**Project3-E-commerce-website-for-Sporty-Shoes**

This document contains sections for:

* [Sprint planning and Task completion](file:///C:\Users\mirshad2\Downloads\JAVA%20Project\LockedMe-main\Specification%20Docs\LockedMe%20-%20Virtual%20Key%20for%20Repositories.docx#Sprint_plan)
* [Core concepts used in project](file:///C:\Users\mirshad2\Downloads\JAVA%20Project\LockedMe-main\Specification%20Docs\LockedMe%20-%20Virtual%20Key%20for%20Repositories.docx#Core_concepts)
* [Flow of the Application](file:///C:\Users\mirshad2\Downloads\JAVA%20Project\LockedMe-main\Specification%20Docs\LockedMe%20-%20Virtual%20Key%20for%20Repositories.docx#Flow).
* [Demonstrating the product capabilities, appearance, and user interactions.](file:///C:\Users\mirshad2\Downloads\JAVA%20Project\LockedMe-main\Specification%20Docs\LockedMe%20-%20Virtual%20Key%20for%20Repositories.docx#Product_capability)
* [Conclusions](file:///C:\Users\mirshad2\Downloads\JAVA%20Project\LockedMe-main\Specification%20Docs\LockedMe%20-%20Virtual%20Key%20for%20Repositories.docx#Conclusions)

The project is developed by Narendra Mandwe.

The code for this project is hosted at: https://github.com/NarendraMandwe/Sportyshoes.git

## Sprints planning and Task completion

The project is planned to be completed in 1 sprint. Tasks assumed to be completed in the sprint are:

* Creating the flow of the application
* Initializing git repository to track changes as development progresses.
* Writing the Java program to fulfill the requirements of the project.
* Testing the Java program with different kinds of User input
* Pushing code to GitHub.
* Creating this specification document highlighting application capabilities, appearance, and user interactions.

## Technologies and tools Used

* • Spring: to do the business logic and works a controller for the project.
* • JSP to handle the presentation view.
* • SQL: to create and manage the database.

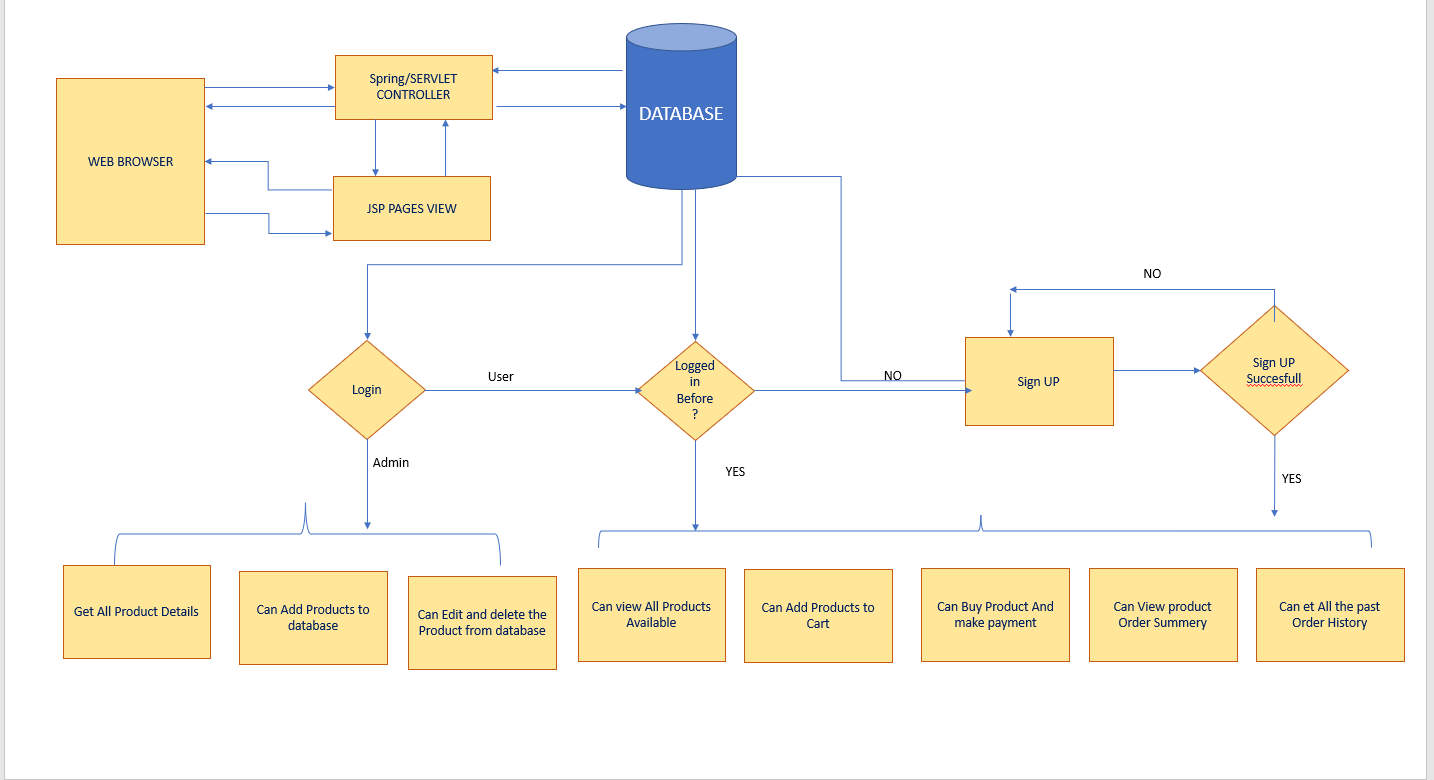
• • Mysql workbench: to administrate and manage the database manually.

* • Spring tool: to write and run the code.
* Application Properties : to run and deploy Spring application.

Product Backlog:

* Create database and tables.
* Connect the database to the project.
* Create models classes.
* Create a database utility class to retrieve data.
* Create login page.
* Create JSP files for pages of the project.
* Create Spring Model view
* Debug and Test the project.

## Flow of the Application:



## Demonstrating the product capabilities, appearance, and user interactions

To demonstrate the product capabilities, below are the sub-sections configured to highlight appearance and user interactions for the project:

1. [Creating the](file:///C:\Users\mirshad2\Downloads\JAVA%20Project\LockedMe-main\Specification%20Docs\LockedMe%20-%20Virtual%20Key%20for%20Repositories.docx#Step_1) Spring Starter Project in Spring tool
2. Creating JSP
3. [Creating](file:///C:\Users\mirshad2\Downloads\JAVA%20Project\LockedMe-main\Specification%20Docs\LockedMe%20-%20Virtual%20Key%20for%20Repositories.docx#Step_3) Spring Controller and repos
4. [Creating](file:///C:\Users\mirshad2\Downloads\JAVA%20Project\LockedMe-main\Specification%20Docs\LockedMe%20-%20Virtual%20Key%20for%20Repositories.docx#Step_4) Java class files
5. Screenshots

## **Step 1:** Creating a new project in Eclipse

* Open Eclipse
* Go to File -> New -> Project -> Spring Starter Project -> Next Enter project name and then click next add Dependencies according to your Project.
* Select your project and go to Webapp -> New -> JSP file.

## **Step 2:** Writing a JSP Code for login front end

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

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

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

<title>Insert title here</title>

<style>

**body,** **html** {

height: *100%*;

font-family: *Arial, Helvetica, sans-serif*;

}

\* {

box-sizing: *border-box*;

}

*.bg-img* {

/\* The image used \*/

background-image: *url("images/flight.jpg")*;

min-height: *1000px*;

/\* Center and scale the image nicely \*/

background-position: *center*;

background-repeat: *no-repeat*;

background-size: *cover*;

position: *relative*;

}

/\* Add styles to the form container \*/

*.container* {

position: *absolute*;

margin: *20px*;

max-width: *300px*;

padding: *16px*;

background-color: *white*;

}

/\* Full-width input fields \*/

**input**[type=text]**,** **input**[type=password] {

width: *100%*;

padding: *15px*;

margin: *5px 0 22px 0*;

border: *none*;

background: *#f1f1f1*;

}

**input**[type=text]*:focus***,** **input**[type=password]*:focus* {

background-color: *#ddd*;

outline: *none*;

}

/\* Set a style for the submit button \*/

*.btn* {

background-color: *#04AA6D*;

color: *white*;

padding: *16px 20px*;

border: *none*;

cursor: *pointer*;

width: *100%*;

opacity: *0.9*;

}

*.btn:hover* {

opacity: *1*;

}

</style>

</head>

<body>

<%String msg = (String)request.getAttribute("msg"); %>

<h1>User Login Page</h1>

<%**if**(msg!=**null**){ %>

<h2><%= msg %></h2>

<%} %>

<form action=*"loginUser"* class=*"container"* style="border:*1px solid #ccc*">

Username<input type=*"text"* name=*"uname"* placeholder=*"Enter Username"* required><br>

Password<input type=*"password"* name=*"upassword"* placeholder=*"Enter Password"* required><br>

<br></br>

<input type=*"submit"* value=*"login"* class=*"btn"*>

<br></br>

<a href=*"register.jsp"*>Register New User</a>

</form>

</body>

</html>

## **Step 3:** Writing a JSP Code for Registration:

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

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

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

<title>Insert title here</title>

<style>

**body,** **html** {

height: *100%*;

font-family: *Arial, Helvetica, sans-serif*;

}

\* {

box-sizing: *border-box*;

}

*.bg-img* {

/\* The image used \*/

background-image: *url("images/flight.jpg")*;

min-height: *1000px*;

/\* Center and scale the image nicely \*/

background-position: *center*;

background-repeat: *no-repeat*;

background-size: *cover*;

position: *relative*;

}

/\* Add styles to the form container \*/

*.container* {

position: *absolute*;

margin: *20px*;

max-width: *300px*;

padding: *16px*;

background-color: *white*;

}

/\* Full-width input fields \*/

**input**[type=text]**,** **input**[type=password] **,input**[type=email]{

width: *100%*;

padding: *15px*;

margin: *5px 0 22px 0*;

border: *none*;

background: *#f1f1f1*;

}

**input**[type=text]*:focus***,** **input**[type=password]*:focus***,input**[type=email]*:focus*{

background-color: *#ddd*;

outline: *none*;

}

/\* Set a style for the submit button \*/

*.btn* {

background-color: *#04AA6D*;

color: *white*;

padding: *16px 20px*;

border: *none*;

cursor: *pointer*;

width: *100%*;

opacity: *0.9*;

}

*.btn:hover* {

opacity: *1*;

}

</style>

</head>

<body>

<h1>Register User</h1>

<form action=*"registerUser"* class=*"container"* style="border:*1px solid #ccc*">

UserName<input type =*"text"* name=*"uname"* placeholder=*"Enter Username"* required><br>

Password<input type =*"password"* name=*"upassword"* placeholder=*"Enter Password"* required><br>

Email<input type =*"email"* name=*"uemail"* placeholder=*"Enter Email"* required><br>

<button type=*"submit"* value=*"Register"* class=*"btn"*>Register</button>

</form>

</body>

</html>

//Registration Successfull

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

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

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

<title>Insert title here</title>

</head>

<body>

<h1>Registration done Successfully!! Please Login</h1>

<%@include file=*"login.jsp"* %>

</body>

</html>

## **Step 4:** Writing a JSP code for User Homepage:

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

pageEncoding=*"ISO-8859-1"*%>

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

<%@ page import=*"java.util.\*"* %>

<!DOCTYPE html>

<html>

<head>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>

<style>

*.bg-img* {

/\* The image used \*/

background-image: *url("images/flight.jpg")*;

min-height: *1000px*;

/\* Center and scale the image nicely \*/

background-position: *center*;

background-repeat: *no-repeat*;

background-size: *cover*;

position: *relative*;

}

**table** {

border-collapse: *collapse*;

border-spacing: *0*;

width: *100%*;

border: *1px solid #ddd*;

}

**th,** **td** {

text-align: *left*;

padding: *8px*;

}

**tr***:nth-child*(*even*){background-color: *#f2f2f2*}

</style>

<%

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

**int** userid = (Integer)request.getAttribute("userid");

String username = (String)request.getAttribute("username");

String password = (String)request.getAttribute("password");%>

<h1>Welcome <%=request.getAttribute("username") %>!!</h1>

<table border=*"1"*>

<tr>

<th>Shoe Name</th><th>Shoe Cost</th><th>Buy</th>

</tr>

<%

**if**(list!=**null**){

**for**(Product p:list){ %>

<tr>

<td><%=p.getSname()%></td>

<td><%=p.getScost()%></td>

<td><a href=*"addToCart?sid=*<%= p.getSid()%>*&sname=*<%=p.getSname()%>*&scost=*<%=p.getScost() %>*&uid=*<%=userid%>*&username=*<%=username%>*&password=*<%=password%>*"*>Add to Cart</a></td>

</tr>

<%}} %>

</table>

<br></br>

<a href=*"/logout"*>Click here to logout</a>

</body>

</html>

## **Step 5:** Writing a JSP code to Display Products:

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

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

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

<title>Insert title here</title>

</head>

<body>

<h1><%= request.getAttribute("msg") %></h1>

<form action=*"getall"*>

<input type=*"submit"* value=*"Get Product details"*>

</form>

</body>

</html>

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

pageEncoding=*"ISO-8859-1"*%>

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

<%@ page import=*"java.util.\*"* %>

<!DOCTYPE html>

<html>

<head>

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

<title>Product Details</title>

</head>

<body>

<%

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

<table border=*"1"*>

<tr><th>Shoe Id</th><th>Shoe Name</th><th>Shoe Cost</th><th>Edit Operation</th><th>Delete Operation</th></tr>

<%

**if**(list!=**null**){

**for**(Product p:list){ %>

<tr>

<td><%=p.getSid()%></td>

<td><%=p.getSname()%></td>

<td><%=p.getScost()%></td>

<td><a href=*"edit?id=*<%= p.getSid()%>*&sname=*<%=p.getSname()%>*&semail=*<%=p.getScost() %>*"*>Edit</a></td>

<td><a href=*"delete?id=*<%= p.getSid()%>*&sname=*<%=p.getSname()%>*&semail=*<%=p.getSname() %>*"*>Delete</a></td>

</tr>

<%}} %>

</table>

</body>

</html>

## **Step 6:** Writing a JSP code for Buy Product :

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

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

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

<title>Shopping Cart</title>

<style>

**body,** **html** {

height: *100%*;

font-family: *Arial, Helvetica, sans-serif*;

}

\* {

box-sizing: *border-box*;

}

*.bg-img* {

/\* The image used \*/

background-image: *url("images/flight.jpg")*;

min-height: *1000px*;

/\* Center and scale the image nicely \*/

background-position: *center*;

background-repeat: *no-repeat*;

background-size: *cover*;

position: *relative*;

}

/\* Add styles to the form container \*/

*.container* {

position: *absolute*;

margin: *20px*;

max-width: *500px*;

padding: *16px*;

background-color: *white*;

}

/\* Full-width input fields \*/

**input**[type=text]**,** **input**[type=password] {

width: *100%*;

padding: *15px*;

margin: *5px 0 22px 0*;

border: *none*;

background: *#f1f1f1*;

}

**input**[type=text]*:focus***,** **input**[type=password]*:focus* {

background-color: *#ddd*;

outline: *none*;

}

/\* Set a style for the submit button \*/

*.btn* {

background-color: *#04AA6D*;

color: *white*;

padding: *16px 20px*;

border: *none*;

cursor: *pointer*;

width: *100%*;

opacity: *0.9*;

}

*.btn:hover* {

opacity: *1*;

}

</style>

</head>

<body>

<% **int** sid = (Integer)request.getAttribute("sid");

String sname = (String)request.getAttribute("sname");

String scost = (String)request.getAttribute("scost");

**int** uid = (Integer)request.getAttribute("userid");

%>

<form action=*"orderProduct"* class=*"container"* style="border:*1px solid #ccc*">

<h1>Order Details</h1>

Shoe Name<input type=*"text"* name=*"sname"* readonly value=*"*<%=sname %>*"* ><br/>

Shoe Cost<input type=*"text"* name=*"scost"* readonly value=*"*<%=scost %>*"* ><br/>

<br/>

<div class=*"bg-img"*>

<h1>Payment Details</h1>

<h2><p align=*"left"* >ENTER YOUR CARD DETAILS</p></h2>

<label for=*"cardName"*><b>CREDIT/DEBIT CARDS</b></label>

<select name=*"type"* name=*"userType"* required>

<option value =*"hdfc"*>HDFC Credit Card</option>

<option value=*"admin"*>HDFC Debit Card</option>

<option value=*"admin"*>ICICI Debit Card</option>

<option value=*"admin"*>ICICI Debit Card</option>

</select>

<br></br>

<label for=*"cardNo"*><b>Card Number</b></label>

<input type=*"text"* placeholder=*"Card number"* name=*"cardNo"* required>

<label for=*"nameOnCard"*><b>Name on Card</b></label>

<input type=*"text"* placeholder=*"Name on card"* name=*"nameOnCard"* required>

<input type=*"submit"* name=*"orderProduct"* value =*"Place Order"* class=*"btn"*>

<input type=*"hidden"* name=*"uid"* value=*"*<%= uid%>*"*>

<input type=*"hidden"* name=*"sid"* value=*"*<%= sid%>*"*>

</form>

</div>

</body>

## </html>

## **Step 7:** Writing a JSP code to Order Success Page:

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

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

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

<title>Insert title here</title>

</head>

<body>

<h1><%=request.getAttribute("msg")%></h1>

<%**int** userid = (Integer)request.getAttribute("userid");

%>

<form action=*"/orderReport"*>

<input type=*"submit"* value=*"View Order Report"*>

<input type=*"hidden"* name=*"userid"* value=*"*<%= userid%>*"*>

</form>

</body>

</html>

## **Step 8:** Writing a program to display order report details from database:

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

pageEncoding=*"ISO-8859-1"*%>

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

<%@ page import=*"java.util.\*"* %>

<!DOCTYPE html>

<html>

<head>

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

<style>

*.bg-img* {

/\* The image used \*/

background-image: *url("images/flight.jpg")*;

min-height: *1000px*;

/\* Center and scale the image nicely \*/

background-position: *center*;

background-repeat: *no-repeat*;

background-size: *cover*;

position: *relative*;

}

**table** {

border-collapse: *collapse*;

border-spacing: *0*;

width: *100%*;

border: *1px solid #ddd*;

}

**th,** **td** {

text-align: *left*;

padding: *8px*;

}

**tr***:nth-child*(*even*){background-color: *#f2f2f2*}

/\* Set a style for the submit button \*/

*.btn* {

background-color: *#04AA6D*;

color: *white*;

padding: *16px 20px*;

border: *none*;

cursor: *pointer*;

width: *10%*;

opacity: *0.9*;

}

*.btn:hover* {

opacity: *1*;

}

</style>

</head>

<body>

<h1>Order Report</h1>

<%

List<UserOrderSummary> orderList=(List<UserOrderSummary>)request.getAttribute("orderList"); %>

<table border=*"1"*>

<tr><th>Order Id</th><th>User Name</th><th>User Email</th><th>Shoe Name</th><th>Shoe Cost</th></tr>

<%

**if**(orderList!=**null**){

**for**(UserOrderSummary o:orderList){ %>

<tr>

<td><%=o.getOrderId() %></td>

<td><%=o.getUserName() %></td>

<td><%=o.getUserEmail() %></td>

<td><%=o.getShoeName() %></td>

<td><%=o.getShoeCost() %></td>

</tr>

<%}} %>

</table>

<br></br>

<a href=*"/logout"*>Click here to logout</a>

</body>

</html>

## **Step 9:** Writing a JSP code for Admin Homepage:

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

pageEncoding=*"ISO-8859-1"*%>

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

<%@ page import=*"java.util.\*"* %>

<!DOCTYPE html>

<html>

<head>

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

<style>

*.bg-img* {

/\* The image used \*/

background-image: *url("images/flight.jpg")*;

min-height: *1000px*;

/\* Center and scale the image nicely \*/

background-position: *center*;

background-repeat: *no-repeat*;

background-size: *cover*;

position: *relative*;

}

**table** {

border-collapse: *collapse*;

border-spacing: *0*;

width: *100%*;

border: *1px solid #ddd*;

}

**th,** **td** {

text-align: *left*;

padding: *8px*;

}

**tr***:nth-child*(*even*){background-color: *#f2f2f2*}

/\* Set a style for the submit button \*/

*.btn* {

background-color: *#04AA6D*;

color: *white*;

padding: *16px 20px*;

border: *none*;

cursor: *pointer*;

width: *10%*;

opacity: *0.9*;

}

*.btn:hover* {

opacity: *1*;

}

</style>

</head>

<body>

<%String message = (String)request.getAttribute("msg"); %>

<h1>Welcome Admin!</h1>

<%**if**(message !=**null**) {%>

<h2><%= message %></h2>

<%} %>

<%

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

<table border=*"1"*>

<tr><th>Shoe Id</th><th>Shoe Name</th><th>Shoe Cost</th><th>Edit Operation</th><th>Delete Operation</th></tr>

<%

**if**(list!=**null**){

**for**(Product p:list){ %>

<tr>

<td><%=p.getSid()%></td>

<td><%=p.getSname()%></td>

<td><%=p.getScost()%></td>

<td><a href=*"edit?sid=*<%= p.getSid()%>*&sname=*<%=p.getSname()%>*&scost=*<%=p.getScost() %>*"*>Edit</a></td>

<td><a href=*"delete?sid=*<%= p.getSid()%>*&sname=*<%=p.getSname()%>*&scost=*<%=p.getSname() %>*"*>Delete</a></td>

</tr>

<%}} %>

</table>

<br></br>

<form action=*"addProduct"*>

<input type=*"submit"* value=*"Add a product"* class=*"btn"*><br>

</form>

<br/>

<a href=*"/logout"*>Click here to logout</a>

</body>

</html>

## **Step 10:** Writing a JSP code for Adding the Product:

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

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

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

<title>Insert title here</title>

<style>

**body,** **html** {

height: *100%*;

font-family: *Arial, Helvetica, sans-serif*;

}

\* {

box-sizing: *border-box*;

}

*.bg-img* {

/\* The image used \*/

background-image: *url("images/flight.jpg")*;

min-height: *1000px*;

/\* Center and scale the image nicely \*/

background-position: *center*;

background-repeat: *no-repeat*;

background-size: *cover*;

position: *relative*;

}

/\* Add styles to the form container \*/

*.container* {

position: *absolute*;

margin: *20px*;

max-width: *300px*;

padding: *16px*;

background-color: *white*;

}

/\* Full-width input fields \*/

**input**[type=text]**,** **input**[type=password] **,input**[type=email]{

width: *100%*;

padding: *15px*;

margin: *5px 0 22px 0*;

border: *none*;

background: *#f1f1f1*;

}

**input**[type=text]*:focus***,** **input**[type=password]*:focus***,input**[type=email]*:focus*{

background-color: *#ddd*;

outline: *none*;

}

/\* Set a style for the submit button \*/

*.btn* {

background-color: *#04AA6D*;

color: *white*;

padding: *16px 20px*;

border: *none*;

cursor: *pointer*;

width: *100%*;

opacity: *0.9*;

}

*.btn:hover* {

opacity: *1*;

}

</style>

</head>

<body>

<h1><i>Add New Shoe Details</i></h1>

<form action=*"insert"* class=*"container"* style="border:*1px solid #ccc*">

Product name<input type=*"text"* name=*"sname"* placeholder=*"Enter Shoe Name"* required><br>

Product cost<input type=*"text"* name=*"scost"* placeholder=*"Enter Shoe Price"* required><br>

<input type=*"submit"* value=*"Add Product"* class=*"btn"*>

</form>

</body>

## </html>

## **Step 11:** Writing a JSP code for EDIT Product:

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

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

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

<title>Edit Shoe Details</title>

<style>

**body,** **html** {

height: *100%*;

font-family: *Arial, Helvetica, sans-serif*;

}

\* {

box-sizing: *border-box*;

}

*.bg-img* {

/\* The image used \*/

background-image: *url("images/flight.jpg")*;

min-height: *1000px*;

/\* Center and scale the image nicely \*/

background-position: *center*;

background-repeat: *no-repeat*;

background-size: *cover*;

position: *relative*;

}

/\* Add styles to the form container \*/

*.container* {

position: *absolute*;

margin: *20px*;

max-width: *300px*;

padding: *16px*;

background-color: *white*;

}

/\* Full-width input fields \*/

**input**[type=text]**,** **input**[type=password] {

width: *100%*;

padding: *15px*;

margin: *5px 0 22px 0*;

border: *none*;

background: *#f1f1f1*;

}

**input**[type=text]*:focus***,** **input**[type=password]*:focus* {

background-color: *#ddd*;

outline: *none*;

}

/\* Set a style for the submit button \*/

*.btn* {

background-color: *#04AA6D*;

color: *white*;

padding: *16px 20px*;

border: *none*;

cursor: *pointer*;

width: *100%*;

opacity: *0.9*;

}

*.btn:hover* {

opacity: *1*;

}

</style>

</head>

<body>

<% **int** sid = Integer.parseInt(request.getParameter("sid"));

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

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

%>

<h1><i>Edit Shoe Details</i></h1>

<form action=*"update"* class=*"container"* style="border:*1px solid #ccc*">

Shoe name<input type=*"text"* name=*"sname"* value=*"*<%=sname%>*"* required><br>

Shoe cost<input type=*"text"* name=*"scost"* value=*"*<%=scost%>*"* required><br>

<input type=*"submit"* value=*"Save Product"* class=*"btn"*>

<input type=*"hidden"* name=*"sid"* value=*"*<%=sid%>*"*>

</form>

</body>

</html>

## **Step 12:** Below are the 3 interface class

1.

**package** com.example.demo;

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

**public** **interface** ProductRepo **extends** JpaRepository<Product,Integer> {

}

2.

package com.example.demo;

import java.util.List;

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

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

public interface OrderDetailsRepo extends JpaRepository<OrderDetails, Integer> {

//@Query("select orders.userId from OrderDetails orders where orders.orderId=?1")

//List<OrderDetails> findOrdersByUser(int userid);

@Query(nativeQuery = true)

List<UserOrderSummary> getUserOrderSummary(Integer userId) ;

}

3. package com.example.demo;

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

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

public interface LoginRepo extends JpaRepository<User, Integer> {

@Query("select user from User user where user.uname=?1 and user.upassword=?2")

public User validateUser(String user,String Password);

}

## **Step 13:** Writing Java code for for DAO and pojo:

1.

/\*\*

\*

\*/

package com.example.demo;

/\*\*

\* @author Bhakti

\*

\*/

public class UserOrderSummary {

public UserOrderSummary() {}

public UserOrderSummary(Integer orderId, String userName, String shoeName,

String shoeCost , String userEmail) {

this.orderId = orderId;

this.userName = userName;

this.shoeName = shoeName;

this.shoeCost = shoeCost;

this.userEmail = userEmail ;

}

private Integer orderId;

private String userName;

private String shoeName;

private String shoeCost;

private String userEmail ;

public Integer getOrderId() {

return orderId;

}

public void setOrderId(Integer orderId) {

this.orderId = orderId;

}

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public String getShoeName() {

return shoeName;

}

public void setShoeName(String shoeName) {

this.shoeName = shoeName;

}

public String getShoeCost() {

return shoeCost;

}

public void setShoeCost(String shoeCost) {

this.shoeCost = shoeCost;

}

public String getUserEmail() {

return userEmail;

}

public void setUserEmail(String userEmail) {

this.userEmail = userEmail;

}

@Override

public String toString() {

return "UserOrderSummary [orderId=" + orderId + ", userName=" + userName + ", shoeName=" + shoeName

+ ", shoeCost=" + shoeCost + ", userEmail=" + userEmail + "]";

}

}

2.

package com.example.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringSecurityDemoTaskApplication {

public static void main(String[] args) {

SpringApplication.run(SpringSecurityDemoTaskApplication.class, args);

}

}

3.

package com.example.demo;

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

import org.springframework.context.annotation.Bean;

import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;

import org.springframework.security.crypto.password.NoOpPasswordEncoder;

import org.springframework.security.crypto.password.PasswordEncoder;

import org.springframework.security.web.authentication.AuthenticationSuccessHandler;

@EnableWebSecurity

public class SecurityConfig extends WebSecurityConfigurerAdapter {

private AuthenticationSuccessHandler authenticationSuccessHandler;

@Autowired

public SecurityConfig(AuthenticationSuccessHandler authenticationSuccessHandler) {

this.authenticationSuccessHandler = authenticationSuccessHandler;

}

/\*

\* @Override protected void configure(AuthenticationManagerBuilder auth) throws

\* Exception {

\*

\* System.out.println("Inside configure Authentication");

\* auth.inMemoryAuthentication() .withUser("admin") .password("admin")

\* .roles("ADMIN") .and() .withUser("user") .password("user") .roles("USER");

\*

\*

\*

\* }

\*/

@Override

public void configure(AuthenticationManagerBuilder auth) throws Exception {

auth.inMemoryAuthentication()

.withUser("user")

.password("user")

.roles("USER")

.and()

.withUser("admin")

.password("admin")

.roles("ADMIN");

}

@Bean

public PasswordEncoder getpassword() {

return NoOpPasswordEncoder.getInstance();

}

@Override

protected void configure(HttpSecurity http) throws Exception {

System.out.println("Inside configure Authorization!!!!!!!");

/\*

\* http.authorizeRequests() .antMatchers("/admin").hasRole("ADMIN")

\* .antMatchers("/user").hasAnyRole("ADMIN","USER").and().formLogin();

\* //.antMatchers("/").permitAll().and().formLogin();

\*/

http.authorizeRequests()

.antMatchers("/admin").hasRole("ADMIN")

.antMatchers("/user").hasAnyRole("USER")

.anyRequest()

.authenticated()

.and()

.formLogin()

.successHandler(authenticationSuccessHandler)

.permitAll()

.and()

.logout()

.invalidateHttpSession(true)

.deleteCookies("JSESSIONID")

.permitAll();

}

}

4.

package com.example.demo;

import java.util.ArrayList;

import java.util.List;

import java.util.Optional;

import java.util.logging.Logger;

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

import org.springframework.stereotype.Service;

@Service

public class ProductDAO {

Logger log = Logger.getAnonymousLogger();

@Autowired

ProductRepo repo;

@Autowired

OrderDetailsRepo orderRepo;

public Product insert(Product p) {

return repo.save(p);

}

public Product update(Product p) {

Product pp = repo.findById(p.getSid()).orElse(null);

pp.setSname(p.getSname());

pp.setScost(p.getScost());

return repo.save(pp);

}

public List<Product> getAll() {

return repo.findAll();

}

public void delete(int id) {

repo.deleteById(id);

}

public OrderDetails placeOrder(OrderDetails o) {

return orderRepo.save(o);

}

public List<UserOrderSummary> viewOrderReport(int userId) {

List<UserOrderSummary> orderSummary = orderRepo.getUserOrderSummary(userId);

Optional.ofNullable(orderSummary).orElse(new ArrayList<>())

.stream()

.forEach(odr -> {

log.info("\n Order Details "+ odr.toString());

});

//List<OrderDetails> userList = orderRepo.findOrdersByUser(userid);

//Product productList = repo.findAllById(ordersList);

return orderSummary ;

}

}

5.

package com.example.demo;

import java.util.ArrayList;

import java.util.List;

import java.util.logging.Logger;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

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

import org.springframework.stereotype.Controller;

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

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

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

import org.springframework.web.servlet.ModelAndView;

@RestController

public class ProductController {

@Autowired

ProductDAO dao;

Logger log = Logger.getAnonymousLogger();

@RequestMapping("/product")

public ModelAndView reDirect(HttpServletRequest request, HttpServletResponse response) {

ModelAndView mv = new ModelAndView();

log.info("in request mapping /");

mv.setViewName("product.jsp");

log.info("loaded admin page");

return mv;

}

@RequestMapping("/addProduct")

public ModelAndView redirect(HttpServletRequest request, HttpServletResponse response) {

ModelAndView mv = new ModelAndView();

mv.setViewName("addProduct.jsp");

return mv;

}

@RequestMapping("/insert")

public ModelAndView insert(HttpServletRequest request, HttpServletResponse response) {

ModelAndView mv = new ModelAndView();

Product p = new Product();

String insertmsg = "Record Inserted Successfully!!";

p.setSname(request.getParameter("sname"));

p.setScost(request.getParameter("scost"));

Product pp = dao.insert(p);

if (pp != null) {

mv.addObject("msg", insertmsg);

List<Product> list = dao.getAll();

mv.addObject("list", list);

mv.setViewName("admin.jsp");

} else {

mv.setViewName("fail.jsp");

}

return mv;

}

@RequestMapping("/update")

public ModelAndView update(HttpServletRequest request, HttpServletResponse response) {

ModelAndView mv = new ModelAndView();

Product p = new Product();

String updatemsg = "Record Updated Successfully!!";

p.setSid(Integer.parseInt(request.getParameter("sid")));

p.setSname(request.getParameter("sname"));

p.setScost(request.getParameter("scost"));

Product pp = dao.update(p);

if (pp != null) {

mv.addObject("msg", updatemsg);

List<Product> list = dao.getAll();

mv.addObject("list", list);

mv.setViewName("admin.jsp");

} else {

mv.setViewName("fail.jsp");

}

return mv;

}

@RequestMapping("/edit")

public ModelAndView editRedirect(HttpServletRequest request, HttpServletResponse response) {

System.out.println("Inside /edit mapping");

ModelAndView mv = new ModelAndView();

Product p = new Product();

request.setAttribute("sid",Integer.parseInt(request.getParameter("sid")));

request.setAttribute("sname",request.getParameter("sname"));

request.setAttribute("scost",request.getParameter("scost"));

mv.setViewName("editProduct.jsp");

return mv;

}

@RequestMapping("/delete")

public ModelAndView delete(HttpServletRequest request, HttpServletResponse response)

{

ModelAndView mv = new ModelAndView();

Product p = new Product();

String deletemsg = "Record Deleted Successfully!!";

int id= Integer.parseInt(request.getParameter("sid"));

request.setAttribute("sname",request.getParameter("sname"));

request.setAttribute("scost",request.getParameter("scost"));

dao.delete(id);

mv.addObject("msg", deletemsg);

List<Product> list = dao.getAll();

mv.addObject("list", list);

mv.setViewName("admin.jsp");

return mv;

}

@RequestMapping("/getall")

public ModelAndView getall(HttpServletRequest request, HttpServletResponse response) {

ModelAndView mv = new ModelAndView();

List<Product> list = dao.getAll();

mv.setViewName("displayproduct");

mv.addObject("list", list);

return mv;

}

@RequestMapping("/addToCart")

public ModelAndView addToCart(HttpServletRequest request, HttpServletResponse response) {

System.out.println("Inside /addToCart mapping");

ModelAndView mv = new ModelAndView();

Product p = new Product();

//int count = 0;

request.setAttribute("sid",Integer.parseInt(request.getParameter("sid")));

request.setAttribute("sname",request.getParameter("sname"));

request.setAttribute("scost",request.getParameter("scost"));

request.setAttribute("userid",Integer.parseInt(request.getParameter("uid")));

request.setAttribute("username",request.getParameter("username"));

request.setAttribute("password",request.getParameter("password"));

//request.setAttribute("count",++count);

List<Product> list = dao.getAll();

mv.addObject("list", list);

mv.setViewName("buyProduct.jsp");

return mv;

}

@ResponseBody

@RequestMapping("/orderProduct")

public ModelAndView placeOrder(HttpServletRequest request, HttpServletResponse response) {

log.info("in request mapping /orderProduct");

ModelAndView mv = new ModelAndView();

String msg ="Order Placed Successfully!!";

OrderDetails order = new OrderDetails();

order.setUserId(Integer.parseInt(request.getParameter("uid")));

order.setShoeId(Integer.parseInt(request.getParameter("sid")));

OrderDetails oo = dao.placeOrder(order);

if(oo != null) {

mv.addObject("msg", msg);

mv.addObject("userid", order.getUserId());

mv.setViewName("orderSuccessPage.jsp");

}

return mv;

}

@ResponseBody

@RequestMapping("/orderReport")

public ModelAndView viewOrderReport(HttpServletRequest request, HttpServletResponse response) {

log.info("in request mapping /orderProduct");

ModelAndView mv = new ModelAndView();

int userid = Integer.parseInt(request.getParameter("userid"));

String msg ="No Orders Placed yet for User : "+userid;

List<UserOrderSummary> orderList = dao.viewOrderReport(userid);

mv.addObject("orderList", orderList);

if(orderList != null) {

mv.setViewName("OrderReport.jsp");

}

return mv;

}

}

6.

package com.example.demo;

import javax.persistence.ConstructorResult;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.NamedNativeQuery;

import javax.persistence.SqlResultSetMapping;

import javax.persistence.Column;

import javax.persistence.ColumnResult;

import lombok.Data;

@Entity

@Data

@NamedNativeQuery(name = "OrderDetails.getUserOrderSummary",

query=

" select "

+ " p.sname as shoeName , p.scost as shoeCost ,"

+ " u.uname as userName , u.uemail as userEmail , "

+ " o.order\_id as orderId "

+ " from "

+ " product p ,user u , order\_details o "

+ " where "

+ " o.user\_id= :userId and o.user\_id = u.uid and "

+ " p.sid = o.shoe\_id",

resultSetMapping = "Mapping.UserOrderSummary"

)

@SqlResultSetMapping(name = "Mapping.UserOrderSummary",

classes = @ConstructorResult (

targetClass = UserOrderSummary.class,

columns = {

@ColumnResult(name= "orderId"),

@ColumnResult(name= "userName"),

@ColumnResult(name= "shoeName"),

@ColumnResult(name= "shoeCost"),

@ColumnResult(name= "userEmail"),

})

)

public class OrderDetails {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private int orderId;

private int userId;

private int shoeId;

}

7.

package com.example.demo;

import javax.annotation.Generated;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import lombok.Data;

import net.bytebuddy.dynamic.loading.ClassReloadingStrategy.Strategy;

@Entity

@Data

public class Product {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private int sid;

private String sname;

private String scost;

}

8.

package com.example.demo;

import java.util.logging.Logger;

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

import org.springframework.stereotype.Service;

import org.springframework.web.servlet.ModelAndView;

@Service

public class LoginDAO {

@Autowired

LoginRepo repo;

public User validateUser(String uname,String upassword) {

User u = repo.validateUser(uname, upassword);

return u;

}

public User registerUser(User u) {

User uu = repo.save(u);

return uu;

}

}

9.

package com.example.demo;

import java.util.List;

import java.util.logging.Logger;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

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

import org.springframework.stereotype.Controller;

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

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

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

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

import org.springframework.web.servlet.ModelAndView;

@Controller

public class LoginController {

@Autowired

ProductDAO dao;

@Autowired

LoginDAO logindao;

Logger log = Logger.getAnonymousLogger();

@ResponseBody

@RequestMapping("/admin")

public ModelAndView loadfrontpageadmin(HttpServletRequest request, HttpServletResponse response) {

ModelAndView mv = new ModelAndView();

List<Product> list = dao.getAll();

mv.addObject("list", list);

log.info("in request mapping /");

mv.setViewName("admin.jsp");

log.info("loaded admin page");

return mv;

}

@ResponseBody

@RequestMapping("/user")

public ModelAndView loadfrontpageuser(HttpServletRequest request, HttpServletResponse response) {

ModelAndView mv = new ModelAndView();

List<Product> list = dao.getAll();

mv.addObject("list", list);

log.info("in request mapping /");

mv.setViewName("user.jsp");

log.info("loaded user page");

return mv;

}

@ResponseBody

@RequestMapping("/userLogin")

public ModelAndView loadUserLoginPage(HttpServletRequest request, HttpServletResponse response) {

ModelAndView mv = new ModelAndView();

List<Product> list = dao.getAll();

mv.addObject("list", list);

log.info("in request mapping /");

mv.setViewName("userLogin.jsp");

log.info("loaded user page");

return mv;

}

@ResponseBody

@RequestMapping("loginUser")

public ModelAndView loginAction(HttpServletRequest request, HttpServletResponse response)

{

ModelAndView mv = new ModelAndView();

log.info("Inside the login action");

User s = new User();

String username = request.getParameter("uname");

String password = request.getParameter("upassword");

log.info("Received input parameters username"+username+" and password"+password);

User u =logindao.validateUser(username, password);

if (u !=null) {

log.info("Login is successfull");

List<Product> list = dao.getAll();

mv.addObject("list", list);

mv.setViewName("user.jsp");

mv.addObject("userid", u.getUid());

mv.addObject("username", username);

mv.addObject("password", password);

log.info("control passed to user.jsp");

}

else

{

String msg = "The Username and Password has some problem ,please check or else register with the below link";

log.info("Login is unsuccessfull");

mv.addObject("msg", msg);

mv.setViewName("userLogin.jsp");

log.info("control passed to userLogin.jsp");

}

return mv;

}

@ResponseBody

@RequestMapping("/registerUser")

public ModelAndView registerUser(HttpServletRequest request, HttpServletResponse response) {

ModelAndView mv = new ModelAndView();

String msg ="User Registered successfully.Login to proceed";

log.info("in request mapping /registerUser");

User user = new User();

user.setUname(request.getParameter("uname"));

user.setUpassword(request.getParameter("upassword"));

user.setUemail(request.getParameter("uemail"));

User u = logindao.registerUser(user);

if(u != null) {

mv.addObject("msg", msg);

mv.setViewName("userLogin.jsp");

}

return mv;

}

}

10.

package com.example.demo;

import java.io.IOException;

import java.util.Set;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.core.Authentication;

import org.springframework.security.core.authority.AuthorityUtils;

import org.springframework.security.web.authentication.AuthenticationSuccessHandler;

@Configuration

public class CustomAuthenticationSuccessHandler implements AuthenticationSuccessHandler {

@Override

public void onAuthenticationSuccess(HttpServletRequest request, HttpServletResponse response,

Authentication authentication) throws IOException, ServletException {

Set<String> roles = AuthorityUtils.authorityListToSet(authentication.getAuthorities());

System.out.println("Roles::::::"+roles);

if (roles.contains("ROLE\_ADMIN")) {

System.out.println("Inside CustomAuthenticationSuccessHandler if condition Role:Admin");

response.sendRedirect("/admin");

} else {

System.out.println("Inside CustomAuthenticationSuccessHandler if condition Role:User");

response.sendRedirect("/userLogin");

}

}

}

## **Step 14:** Enter Below code in you Application Properties file

server.port=8087

#datasource

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

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

spring.datasource.username=root

spring.datasource.password=Invisible16

#jpa-hibernate

spring.jpa.hibernate.ddl-auto=update

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

spring.jpa.show-sql=true

## **Step 15:** Pushing the code to GitHub repository

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

**cd <folder path>**

* Initialize 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 <commit message>**

* Push the files to the folder you initially created using the following command:

**git push -u origin master**

## Conclusions

* Admin can ADD Product, EDIT and Delete
* User can View All the Product
* User can Add the Product to Cart and buy them.
* User can view the order Summery