

Sporty Shoes

Project Source Code

Version History:

Author:	Mayur Bharatkumar Solanki
Purpose:	Project Source Code and GITHUB
Date:	9 th December 2021
Version:	1.0

Table of Content:

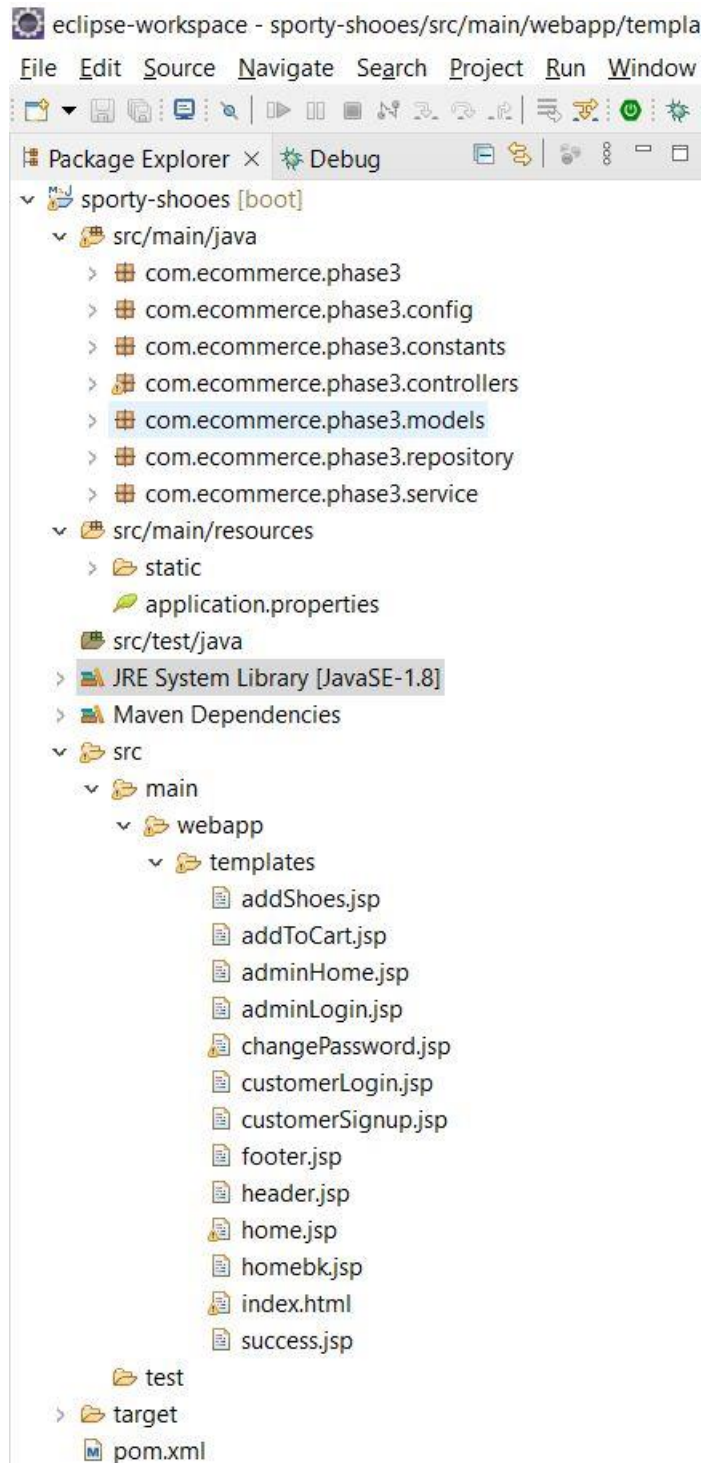
Project GITHUB Link: 3

Project Code: **Error! Bookmark not defined.**

Project GITHUB Link:

Repository Name:	SL_FSD_PHASE3_Project
Github Link:	https://github.com/mursky66/SL_FSD_PHASE3_Project.git

PROJECT CODE :



Index :

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="Mark Otto, Jacob Thornton, and Bootstrap contributors">
    <meta name="generator" content="Hugo 0.88.1">
    <title>Album example · Bootstrap v5.1</title>

    <link rel="canonical" href="https://getbootstrap.com/docs/5.1/examples/album/">

    <!-- Bootstrap core CSS -->
    <link href="../../assets/dist/css/bootstrap.min.css" rel="stylesheet">

    <style>
      .bd-placeholder-img {
        font-size: 1.125rem;
        text-align: middle;
        -webkit-user-select: none;
        -moz-user-select: none;
        user-select: none;
      }

      @media (min-width: 768px) {
        .bd-placeholder-img-lg {
          font-size: 3.5rem;
        }
      }
    </style>

  </head>
  <body>

    <header>
      <div class="collapse bg-dark" id="navbarHeader">
        <div class="container">
          <div class="row">
            <div class="col-sm-8 col-md-7 py-4">
              <h4 class="text-white">About</h4>
              <p class="text-muted">Add some information about the album below, the author, or any other background context. Make it a few sentences long so folks can pick up some informative tidbits. Then, link them off to some social networking sites or contact information.</p>
            </div>
            <div class="col-sm-4 offset-md-1 py-4">
              <h4 class="text-white">Contact</h4>
            </div>
          </div>
        </div>
      </div>
    </header>
  </body>
</html>
```

```

        <ul class="list-unstyled">
            <li><a href="#" class="text-white">Follow on Twitter</a></li>
            <li><a href="#" class="text-white">Like on Facebook</a></li>
            <li><a href="#" class="text-white">Email me</a></li>
        </ul>
    </div>
</div>
</div>
</div>
<div class="navbar navbar-dark bg-dark shadow-sm">
    <div class="container">
        <a href="#" class="navbar-brand d-flex align-items-center">
            <svg xmlns="http://www.w3.org/2000/svg" width="20" height="20" fill="none"
stroke="currentColor" stroke-linecap="round" stroke-linejoin="round" stroke-width="2"
aria-hidden="true" class="me-2" viewBox="0 0 24 24"><path d="M23 19a2 2 0 0 1-2 2H3a2
2 0 0 1-2-2V8a2 2 0 0 1 2-2h4L2-3h6L2 3h4a2 2 0 0 1 2 2z"/><circle cx="12" cy="13"
r="4"/></svg>
            <strong>Album</strong>
        </a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarHeader" aria-controls="navbarHeader" aria-expanded="false" aria-
label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
    </div>
</div>
</header>

<main>

    <section class="py-5 text-center container">
        <div class="row py-lg-5">
            <div class="col-lg-6 col-md-8 mx-auto">
                <h1 class="fw-light">Album example</h1>
                <p class="lead text-muted">Something short and leading about the collection
below—its contents, the creator, etc. Make it short and sweet, but not too short so
folks don't simply skip over it entirely.</p>
                <p>
                    <a href="#" class="btn btn-primary my-2">Main call to action</a>
                    <a href="#" class="btn btn-secondary my-2">Secondary action</a>
                </p>
            </div>
        </div>
    </section>

    <div class="album py-5 bg-light">
        <div class="container">

            <div class="row row-cols-1 row-cols-sm-2 row-cols-md-3 g-3">
                <div class="col">
                    <div class="card shadow-sm">
                        <svg class="bd-placeholder-img card-img-top" width="100%" height="225"
xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: Thumbnail"
preserveAspectRatio="xMidYMid slice">

```

```
focusable="false"><title>Placeholder</title><rect width="100%" height="100%"
fill="#55595c"/><text x="50%" y="50%" fill="#eceeef" dy=".3em">Thumbnail</text></svg>
```

```
<div class="card-body">
  <p class="card-text">This is a wider card with supporting text below as
a natural lead-in to additional content. This content is a little bit longer.</p>
  <div class="d-flex justify-content-between align-items-center">
    <div class="btn-group">
      <button type="button" class="btn btn-sm btn-outline-
secondary">View</button>
      <button type="button" class="btn btn-sm btn-outline-
secondary">Edit</button>
    </div>
    <small class="text-muted">9 mins</small>
  </div>
</div>
</div>
<div class="col">
```

```
  <div class="card shadow-sm">
    <svg class="bd-placeholder-img card-img-top" width="100%" height="225"
xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: Thumbnail"
preserveAspectRatio="xMidYMid slice"
focusable="false"><title>Placeholder</title><rect width="100%" height="100%"
fill="#55595c"/><text x="50%" y="50%" fill="#eceeef" dy=".3em">Thumbnail</text></svg>
```

```
  <div class="card-body">
    <p class="card-text">This is a wider card with supporting text below as
a natural lead-in to additional content. This content is a little bit longer.</p>
    <div class="d-flex justify-content-between align-items-center">
      <div class="btn-group">
        <button type="button" class="btn btn-sm btn-outline-
secondary">View</button>
        <button type="button" class="btn btn-sm btn-outline-
secondary">Edit</button>
      </div>
      <small class="text-muted">9 mins</small>
    </div>
  </div>
</div>
<div class="col">
```

```
  <div class="card shadow-sm">
    <svg class="bd-placeholder-img card-img-top" width="100%" height="225"
xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: Thumbnail"
preserveAspectRatio="xMidYMid slice"
focusable="false"><title>Placeholder</title><rect width="100%" height="100%"
fill="#55595c"/><text x="50%" y="50%" fill="#eceeef" dy=".3em">Thumbnail</text></svg>
```

```
  <div class="card-body">
    <p class="card-text">This is a wider card with supporting text below as
a natural lead-in to additional content. This content is a little bit longer.</p>
    <div class="d-flex justify-content-between align-items-center">
      <div class="btn-group">
```

```

        <button type="button" class="btn btn-sm btn-outline-
secondary">View</button>
        <button type="button" class="btn btn-sm btn-outline-
secondary">Edit</button>
    </div>
    <small class="text-muted">9 mins</small>
</div>
</div>
</div>
</div>

<div class="col">
    <div class="card shadow-sm">
        <svg class="bd-placeholder-img card-img-top" width="100%" height="225"
xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: Thumbnail"
preserveAspectRatio="xMidYMid slice"
focusable="false"><title>Placeholder</title><rect width="100%" height="100%"
fill="#55595c"/><text x="50%" y="50%" fill="#eceeef" dy=".3em">Thumbnail</text></svg>

        <div class="card-body">
            <p class="card-text">This is a wider card with supporting text below as
a natural lead-in to additional content. This content is a little bit longer.</p>
            <div class="d-flex justify-content-between align-items-center">
                <div class="btn-group">
                    <button type="button" class="btn btn-sm btn-outline-
secondary">View</button>
                    <button type="button" class="btn btn-sm btn-outline-
secondary">Edit</button>
                </div>
                <small class="text-muted">9 mins</small>
            </div>
        </div>
    </div>
</div>
<div class="col">
    <div class="card shadow-sm">
        <svg class="bd-placeholder-img card-img-top" width="100%" height="225"
xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: Thumbnail"
preserveAspectRatio="xMidYMid slice"
focusable="false"><title>Placeholder</title><rect width="100%" height="100%"
fill="#55595c"/><text x="50%" y="50%" fill="#eceeef" dy=".3em">Thumbnail</text></svg>

        <div class="card-body">
            <p class="card-text">This is a wider card with supporting text below as
a natural lead-in to additional content. This content is a little bit longer.</p>
            <div class="d-flex justify-content-between align-items-center">
                <div class="btn-group">
                    <button type="button" class="btn btn-sm btn-outline-
secondary">View</button>
                    <button type="button" class="btn btn-sm btn-outline-
secondary">Edit</button>
                </div>
                <small class="text-muted">9 mins</small>
            </div>
        </div>
    </div>
</div>

```



```

    </div>
  </div>
  <div class="col">
    <div class="card shadow-sm">
      <svg class="bd-placeholder-img card-img-top" width="100%" height="225"
xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: Thumbnail"
preserveAspectRatio="xMidYMid slice"
focusable="false"><title>Placeholder</title><rect width="100%" height="100%"
fill="#55595c"/><text x="50%" y="50%" fill="#eceeef" dy=".3em">Thumbnail</text></svg>

      <div class="card-body">
        <p class="card-text">This is a wider card with supporting text below as
a natural lead-in to additional content. This content is a little bit longer.</p>
        <div class="d-flex justify-content-between align-items-center">
          <div class="btn-group">
            <button type="button" class="btn btn-sm btn-outline-
secondary">View</button>
            <button type="button" class="btn btn-sm btn-outline-
secondary">Edit</button>
          </div>
          <small class="text-muted">9 mins</small>
        </div>
      </div>
    </div>
  </div>

  <div class="col">
    <div class="card shadow-sm">
      <svg class="bd-placeholder-img card-img-top" width="100%" height="225"
xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: Thumbnail"
preserveAspectRatio="xMidYMid slice"
focusable="false"><title>Placeholder</title><rect width="100%" height="100%"
fill="#55595c"/><text x="50%" y="50%" fill="#eceeef" dy=".3em">Thumbnail</text></svg>

      <div class="card-body">
        <p class="card-text">This is a wider card with supporting text below as
a natural lead-in to additional content. This content is a little bit longer.</p>
        <div class="d-flex justify-content-between align-items-center">
          <div class="btn-group">
            <button type="button" class="btn btn-sm btn-outline-
secondary">View</button>
            <button type="button" class="btn btn-sm btn-outline-
secondary">Edit</button>
          </div>
          <small class="text-muted">9 mins</small>
        </div>
      </div>
    </div>
  </div>
  <div class="col">
    <div class="card shadow-sm">
      <svg class="bd-placeholder-img card-img-top" width="100%" height="225"
xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: Thumbnail"
preserveAspectRatio="xMidYMid slice"

```

```
<focusable=false"><title>Placeholder</title><rect width="100%" height="100%"
fill="#55595c"/><text x="50%" y="50%" fill="#eceeef" dy=".3em">Thumbnail</text></svg>

    <div class="card-body">
      <p class="card-text">This is a wider card with supporting text below as
a natural lead-in to additional content. This content is a little bit longer.</p>
      <div class="d-flex justify-content-between align-items-center">
        <div class="btn-group">
          <button type="button" class="btn btn-sm btn-outline-
secondary">View</button>
          <button type="button" class="btn btn-sm btn-outline-
secondary">Edit</button>
        </div>
        <small class="text-muted">9 mins</small>
      </div>
    </div>
  </div>
</div>
<div class="col">
  <div class="card shadow-sm">
    <svg class="bd-placeholder-img card-img-top" width="100%" height="225"
xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: Thumbnail"
preserveAspectRatio="xMidYMid slice"
focusable=false"><title>Placeholder</title><rect width="100%" height="100%"
fill="#55595c"/><text x="50%" y="50%" fill="#eceeef" dy=".3em">Thumbnail</text></svg>

    <div class="card-body">
      <p class="card-text">This is a wider card with supporting text below as
a natural lead-in to additional content. This content is a little bit longer.</p>
      <div class="d-flex justify-content-between align-items-center">
        <div class="btn-group">
          <button type="button" class="btn btn-sm btn-outline-
secondary">View</button>
          <button type="button" class="btn btn-sm btn-outline-
secondary">Edit</button>
        </div>
        <small class="text-muted">9 mins</small>
      </div>
    </div>
  </div>
</div>
</div>
</div>
</div>
</main>

<footer class="text-muted py-5">
  <div class="container">
    <p class="float-end mb-1">
      <a href="#">Back to top</a>
    </p>
    <p class="mb-1">Album example is &copy; Bootstrap, but please download and
customize it for yourself!</p>
```

```
    <p class="mb-0">New to Bootstrap? <a href="/">Visit the homepage</a> or read our  
<a href="../getting-started/introduction/">getting started guide</a>.</p>  
  </div>  
</footer>
```

```
  <script src="../assets/dist/js/bootstrap.bundle.min.js"></script>
```

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

Home :

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="description" content="">
<meta name="author"
      content="Mark Otto, Jacob Thornton, and Bootstrap contributors">
<meta name="generator" content="Hugo 0.88.1">
<title>SportyShoes</title>

<link rel="canonical"
      href="https://getbootstrap.com/docs/5.1/examples/album/">
<!-- Bootstrap core CSS -->
<link href="./assets/dist/css/bootstrap.min.css" rel="stylesheet">

<style>
.bd-placeholder-img {
  font-size: 1.125rem;
  text-anchor: middle;
  -webkit-user-select: none;
  -moz-user-select: none;
  user-select: none;
}

@media ( min-width : 768px) {
  .bd-placeholder-img-lg {
    font-size: 3.5rem;
  }
}
</style>
</head>
<body onload="loadImages()">
  <header>
    <div class="collapse bg-dark" id="navbarHeader">
      <div class="container">
        <div class="row">
          <div class="col-sm-12 offset-md-1 py-4">
            <ul class="navbar-nav me-auto mb-2">
              <li class="nav-item"><a class="nav-link
active"
                                aria-current="page"
href="./home">Home</a></li>
              <li class="nav-item"><a class="nav-link
active" href="./addShoes">Add Shoes</a>
            </li>
            </ul>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

```

        <div class="navbar navbar-dark bg-dark shadow-sm">
            <div class="container">
                <a href="#" class="navbar-brand d-flex align-items-
center"> <svg
                                xmlns="http://www.w3.org/2000/svg" width="18"
height="18"
                                fill="currentColor" class="bi bi-handbag-
fill" viewBox="0 0 16 16">
                                    <path
                                                d="M8 1a2 2 0 0 0-2 2v2H5V3a3 3 0 1 1 6
0v2h-1V3a2 2 0 0 0-2 2z" M5 3.36a1.5 1.5 0 0 0-1.483 1.277L.85 13.13A2.5 2.5 0 0 0
3.322 16h9.355a2.5 2.5 0 0 0 2.473-2.87L1.028-6.853A1.5 1.5 0 0 0 12.64
5H11v1.5a.5.5 0 0 1-1 0V5H6v1.5a.5.5 0 0 1-1 0V5z" />
                                    </svg>
                                <strong>SportyShoes</strong>
                            </a>
                            <button class="navbar-toggler" type="button"
                                data-bs-toggle="collapse" data-bs-
target="#navbarHeader"
                                aria-controls="navbarHeader" aria-expanded="false"
                                aria-label="Toggle navigation">
                                <span class="navbar-toggler-icon"></span>
                            </button>
                        </div>
                    </div>
                </header>
                <main>
                    <section class="py-5 text-center container">
                        <div class="row py-lg-5">
                            <div class="col-lg-6 col-md-8 mx-auto">
                                <h1 class="fw-light">SportyShoes</h1>
                            </div>
                        </div>
                    </section>

                    <div class="album py-5 bg-light">
                        <div class="container">

                            <div class="row row-cols-1 row-cols-sm-2 row-cols-md-3 g-
3"
                                id="listShoes"></div>
                            </div>
                        </div>
                    </main>

                    <footer class="text-muted py-5">
                        <div class="container">
                            <p class="float-end mb-1">
                                <a href="#">Back to top</a>
                            </p>
                            <p class="mb-1">Copyrights &copy; 2021</p>
                            <p class="mb-0">SportyShoes</p>
                        </div>
                    </footer>
                <script src="./assets/dist/js/bootstrap.bundle.min.js"></script>

```

```
<script src="./script/jquery.min.js"></script>  
<script src="./script/home.js"></script>
```

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

ADD SHOES:

```
<%@page import="com.ecommerce.phase3.constants.SportyShoesConstant"%>
<%@page import="java.util.Objects"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>SportyShoes-Add Shoess</title>
</head>
<body>

    <jsp:include page="header.jsp" />
    <h1>Add Shoes</h1>

    <%
    if (Objects.nonNull(request.getAttribute(SportyShoesConstant.MESSAGE))) {
    %>
    <%=request.getAttribute(SportyShoesConstant.MESSAGE).toString()%>
    <%
    }
    %>

    <form action="./addShoes" method="POST" enctype="multipart/form-data">

        Shoes Category <select name="shoesCategory">
            <option value="sports">Sports</option>
            <option value="casual">Casual</option>
            <option
value="traditional">Traditional</option>
            </select> <br>
        Shoes Name <input type="text" name="shoesName"> <br>
        Shoes Price <input type="text" name="shoesPrice"> <br>
        Shoes Image <input type="file" name="shoesImge"> <br> <input
type="submit"
        name="btnAddProduct" value="Add Shoes">

    </form>
    <jsp:include page="footer.jsp" />

</body>
</html>
```

ADD CART :

```
<!DOCTYPE html>
<%@page import="com.ecommerce.phase3.models.Shoes"%>
<%@page import="java.util.List"%>
<%@page import="java.util.Objects"%>
<%@page import="com.ecommerce.phase3.constants.SportyShoesConstant"%>
<html>
<head>
<meta charset="ISO-8859-1">
<title>SportyShoes-AddToCart</title>
</head>
<body onload="loadImages()">

    <jsp:include page="header.jsp" />
    <h1>Add To Cart</h1>

    <%
    if (Objects.nonNull(request.getAttribute(SportyShoesConstant.DATA))) {
        Shoes shoes = (Shoes) request.getAttribute(SportyShoesConstant.DATA);
    %>
    <form action="./checkout" method="post">
    Shoes Name: <%=shoes.getShoesName() %> <br><br>

    Shoes Price: <%=shoes.getShoesPrice() %> <br><br>

    <input type="submit" value="checkout">
    </form>
    <%
    }
    %>

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

    <script src="./script/jquery.min.js"></script>
    <script src="./script/home.js"></script>

</body>
</html>
```


ADMIN HOME:

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

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

    <table>
        <tr>
            <td width="20%">
                <p>
                    <a href="./addShoes">Add
                        Shoes</a>
                    <a href="./viewShoes">View Shoes</a>
                </p>
            </td>
            <td><h1>Admin Dashboard</h1></td>
        </tr>
    </table>

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

</body>
</html>
```

ADMIN LOGIN:

```
<%@page import="com.ecommerce.phase3.constants.SportyShoesConstant"%>
<%@page import="java.util.Objects"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>SportyShoes-AdminLogin</title>
</head>
<body>

    <jsp:include page="header.jsp" />
    <h1>Admin Login</h1>

    <%

        if(Objects.nonNull(request.getAttribute(SportyShoesConstant.MESSAGE))){
            %>
            <%=request.getAttribute(SportyShoesConstant.MESSAGE).toString() %>
            <%
        }

    %>

    <form action="./adminLogin" method="POST">

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

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

    <input type="submit" name="btnLogin" value="Admin Login">

    </form>

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

</body>
</html>
```

CHANGE PASSWORD:

```
<%@page import="com.ecommerce.phase3.constants.SportyShoesConstant"%>
<%@page import="java.util.Objects"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>SportyShoes-Change Password</title>
</head>
<body>

    <jsp:include page="header.jsp" />
    <h1>Change Password</h1>

<%

    if(Objects.nonNull(request.getAttribute(SportyShoesConstant.MESSAGE))){
        %>
        <%=request.getAttribute(SportyShoesConstant.MESSAGE).toString() %>
        <%
    }

    %>

    <form action="./changePassword" method="POST">

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

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

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

        <input
            type="submit" name="btnChngPwd" value="Change Password">

    </form>

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

</body>
</html>
```

CUSTOMER LOGIN:

```
<%@page import="com.ecommerce.phase3.constants.SportyShoesConstant"%>
<%@page import="java.util.Objects"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>SportyShoes-CustomerLogin</title>
</head>
<body>

    <jsp:include page="header.jsp" />
    <h1>Customer Login</h1>

    <%
    if (Objects.nonNull(request.getAttribute(SportyShoesConstant.MESSAGE))) {
    %>
    <%=request.getAttribute(SportyShoesConstant.MESSAGE).toString()%>
    <%
    }
    %>

    <form action="./customer-login" method="POST">

        Email <input type="text" name="email"> <br> Password <input
            type="password" name="password"> <br> <input
            type="submit" name="btnLogin" value="Login">

    </form>

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

</body>
</html>
```

CUSTOMER_SIGNUP :

```
<%@page import="com.ecommerce.phase3.constants.SportyShoesConstant"%>
<%@page import="java.util.Objects"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>SportyShoes-Customer Signup</title>
</head>
<body>

    <jsp:include page="header.jsp" />
    <h1>Customer Signup</h1>

    <%
    if (Objects.nonNull(request.getAttribute(SportyShoesConstant.MESSAGE))) {
    %>
    <%=request.getAttribute(SportyShoesConstant.MESSAGE).toString()%>
    <%

    }
    %>

    <form action="./customer-signup" method="POST">

        Name <input type="text" name="name"> <br> Email <input
            type="text" name="email"> <br> Gender <select
                name="gender">
                <option value="male">Male</option>
                <option value="female">Female</option>
            </select> <br> Password <input type="password" name="password"> <br>
        Mobile <input type="text" name="mobile"> <br> <input
            type="submit" name="btnSignup" value="Signup">

    </form>

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

</body>
</html>
```

HOME_BK :

```
<!DOCTYPE html>
<%@page import="com.ecommerce.phase3.models.Shoes"%>
<%@page import="java.util.List"%>
<%@page import="java.util.Objects"%>
<%@page import="com.ecommerce.phase3.constants.SportyShoesConstant"%>
<html>
<head>
<meta charset="ISO-8859-1">
<title>SportyShoes-Home</title>
</head>
<body onload="loadImages()">

    <jsp:include page="header.jsp" />
    <h1>Home</h1>

    <div id="listShoes"></div>

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

    <script src="./script/jquery.min.js"></script>
    <script src="./script/home.js"></script>

</body>
</html>
```

SUCCESS :

```
<!DOCTYPE html>
<%@page import="com.ecommerce.phase3.constants.SportyShoesConstant"%>
<%@page import="java.util.Objects"%>
<html>
<head>
<meta charset="ISO-8859-1">
<title>SportyShoes-Home</title>
</head>
<body>

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

    <%

        if(Objects.nonNull(request.getAttribute(SportyShoesConstant.MESSAGE))){
            %>
            <h1> <%=request.getAttribute(SportyShoesConstant.MESSAGE).toString() %>
        </h1>
            <%
        }

    %>

    <jsp:include page="footer.jsp" />
</body>
</html>
```

HEADER :

```
<p>  
    <a href="./home">Home</a>  
  
    <a href="./adminLogin">Admin Login</a>  
  
    <a href="./customer-signup">Customer Signup</a>  
  
    <a href="./customer-login">Customer Login</a>  
  
    <a href="./addShoes">Add Shoes</a>  
  
    <a href="./changePassword">Password Change</a>  
</p>
```

FOOTER :

```
<p> SportyShoes &copy; 2021</p>
```


POM.XML :

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>
  <packaging>war</packaging>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.6.1</version>
    <relativePath />
  </parent>

  <groupId>com.ecommerce</groupId>
  <artifactId>sporty-shoes</artifactId>
  <version>0.0.1</version>
  <name>sporty-shoes</name>
  <description>sporty-shoes</description>

  <properties>
    <java.version>1.8</java.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-validation</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <scope>runtime</scope>
    </dependency>

    <!-- JSTL -->
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>jstl</artifactId>
    </dependency>
  </dependencies>
</project>
```

```
<!-- To compile JSP files -->
<dependency>
  <groupId>org.apache.tomcat.embed</groupId>
  <artifactId>tomcat-embed-jasper</artifactId>
  <scope>provided</scope>
</dependency>
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <optional>true</optional>
</dependency>
<dependency>
  <groupId>commons-io</groupId>
  <artifactId>commons-io</artifactId>
  <version>2.11.0</version>
</dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
      <configuration>
        <excludes>
          <exclude>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
          </exclude>
        </excludes>
      </configuration>
    </plugin>
  </plugins>
</build>
</project>
```

APPLICATION.PROPERTIES:

```
spring.datasource.url=jdbc:mysql://localhost:3306/mecommerce
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

```
# Keep the connection alive if idle for a long time (needed in production)
spring.datasource.testWhileIdle=true
spring.datasource.validationQuery=SELECT 1
```

```
spring.jpa.show-sql=true
# Hibernate ddl auto (create, create-drop, update): with "create-drop" the database
spring.jpa.hibernate.ddl-auto=update
```

```
# Naming strategy
spring.jpa.hibernate.naming.implicit-
strategy=org.hibernate.boot.model.naming.ImplicitNamingStrategyLegacyHbmImpl
spring.jpa.hibernate.naming.physical-
strategy=org.springframework.boot.orm.jpa.hibernate.SpringPhysicalNamingStrategy
```

```
# Allows Hibernate to generate SQL optimized for a particular DBMS
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
```

JAVA CLASS :

SportyShooesApplication .Java

```
package com.ecommerce.phase3;
```

```
import org.springframework.boot.SpringApplication;
```

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

```
import lombok.extern.log4j.Log4j2;
```

```
@Log4j2
```

```
@SpringBootApplication
```

```
public class SportyShooesApplication {
```

```
    public static void main(String[] args) {
```

```
        SpringApplication.run(SportyShooesApplication.class, args);
```

```
        log.info("SportyShooesApplication Started");
```

```
    }
```

```
}
```

SportyShoesConfig :

```
package com.ecommerce.phase3.config;
```

```
import org.springframework.context.annotation.Bean;
```

```
import org.springframework.context.annotation.ComponentScan;
```

```
import org.springframework.context.annotation.Configuration;
```

```
import org.springframework.web.servlet.ViewResolver;
```

```
import org.springframework.web.servlet.config.annotation.EnableWebMvc;
```

```
import org.springframework.web.servlet.config.annotation.ResourceHandlerRegistry;
```

```
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
```

```
import org.springframework.web.servlet.view.InternalResourceViewResolver;
```

```
@EnableWebMvc
```

```
@Configuration
```

```
@ComponentScan("com.ecommerce")
```

```
public class SportyShoesConfig implements WebMvcConfigurer {
```

```
    @Bean
```

```
    public ViewResolver jspViewResolver() {
```

```
        InternalResourceViewResolver bean = new InternalResourceViewResolver();
```

```
        bean.setPrefix("/templates/");
```

```
        bean.setSuffix(".jsp");
```

```
        return bean;
```

```
    }
```

```
    private static final String[] CLASSPATH_RESOURCE_LOCATIONS = { "classpath:/resources/",  
"classpath:/static/", "classpath:/public/" };
```

```
@Override  
public void addResourceHandlers(ResourceHandlerRegistry registry){  
  
    registry.addResourceHandler("/**").addResourceLocations(CLASSPATH_RESOURCE_LOCATIO  
NS);  
    }  
  
}
```

SportyShoesConstant :

```
package com.ecommerce.phase3.constants;

public class SportyShoesConstant {

    private SportyShoesConstant() {
    }

    public static final String MESSAGE = "MESSAGE";
    public static final String DATA = "DATA";
    public static final String UPLOAD_DIRECTORY = "prjImages/";
}
```

AddToCartController :

```
package com.ecommerce.phase3.controllers;

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.PostMapping;

import com.ecommerce.phase3.constants.SportyShoesConstant;
import com.ecommerce.phase3.models.Shoes;
import com.ecommerce.phase3.service.ShoesService;

import lombok.extern.log4j.Log4j2;

@Controller
public class AddToCartController {

    @Autowired
    private ShoesService shoesService;

    @PostMapping(path = "/addToCart")
    public String addToCartPage(HttpServletRequest request, HttpServletResponse response){
        log.info("addToCartPage called");
        try {
```



```

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

        Shoes shoes = shoesService.getShoesById(Long.valueOf(shoesId));

        request.setAttribute(SportyShoesConstant.DATA, shoes);

    } catch (Exception ex) {

        log.error("Error: {}", ex.getMessage());

    }

    return "addToCart";

}

@PostMapping(path = "/checkout")
public String checkout(HttpServletRequest request, HttpServletResponse response) {

    log.info("checkout called");

    try {

        request.setAttribute(SportyShoesConstant.MESSAGE, "Your order placed
successfully");

    } catch (Exception ex) {

        log.error("Error: {}", ex.getMessage());

    }

    return "success";

}

// @PostMapping(path = "/adminLogin")
// public String adminLogin(HttpServletRequest request, HttpServletResponse response) {
//     log.info("adminLogin called");
//     try {
//         String email = request.getParameter("email");
//         String password = request.getParameter("password");

```

```
//  
//          log.info(email);  
//          log.info(password);  
//  
//          boolean isLoginSuccess = adminService.performAdminLogin(email, password );  
//          if (isLoginSuccess) {  
//              return "adminHome";  
//          }  
//  
//          request.setAttribute(SportyShoesConstant.MESSAGE, "Invalid credential");  
//  
//      } catch (Exception ex) {  
//          log.error("Error: {}", ex.getMessage());  
//      }  
//      return "adminLogin";  
//  }  
}
```

AdminController :

```
package com.ecommerce.phase3.controllers;
```

```
import java.io.IOException;
```

```
import java.io.InputStream;
```

```
import java.io.OutputStream;
```

```
import java.util.Base64;
```

```
import java.util.Objects;
```

```
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.GetMapping;
```

```
import org.springframework.web.bind.annotation.PostMapping;
```

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

```
import org.springframework.web.multipart.MultipartFile;
```

```
import com.ecommerce.phase3.constants.SportyShoesConstant;
```

```
import com.ecommerce.phase3.models.Shoes;
```

```
import com.ecommerce.phase3.service.AdminService;
```

```
import lombok.extern.log4j.Log4j2;
```

```
@Log4j2
```

```
@Controller
```

```
public class AdminController {

    @Autowired
    private AdminService adminService;

    @GetMapping(path = "/adminLogin")
    public String adminLoginPage() {
        log.info("adminLoginPage called");
        return "adminLogin";
    }

    @PostMapping(path = "/adminLogin")
    public String adminLogin(HttpServletRequest request,
        HttpServletResponse response) {
        log.info("adminLogin called");
        try {
            String email = request.getParameter("email");
            String password = request.getParameter("password");

            log.info(email);
            log.info(password);

            boolean isLoginSuccess =
adminService.performAdminLogin(email, password);
            if (isLoginSuccess) {
                return "adminHome";
            }
        }
    }
}
```

```
        request.setAttribute(SportyShoesConstant.MESSAGE,
"Invalid credential");
```

```
    } catch (Exception ex) {
        log.error("Error: {}", ex.getMessage());
    }
    return "adminLogin";
}
```

```
@GetMapping(path = "/changePassword")
public String changePassword() {
    log.info("changePassword called");
    return "changePassword";
}
```

```
@PostMapping(path = "/changePassword")
public String changePassword(HttpServletRequest request,
HttpServletResponse response) {
    log.info("changePassword called");
    try {
        String email = request.getParameter("email");
        String oldpassword =
request.getParameter("oldpassword");
        String newpassword =
request.getParameter("newpassword");

        log.info(email);
        log.info(newpassword);
```

```
        boolean isPasswordChanged =  
adminService.changeAdminPassword(email, oldpassword, newpassword);  
  
        if (isPasswordChanged) {  
  
            request.setAttribute(SportyShoesConstant.MESSAGE,  
"Password has changed successfully");  
  
            return "success";  
  
        }
```

```
        request.setAttribute(SportyShoesConstant.MESSAGE,  
"Password does not changed");
```

```
    } catch (Exception ex) {  
  
        log.error("Error: {}", ex.getMessage());  
  
    }  
  
    return "changePassword";  
  
}
```

```
@GetMapping(path = "/addShoes")  
  
public String addShoes() {  
  
    log.info("addShoes called");  
  
    return "addShoes";  
  
}
```

```
@PostMapping(value = "/addShoes")  
  
public String addShoes(@RequestParam("shoesImge") MultipartFile  
multipartFile, HttpServletRequest request,  
  
    HttpServletResponse response) throws IOException {  
  
    log.info("addShoes called");
```

```
        OutputStream outputStream = null;
        InputStream inputStream = null;
        Shoes shoes = new Shoes();
        try {
            String fileName = multipartFile.getOriginalFilename();
            if (Objects.nonNull(fileName)) {
                String encodedString =
Base64.getEncoder().encodeToString(multipartFile.getBytes());

                shoes.setShoesImg(encodedString);
            }

            String shoesCategory =
request.getParameter("shoesCategory");
            String shoesName = request.getParameter("shoesName");
            String shoesPrice =
request.getParameter("shoesPrice");

            shoes.setShoesCategory(shoesCategory);
            shoes.setShoesName(shoesName);

            if (Objects.nonNull(shoesPrice) &&
!shoesPrice.isEmpty()) {
                shoes.setShoesPrice(Double.valueOf(shoesPrice));
            }

            adminService.saveShoes(shoes);
            request.setAttribute(SportyShoesConstant.MESSAGE,
"Shoes added successfully");
            return "success";
        }
```

```
        } catch (Exception ex) {  
            log.error("Error: {}", ex.getMessage());  
            request.setAttribute(SportyShoesConstant.MESSAGE,  
"Shoes not added");  
  
        } finally {  
            if (inputStream != null) {  
                inputStream.close();  
            }  
            if (outputStream != null) {  
                outputStream.close();  
            }  
        }  
  
        return "addShoes";  
  
    }  
  
}
```


CustomerController :

```
package com.ecommerce.phase3.controllers;
```

```
import java.io.IOException;
```

```
import java.util.Objects;
```

```
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.GetMapping;
```

```
import org.springframework.web.bind.annotation.PostMapping;
```

```
import com.ecommerce.phase3.constants.SportyShoesConstant;
```

```
import com.ecommerce.phase3.models.Customer;
```

```
import com.ecommerce.phase3.service.CustomerService;
```

```
import lombok.extern.log4j.Log4j2;
```

```
@Log4j2
```

```
@Controller
```

```
public class CustomerController {
```

```
    @Autowired
```

```
    private CustomerService customerService;
```

```
@GetMapping(path = { "/customer-signup" })
```

```
public String customerSignUpPage() {
```

```
    log.info("customerSignUpPage called");
```

```
    return "customerSignup";
```

```
}
```

```
@PostMapping(value = "/customer-signup")
```

```
public String addCustomer(HttpServletRequest request, HttpServletResponse response)  
throws IOException {
```

```
    log.info("addCustomer called");
```

```
    try {
```

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

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

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

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

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

```
        Customer cust = new Customer();
```

```
        cust.setName(name);
```

```
        cust.setEmail(email);
```

```
        cust.setGender(gender);
```

```
        cust.setPassword(password);
```

```
        cust.setMobile(mobile);
```

```
        Customer savedCust = customerService.saveCustomer(cust);
```

```
        if (Objects.nonNull(savedCust)) {
```

```
        request.setAttribute(SportyShoesConstant.MESSAGE, "Signup  
successfully");  
  
        return "success";  
    }  
  
    } catch (Exception ex){  
        log.error("Error: {}", ex.getMessage());  
  
    } finally {  
    }  
    request.setAttribute(SportyShoesConstant.MESSAGE, "Customer signup failed");  
  
    return "customerSignup";  
  
}
```

```
@GetMapping(path = "/customer-login")  
public String customerLoginPage() {  
    log.info("customerLoginPage called");  
    return "customerLogin";  
}
```

```
@PostMapping(path = "/customer-login")  
public String customerLogin(HttpServletRequest request, HttpServletResponse response) {  
    log.info("customerLogin called");  
    try {  
        String email = request.getParameter("email");  
        String password = request.getParameter("password");
```

```
        log.info(email);
        log.info(password);

        boolean isLoginSuccess = customerService.performAdminLogin(email,
password);

        if (isLoginSuccess) {
            return "addToCart";
        }

        request.setAttribute(SportyShoesConstant.MESSAGE, "Invalid credential");

    } catch (Exception ex) {
        log.error("Error: {}", ex.getMessage());
    }
    return "adminLogin";
}

}
```

HomeController :

package com.ecommerce.phase3.controllers;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import

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

import org.springframework.stereotype.Controller;

import

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

import

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

import

com.ecommerce.phase3.constants.SportyShoesConstant;

import com.ecommerce.phase3.models.Shoes;

import com.ecommerce.phase3.service.ShoesService;

import lombok.extern.log4j.Log4j2;

@Log4j2

@Controller

public class HomeController {

 @Autowired

 private ShoesService shoesService;

 @GetMapping(path = { "/", "/home" })

 public String homePage(HttpServletRequest request) {

 log.info("homePage called");

// List<Shoes> allShoes = shoesService.findAll();

// log.info("allShoes size: {}", allShoes.size());

// request.setAttribute(SportyShoesConstant.DATA,
allShoes);

 return "home";

 }

```
@GetMapping(path = { "/listShoes" })  
public @ResponseBody List<Shoes> listShoes() {  
    log.info("listShoes called");  
    return shoesService.findAll();  
}  
  
}
```

ADMIN :

```
package com.ecommerce.phase3.models;
```

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

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

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

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

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

```
import lombok.Data;
```

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

```
@Entity
```

```
@Data
```

```
public class Admin {
```

```
    @Id
```

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

```
    private Long id;
```

```
    private String email;
```

```
    private String password;
```

```
}
```


CUSTOMER :

```
package com.ecommerce.phase3.models;
```

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

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

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

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

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

```
import lombok.Data;
```

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

```
@Entity
```

```
@Data
```

```
public class Customer {
```

```
    @Id
```

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

```
    private Long id;
```

```
    private String name;
```

```
    private String email;
```

```
    private String password;
```

```
    private String mobile;
```

```
    private String gender;
```

```
}
```

SHOES:

```
package com.ecommerce.phase3.models;
```

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

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

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

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

```
import javax.persistence.Lob;
```

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

```
import lombok.Data;
```

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

```
@Entity
```

```
@Data
```

```
public class Shoes {
```

```
    @Id
```

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

```
    private Long id;
```

```
    private String shoesCategory;
```

```
    private String shoesName;
```

```
    private Double shoesPrice;
```

```
    @Lob
```

```
    private String shoesImg;
```

```
}
```

AdminRepo:

```
package com.ecommerce.phase3.repository;

import org.springframework.data.repository.CrudRepository;

import com.ecommerce.phase3.models.Admin;

public interface AdminRepo extends CrudRepository<Admin, Long>{

    Admin findByEmailAndPassword(String email, String password);

    Admin findByEmail(String email);

}
```

CustomerRepo:

```
package com.ecommerce.phase3.repository;

import org.springframework.data.repository.CrudRepository;

import com.ecommerce.phase3.models.Customer;

public interface CustomerRepo extends CrudRepository<Customer, Long> {

    Customer findByEmailAndPassword(String email, String password);

}
```

ShoesRepo:

```
package com.ecommerce.phase3.repository;

import org.springframework.data.repository.CrudRepository;

import com.ecommerce.phase3.models.Shoes;

public interface ShoesRepo extends CrudRepository<Shoes, Long> {

}
```

AdminService:

```
package com.ecommerce.phase3.service;
```

```
import java.util.Objects;
```

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

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

```
import com.ecommerce.phase3.models.Admin;
```

```
import com.ecommerce.phase3.models.Shoes;
```

```
import com.ecommerce.phase3.repository.AdminRepo;
```

```
import com.ecommerce.phase3.repository.ShoesRepo;
```

```
import lombok.extern.log4j.Log4j2;
```

```
@Log4j2
```

```
@Service
```

```
public class AdminService {
```

```
    @Autowired
```

```
    private AdminRepo adminRepo;
```

```
    @Autowired
```

```
    private ShoesRepo shoesRepo;
```

```
    public boolean performAdminLogin(String email, String password) {
```

```
        try {
```

```
            Admin adminObj = adminRepo.findByEmailAndPassword(email, password);
```

```

        if (Objects.nonNull(adminObj)){
            return true;
        }
    } catch (Exception ex) {
        log.error("Error: {}", ex.getMessage());
    }
    return false;
}

public boolean changeAdminPassword(String email, String oldPassword, String newPassword)
{
    try {
        Admin adminObj = adminRepo.findByEmail(email);

        if (Objects.nonNull(adminObj)) {

            if (!oldPassword.equals(adminObj.getPassword())) {
                return false;
            }

            adminObj.setPassword(newPassword);
            adminRepo.save(adminObj);
            return true;
        }

    } catch (Exception ex) {
        log.error("Error: {}", ex.getMessage());
    }
}

```

```
        return false;
    }

    public Shoes saveShoes(Shoes shoes){
        try{
            shoes = shoesRepo.save(shoes);
            return shoes;
        } catch (Exception ex){
            log.error("Error: {}", ex.getMessage());
            return null;
        }
    }
}
```

CustomerService :

```
package com.ecommerce.phase3.service;
```

```
import java.util.Objects;
```

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

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

```
import com.ecommerce.phase3.models.Customer;
```

```
import com.ecommerce.phase3.repository.CustomerRepo;
```

```
import lombok.extern.log4j.Log4j2;
```

```
@Log4j2
```

```
@Service
```

```
public class CustomerService {
```

```
    @Autowired
```

```
    private CustomerRepo custRepo;
```

```
    public Customer saveCustomer(Customer c) {
```

```
        try {
```

```
            return custRepo.save(c);
```

```
        } catch (Exception ex) {
```

```
            log.error("Error: {}", ex.getMessage());
```

```
        }
```

```
        return null;
```

```
    }

    public boolean performAdminLogin(String email, String password){
        try{
            Customer customerObj = custRepo.findByEmailAndPassword(email,
password);

            if (Objects.nonNull(customerObj)){
                return true;
            }
        } catch (Exception ex){
            log.error("Error: {}", ex.getMessage());
        }
        return false;
    }
}
```


ShoesService:

```
package com.ecommerce.phase3.service;
```

```
import java.util.ArrayList;
```

```
import java.util.Iterator;
```

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

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

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

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

```
import com.ecommerce.phase3.models.Shoes;
```

```
import com.ecommerce.phase3.repository.ShoesRepo;
```

```
import lombok.extern.log4j.Log4j2;
```

```
@Log4j2
```

```
@Service
```

```
public class ShoesService {
```

```
    @Autowired
```

```
    private ShoesRepo shoesRepo;
```

```
    public List<Shoes> findAll() {
```

```
        try {
```

```
            Iterable<Shoes> shoes = shoesRepo.findAll();
```

```
            Iterator<Shoes> shoesItr = shoes.iterator();
```

```
            List<Shoes> allShoes = new ArrayList<Shoes>();
```

```
        while (shoesItr.hasNext()) {  
            Shoes s = shoesItr.next();  
            allShoes.add(s);  
        }  
        return allShoes;  
    } catch (Exception ex) {  
        log.error("Error: {}", ex.getMessage());  
    }  
    return null;  
}  
  
public Shoes getShoesById(Long shoesId) {  
    Optional<Shoes> shoesOpt = shoesRepo.findById(shoesId);  
    return shoesOpt.orElse(null);  
}  
}
```

DATABASE:

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- flyway
- flyway2
- hibernatedemo
- mecommerce**
 - Tables
 - hibernate_sequence
 - tbl_admin
 - tbl_customer
 - tbl_shoes
 - Views
 - Stored Procedures
 - Functions
 - shoppingssystem
- sys

Administration Schemas

Information

No object selected

tbl_admin 1 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	17:02:01	SELECT * FROM mecommerce.tbl_admin LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Result Grid

id	email	password
1	mayunsolanki66@gmail.com	27528592
2	demo@gmail.com	123

HIBERNATE_SEQUENCE_TABLE

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- flyway
- flyway2
- hibernatedemo
- mecommerce**
 - Tables
 - hibernate_sequence
 - tbl_admin
 - tbl_customer
 - tbl_shoes
 - Views
 - Stored Procedures
 - Functions
- shoppingsystem
- sys

tbl_shoes hibernate_sequence tbl_admin tbl_customer tbl_shoes tbl_a

Limit to 1000 rows

```
1 • SELECT * FROM mecommerce.hibernate_sequence;
```

Result Grid

	next_val
▶	23

Administration Schemas

ADMIN TABLE :

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- flyway
- flyway2
- hibernatedemo
- mecommerce**
 - Tables
 - hibernate_sequence
 - tbl_admin
 - tbl_customer
 - tbl_shoes
 - Views
 - Stored Procedures
 - Functions
- shoppingsystem
- sys

Administration Schemas

Information

No object selected

tbl_admin 1 x

tbl_admin

tbl_customer

tbl_shoes

tbl_admin

1 • SELECT * FROM mecommerce.tbl_admin;

Result Grid

Filter Rows:

Edit:

	id	email	password
▶	1	mayursolanki66@gmail.com	27528592
	2	demo@gmail.com	123
★	NULL	NULL	NULL

CUSTOMER TABLE :

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- flyway
- flyway2
- hibernatedemo
- mecommerce**
 - Tables
 - hibernate_sequence
 - tbl_admin
 - tbl_customer
 - tbl_shoes
 - Views
 - Stored Procedures
 - Functions
- shoppingsystem
- sys

tbl_admin tbl_customer tbl_shoes tbl_admin hibernate_sequence tbl_customer tbl

1 • `SELECT * FROM mecommerce.tbl_customer;`

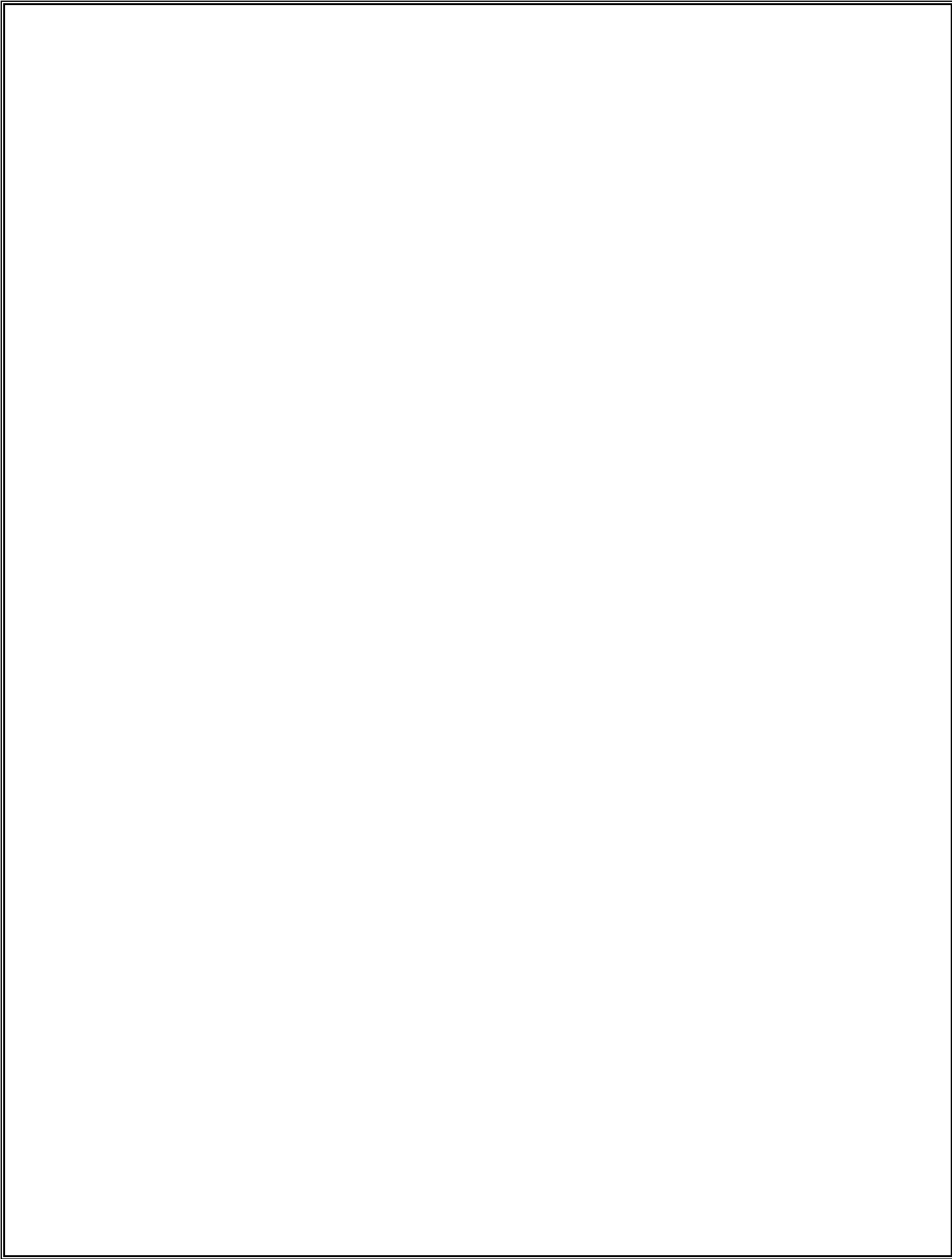
Result Grid

	id	email	gender	mobile	name	password
▶	12	mayursolanki66@gmail.com	male	07927528592	mayur	27528592
	21	jaybahuchar2510@gmail.com	male	3724444035	BharatBhai	9724444035
	22	tara@gmail.com	female	4546597897	Tara	123456789
*	NULL	NULL	NULL	NULL	NULL	NULL

Administration Schemas

Information

No object selected



SHOES TABLE

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- flyway
- flyway2
- hibernatedemo
- mecommerce**
 - Tables
 - hibernate_sequence
 - tbl_admin
 - tbl_customer
 - tbl_shoes
 - Views
 - Stored Procedures
 - Functions
- shoppingssystem
- sys

Administration Schemas

Information

No object selected

tbl_customer tbl_shoes tbl_admin hibernate_sequence tbl_customer tbl_shoes tbl_shoes hibernate_sequence tbl_admin hibernate_sequence tbl_admin tbl_customer **tbl_shoes**

1 • `SELECT * FROM mecommerce.tbl_shoes;`

Limit to 1000 rows

Result Grid

Filter Rows:

Edit: Export/Import: Wrap Cell Contents: 15

	id	img_url	shoes_category	shoes_name	shoes_price	shoes_img
▶	16	data	casual	Bata	3999	/j/4AAQ9kZRpABAQAAQABAAD/2wEDABAL...
	17	data	traditional	adidas_M	2999	/j/4AAQ9kZRpABAQAAQABAAD/2wCEAAk...
	18	data	sports	Nike_M	4999	/j/4AAQ9kZRpABAQAAQABAAD/2wEDABAL...
	19	data	sports	PUMA_I	1999	/j/4AAQ9kZRpABAQAAQABAAD/2wEDAAk...
	20	data	sports	Nike_Mode	999	/j/4AAQ9kZRpABAQAAQABAAD/2wEDABAL...
*	data	data	sports	Skechers	3999	/j/4AAQ9kZRpABAQAAQABAAD/2wEDAAk...

WEBSITE LOOK :

