl
"
crossorigin="anonymous"></script>

<script src="https://code.jquery.com/jquery-3.6.0.js"
integrity="sha256-H+K7U5CnXl1h5ywQfKtSj8PCmoN9aaq30gDh27Xc0jk="
crossorigin="anonymous"></script>
<script src='http://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.2.1.js'></script>

<script th:src="@{/js/script.js}"></script>

</body>
</html>
```

5.9.13 .Base3

```
<!doctype html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
th:fragment="layout(content)">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport"
content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.c
ss"
```

```
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6J
Xm"
crossorigin="anonymous">

<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-
beta2/css/all.min.css"
integrity="sha512-
YWzhKL2whUzgiheMoBFwW8CKV4qpHQAEuvilg9FAn5VJUDwKZZxkJNuGM4X
kWuk94WCrrwslk8yWNGmY1EduTA=="
crossorigin="anonymous" referrerpolicy="no-referrer" />
<link rel="stylesheet" th:href="@{/css/style1.css}" />

<title th:text="${title}"></title>
</head>
<body>


<!-- this is navbar  -->


<nav class="navbar navbar-expand-lg navbar-dark bg-custom" id="navbar">





<h2>SPortyShoes Portal</h2>


<div class="collapse navbar-collapse" id="navbarSupportedContent">
<ul class="navbar-nav ml-auto">
```

```
<li class="nav-item"><a th:href="@{/adminDashboard}" href="#"
class="nav-link"><i class="fas fa-users-cog"></i>AdminDashboard</a></li>



<li class="nav-item"><a th:href="@{/}" href="#"
class="nav-link"><i class="fas fa-sign-out-alt"></i>Logout</a></li>



</ul>




</div>

</nav>


<!-- End bar  -->

<div th:replace="${content}"></div>

<!--  End of content -->

<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5
KkN"
crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min
.js"
```

```
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>

<script src="https://code.jquery.com/jquery-3.6.0.js"
integrity="sha256-H+K7U5CnXl1h5ywQfKtSj8PCmoN9aaq30gDh27Xc0jk="
crossorigin="anonymous"></script>
<script src='http://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.2.1.js'></script>
<script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
<script>

function deleteProduct(product_id){



swal({
title: "Are you sure?",
text: "You want to delete this product..!",
icon: "warning",
buttons: true,
dangerMode: true,
})
.then((willDelete) => {
if (willDelete) {
window.location="/manageProduct/"+product_id;


} else {
swal("Your product is safe!!");
}
});
}

</script>

</body>
</html>
```

5.10 .Application.properties

```
# h2 Database config
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.h2.console.enabled=true
```

5.11. Data.sql

```
create table userlist
(
user_id int not null,
user_name varchar(200) not null,
user_email varchar(200) not null,
user_location varchar(200) not null,
primary key(user_id)
);
```

```
INSERT INTO USERLIST (USER_ID,USER_NAME,USER_EMAIL,USER_LOCATION)
VALUES(1,'John','jhon@gmail.com','Delhi');
INSERT INTO USERLIST (USER_ID,USER_NAME,USER_EMAIL,USER_LOCATION)
VALUES(2,'Sayra','sayra@gmail.com','Mumbai');
INSERT INTO USERLIST (USER_ID,USER_NAME,USER_EMAIL,USER_LOCATION)
VALUES(3,'Rahul','rahul@12gmail.com','Delhi');
INSERT INTO USERLIST (USER_ID,USER_NAME,USER_EMAIL,USER_LOCATION)
VALUES(4,'Teena','teena@gmail.com','Jaipur');
INSERT INTO USERLIST (USER_ID,USER_NAME,USER_EMAIL,USER_LOCATION)
VALUES(5,'Sanjay','sanjay@gmail.com','Chennai');
INSERT INTO USERLIST (USER_ID,USER_NAME,USER_EMAIL,USER_LOCATION)
VALUES(6,'Pooja','pooja@gmail.com','Agra');
INSERT INTO USERLIST (USER_ID,USER_NAME,USER_EMAIL,USER_LOCATION)
VALUES(7,'Arhaan','arhaan@gmail.com','Banglore');




create table purchase(
product_id integer not null,
product_category varchar(255) not null,
product_brand varchar(255) not null,

product_size int(30) not null,
product_price int(30) not null,

purchase_date DATE,
primary key(product_id)
);
INSERT INTO PURCHASE(PRODUCT_ID,
PRODUCT_CATEGORY,PRODUCT_BRAND,PRODUCT_PRICE,PRODUCT_SIZE,PURCHASE_DATE)
VALUES (1, 'Soccer Shoes', 'Nike',3000, 6,TO_DATE('2021-06-06', 'yyyy-MM-dd'));
INSERT INTO PURCHASE(PRODUCT_ID, PRODUCT_CATEGORY,PRODUCT_BRAND,PRODUCT_PRICE,
PRODUCT_SIZE,PURCHASE_DATE)
VALUES (2, 'Golf Shoes', 'Puma', 2000, 5,TO_DATE('2021-01-01', 'yyyy-MM-dd'));
INSERT INTO PURCHASE(PRODUCT_ID, PRODUCT_CATEGORY,PRODUCT_BRAND,PRODUCT_PRICE,
PRODUCT_SIZE,PURCHASE_DATE)
VALUES (3, 'Golf Shoes', 'Reebok',2000, 7,SYSDATE());
INSERT INTO PURCHASE(PRODUCT_ID, PRODUCT_CATEGORY,PRODUCT_BRAND,PRODUCT_PRICE,
PRODUCT_SIZE,PURCHASE_DATE)
VALUES (4, 'Soccer Shoes', 'Puma', 1000, 5,TO_DATE('2021-12-30', 'yyyy-MM-dd'));
INSERT INTO PURCHASE(PRODUCT_ID, PRODUCT_CATEGORY,PRODUCT_BRAND,PRODUCT_PRICE,
PRODUCT_SIZE,PURCHASE_DATE)
VALUES (5, 'Running Shoes', 'Adidas',4000, 8,SYSDATE());
INSERT INTO PURCHASE(PRODUCT_ID, PRODUCT_CATEGORY,PRODUCT_BRAND,PRODUCT_PRICE,
PRODUCT_SIZE,PURCHASE_DATE)
VALUES (6, 'High Top Shoes', 'Reebok', 2200, 6, TO_DATE('2021-11-10', 'yyyy-MM-dd'));
```

```
INSERT INTO PURCHASE(PRODUCT_ID, PRODUCT_CATEGORY,PRODUCT_BRAND,PRODUCT_PRICE,
PRODUCT_SIZE,PURCHASE_DATE)
VALUES (7, 'Hiking Boots Shoes', 'Red Tap',3000, 4 ,TO_DATE('2021-05-02', 'yyyy-MM-dd'));
INSERT INTO PURCHASE(PRODUCT_ID, PRODUCT_CATEGORY,PRODUCT_BRAND,PRODUCT_PRICE,
PRODUCT_SIZE,PURCHASE_DATE)
VALUES (8, 'Running Shoes', 'Nike', 2000, 5, TO_DATE('2021-05-10', 'yyyy-MM-dd'));
INSERT INTO PURCHASE(PRODUCT_ID, PRODUCT_CATEGORY,PRODUCT_BRAND,PRODUCT_PRICE,
PRODUCT_SIZE,PURCHASE_DATE)
VALUES (9, 'Hiking Boots Shoes', 'Adidas', 1000, 4,TO_DATE('2021-10-11', 'yyyy-MM-dd'));
```

2. Pushing the Code to GitHub Repository

- Open your command prompt and navigate to the folder where you have created your files. cd

- Initialize your repository using the following command: **git init**

- Add all the files to your git repository using the following command: **git add .**

- Commit the changes using the following command: **git commit . -m "Changes have been committed."**

- Push the files to the folder you initially created using the following command: **git push -u origin main**