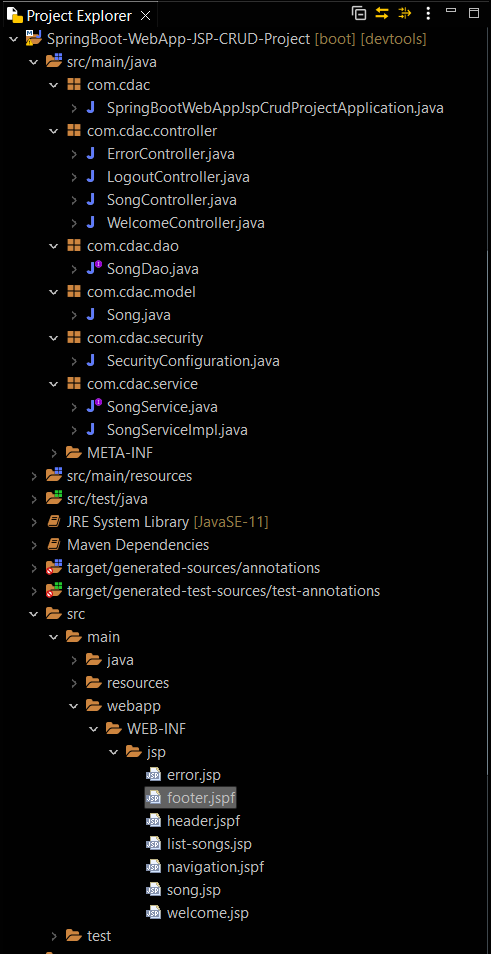
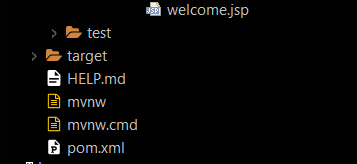
**Project Structure**

**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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.7.10</version>

<relativePath /> <!-- lookup parent from repository -->

</parent>

<groupId>com.cdac</groupId>

<artifactId>SpringBoot-WebApp-JSP-CRUD-Project</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>SpringBoot-WebApp-JSP-CRUD-Project</name>

<description>Spring Boot Project</description>

<properties>

<java.version>11</java.version>

</properties>

<dependencies>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

</dependency>

<dependency>

<groupId>org.webjars</groupId>

<artifactId>bootstrap</artifactId>

<version>3.3.6</version>

</dependency>

<dependency>

<groupId>org.webjars</groupId>

<artifactId>bootstrap-datepicker</artifactId>

<version>1.0.1</version>

</dependency>

<dependency>

<groupId>org.webjars</groupId>

<artifactId>jquery</artifactId>

<version>1.9.1</version>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-security</artifactId>

</dependency>

<dependency>

<groupId>org.apache.tomcat.embed</groupId>

<artifactId>tomcat-embed-jasper</artifactId>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

<groupId>com.mysql</groupId>

<artifactId>mysql-connector-j</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

<groupId>jakarta.xml.bind</groupId>

<artifactId>jakarta.xml.bind-api-test</artifactId>

<version>3.0.1</version>

</dependency>

<!-- Thanks for using https://jar-download.com -->

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

**Application.properties**

## Spring view resolver set up

spring.mvc.view.prefix=/WEB-INF/jsp/

spring.mvc.view.suffix=.jsp

## Spring DATASOURCE (DataSourceAutoConfiguration & DataSourceProperties)

spring.datasource.url = jdbc:mysql://localhost:3306/pune?useSSL=false

spring.datasource.username = root

spring.datasource.password = cdac

## Hibernate Properties

# The SQL dialect makes Hibernate generate better SQL for the chosen database

spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5InnoDBDialect

# Hibernate ddl auto (create, create-drop, validate, update)

spring.jpa.hibernate.ddl-auto = update

**Song.java**

package com.cdac.model;

import java.util.Date;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "song\_table2")

public class Song {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private int id;

private String userName;

private String songName;

private String singerName;

private String movieName;

private Date releseDate;

public Song() {

super();

}

public Song(int id) {

super();

this.id = id;

}

public Song(String user, String songName, String singerName, String movieName, Date releseDate) {

super();

this.userName = user;

this.songName = songName;

this.singerName = singerName;

this.movieName = movieName;

this.releseDate = releseDate;

}

public Song(int id, String user, String songName, String singerName, String movieName, Date releseDate) {

super();

this.id = id;

this.userName = user;

this.songName = songName;

this.singerName = singerName;

this.movieName = movieName;

this.releseDate = releseDate;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public String getSongName() {

return songName;

}

public void setSongName(String songName) {

this.songName = songName;

}

public String getSingerName() {

return singerName;

}

public void setSingerName(String singerName) {

this.singerName = singerName;

}

public String getMovieName() {

return movieName;

}

public void setMovieName(String movieName) {

this.movieName = movieName;

}

public Date getReleseDate() {

return releseDate;

}

public void setReleseDate(Date releseDate) {

this.releseDate = releseDate;

}

}

**SongDao.java**

package com.cdac.dao;

import java.util.List;

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

import org.springframework.stereotype.Repository;

import com.cdac.model.Song;

@Repository

public interface SongDao extends JpaRepository<Song, Integer>{

List<Song> getSongsByMovieName(String movieName );

List<Song> getSongsBySingerName(String singerName);

List<Song> getSongsByUserName(String userName);

}

**SongService.java**

package com.cdac.service;

import java.util.Date;

import java.util.List;

import java.util.Optional;

import com.cdac.model.Song;

public interface SongService {

void add(Song song);

List <Song> getSongsByUserName(String userName);

void update(Song song);

void deleteById(int id);

Optional < Song > getById(int id);

List<Song> getAll();

void addSong(String userName, String songName, String singerName, String movieName, Date releseDate);

List<Song> getSongsBySingerName(String singerName);

List<Song> getSongsByMovieName(String movieName );

}

**SongServiceImpl.java**

package com.cdac.service;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

import com.cdac.dao.SongDao;

import com.cdac.model.Song;

@Service

public class SongServiceImpl implements SongService {

@Autowired

private SongDao songDao;

@Override

public void add(Song song) {

songDao.save(song);

}

@Override

public void addSong(String userName, String songName, String singerName, String movieName, Date releseDate) {

songDao.save(new Song(userName, songName, singerName, movieName, releseDate));

}

@Override

public void update(Song song) {

songDao.save(song);

}

@Override

public void deleteById(int id) {

songDao.deleteById(id);

}

@Override

public Optional<Song> getById(int id) {

return songDao.findById(id);

}

@Override

public List<Song> getAll() {

Iterable<Song> itr = songDao.findAll();

List<Song> songList = new ArrayList<Song>();

itr.forEach(element -> songList.add(element));

return songList;

}

@Override

public List<Song> getSongsByUserName(String userName) {

return songDao.getSongsByUserName(userName);

}

@Override

public List<Song> getSongsBySingerName(String singerName) {

return songDao.getSongsBySingerName(singerName);

}

@Override

public List<Song> getSongsByMovieName(String movieName) {

return songDao.getSongsByMovieName(movieName);

}

}

**WelcomeController.java**

package com.cdac.controller;

import org.springframework.security.core.context.SecurityContextHolder;

import org.springframework.security.core.userdetails.UserDetails;

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

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

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

@Controller

public class WelcomeController {

@RequestMapping(value = "/", method = RequestMethod.GET)

public String showWelcomePage(ModelMap model) {

model.put("name", getLoggedinUserName());

return "welcome";

}

private String getLoggedinUserName() {

Object principal = SecurityContextHolder.getContext().getAuthentication().getPrincipal();

if (principal instanceof UserDetails) {

return ((UserDetails) principal).getUsername();

}

return principal.toString();

}

}

**LogoutController.java**

package com.cdac.controller;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.security.core.Authentication;

import org.springframework.security.core.context.SecurityContextHolder;

import org.springframework.security.web.authentication.logout.SecurityContextLogoutHandler;

import org.springframework.stereotype.Controller;

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

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

@Controller

public class LogoutController {

@RequestMapping(value = "/logout", method = RequestMethod.GET)

public String logout(HttpServletRequest request, HttpServletResponse response) {

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

if (authentication != null) {

new SecurityContextLogoutHandler().logout(request, response, authentication);

}

return "redirect:/";

}

}

**ErrorController.java**

package com.cdac.controller;

import javax.servlet.http.HttpServletRequest;

import org.springframework.stereotype.Controller;

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

import org.springframework.web.servlet.ModelAndView;

@Controller("error")

public class ErrorController {

@ExceptionHandler(Exception.class)

public ModelAndView handleException(HttpServletRequest request, Exception ex) {

ModelAndView mv = new ModelAndView();

mv.addObject("exception", ex.getLocalizedMessage());

mv.addObject("url", request.getRequestURL());

mv.setViewName("error");

return mv;

}

}

**SongController.java**

package com.cdac.controller;

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

import org.springframework.security.core.context.SecurityContextHolder;

import org.springframework.security.core.userdetails.UserDetails;

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

import org.springframework.validation.BindingResult;

import org.springframework.validation.annotation.Validated;

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

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

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

import com.cdac.model.Song;

import com.cdac.service.SongService;

@Controller

public class SongController {

@Autowired

private SongService songService;

@RequestMapping(value = "/list-songs", method = RequestMethod.GET)

public String showSongs(ModelMap model) {

String UserName = getLoggedInUserName(model);

model.put("songs", songService.getSongsByUserName(UserName));

return "list-songs";

}

private String getLoggedInUserName(ModelMap model) {

Object principal = SecurityContextHolder.getContext().getAuthentication().getPrincipal();

if (principal instanceof UserDetails) {

return ((UserDetails) principal).getUsername();

}

return principal.toString();

}

@RequestMapping(value = "/add-song", method = RequestMethod.GET)

public String showAddSongPage(ModelMap model) {

model.addAttribute("song", new Song());

return "song";

}

@RequestMapping(value = "/delete-song", method = RequestMethod.GET)

public String deleteSong(@RequestParam int id) {

songService.deleteById(id);

// service.deleteTodo(id);

return "redirect:/list-songs";

}

@RequestMapping(value = "/update-song", method = RequestMethod.GET)

public String showUpdateSongPage(@RequestParam int id, ModelMap model) {

Song song = songService.getById(id).get();

model.put("song", song);

return "song";

}

@RequestMapping(value = "/update-song", method = RequestMethod.POST)

public String updateSong(ModelMap model, @Validated Song song, BindingResult result) {

if (result.hasErrors()) {

return "song";

}

song.setUserName(getLoggedInUserName(model));

songService.update(song);

return "redirect:/list-songs";

}

@RequestMapping(value = "/add-song", method = RequestMethod.POST)

public String addSong(ModelMap model, @Validated Song song, BindingResult result) {

if (result.hasErrors()) {

return "song";

}

song.setUserName(getLoggedInUserName(model));

songService.add(song);

return "redirect:/list-songs";

}

}

**Welcome.jsp**

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

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

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

<title>Home Page</title>

</head>

<body>

<div class="container">

<%@ **include** file="header.jspf"%>

<%@ **include** file="navigation.jspf"%>

<div class="panel panel-primary">

<div class="panel-heading">Home Page</div>

<div class="panel-body">

Welcome ${name} <br> <a href="/list-songs">Click here</a> to

manage your Songs.

</div>

</div>

</div>

<%@ **include** file="footer.jspf"%>

</body>

</html>

**Song.jsp**

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

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

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

<title>Add song</title>

</head>

<body>

<%@ **include** file="header.jspf"%>

<%@ **include** file="navigation.jspf"%>

<div class="container">

<div class="row">

<div class="col-md-6 col-md-offset-3 ">

<div class="panel panel-primary">

<div class="panel-heading">Add Song</div>

<div class="panel-body">

<**form:form** method="post" modelAttribute="song">

<**form:hidden** path="id" />

<fieldset class="form-group">

<**form:label** path="songName">Song Name</**form:label**>

<**form:input** path="songName" type="text" class="form-control"

required="required" />

<**form:errors** path="songName" cssClass="text-warning" />

</fieldset>

<fieldset class="form-group">

<**form:label** path="singerName">Singer Name</**form:label**>

<**form:input** path="singerName" type="text" class="form-control"

required="required" />

<**form:errors** path="singerName" cssClass="text-warning" />

</fieldset>

<fieldset class="form-group">

<**form:label** path="movieName">Movie Name</**form:label**>

<**form:input** path="movieName" type="text" class="form-control"

required="required" />

<**form:errors** path="movieName" cssClass="text-warning" />

</fieldset>

<button type="submit" class="btn btn-success">Save</button>

</**form:form**>

</div>

</div>

</div>

</div>

</div>

<%@ **include** file="footer.jspf"%>

</body>

</html>

**List-songs.jsp**

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

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

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

<title>Songs list</title>

</head>

<body>

<%@ **include** file="header.jspf"%>

<%@ **include** file="navigation.jspf"%>

<div class="container">

<div class="panel panel-primary">

<div class="panel-heading">

<h3 align="center">List of Songs</h3>

</div>

<div class="panel-body">

<table class="table table-striped">

<thead>

<tr>

<th width="25%">Song Name</th>

<th width="25%">Singer Name</th>

<th width="25%">Movie Name</th>

<th width="25%">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Action</th>

</tr>

</thead>

<tbody>

<**c:forEach** items="${songs}" var="songs">

<tr>

<td>${songs.songName}</td>

<td>${songs.singerName}</td>

<td>${songs.movieName}</td>

<td><a type="button" class="btn btn-primary"

href="/update-song?id=${songs.id}">Update</a> <a type="button"

class="btn btn-danger" href="/delete-song?id=${songs.id}">Delete</a></td>

</tr>

</**c:forEach**>

</tbody>

</table>

</div>

</div>

</div>

<div align="center">

<br> <a type="button" class="btn btn-primary btn-lg"

href="/add-song">Add Song</a>

</div>

<%@ **include** file="footer.jspf"%>

</body>

</html>

**Header.jsp**

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

pageEncoding="ISO-8859-1"%>

<%@ **taglib** uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<%@ **taglib** uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>

<%@**taglib** uri="http://www.springframework.org/tags/form" prefix="form"%>

<!DOCTYPE html>

<html>

<head>

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

<title>Song Management</title>

<link href="webjars/bootstrap/3.3.6/css/bootstrap.min.css"

rel="stylesheet">

<style>

.footer {

position: fixed;

left: 0;

bottom: 0;

width: 100%;

background-color: black;

color: white;

height: 100px;

text-align: center;

}

</style>

</head>

<body>

</body>

</html>

**Footer.jsp**

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

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

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

<title>Insert title here</title>

<script src="webjars/jquery/1.9.1/jquery.min.js"></script>

<script src="webjars/bootstrap/3.3.6/js/bootstrap.min.js"></script>

<script

src="webjars/bootstrap-datepicker/1.0.1/js/bootstrap-datepicker.js"></script>

<script>

$('#targetDate').datepicker({

format : 'dd/mm/yyyy'

});

</script>

</head>

<body>

<div class="footer">

<h3>

<a href="/">Spring Boot Tutorial</a>

</h3>

</div>

</body>

</html>

**Navigation.jsp**

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

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

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

<title>Insert title here</title>

</head>

<body>

<nav role="navigation" class="navbar navbar-default">

<div class="">

<a href="/" class="navbar-brand">Musical World</a>

</div>

<div class="navbar-collapse">

<ul class="nav navbar-nav">

<li class="active"><a href="/">Home</a></li>

<li><a href="/list-songs">Songs</a></li>

</ul>

<ul class="nav navbar-nav navbar-right">

<li><a href="/logout">Logout</a></li>

</ul>

</div>

</nav>

</body>

</html>

**Error.jsp**

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

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

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

<title>error</title>

</head>

<body>

<%@ **include** file="header.jspf"%>

<%@ **include** file="navigation.jspf"%>

<div class="container">

<h1>Something went wrong...</h1>

</div>

<%@ **include** file="footer.jspf"%>

</body>

</html>

**SpringBootHibernateCRUDAppApplication.java**

package com.cdac;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

*@SpringBootApplication*

public class SpringBootHibernateCrudAppApplication {

public static void main(String[] args) {

SpringApplication.*run*(SpringBootHibernateCrudAppApplication.class, args);

}

}

**Web Browser Screenshots**

