**POM.xml file :**

<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/maven-v4\_0\_0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.operation</groupId>

<artifactId>productcrudapp</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>productcrudapp Maven Webapp</name>

<url>http://maven.apache.org</url>

<dependencies>

<!-- Step 1 : web-mvc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.3.20</version>

</dependency>

<!-- Step 2 : Orm -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>5.3.20</version>

</dependency>

<!-- Step 3 hibernate -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.6.5.Final</version>

</dependency>

<!-- Step 4 My sql connector -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.28</version>

</dependency>

<!--Step 5 jstl for using c:tags etc-->

<dependency>

<groupId>jstl</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<finalName>productcrudapp</finalName>

</build>

</project>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Web.xml file**

<!DOCTYPE web-app PUBLIC

"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"

"http://java.sun.com/dtd/web-app\_2\_3.dtd" >

<web-app>

<display-name>Archetype Created Web Application</display-name>

<!--Step 6 Configruee the servet and map it with url -->

<servlet>

<servlet-name>spring</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>spring</servlet-name>

<url-pattern>/</url-pattern> <!-- / its handling all the url not specific statrign with /spring -->

</servlet-mapping>

</web-app>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**spring-servlet.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:p=*"http://www.springframework.org/schema/p"*

xmlns:tx=*"http://www.springframework.org/schema/tx"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context.xsd*

*http://www.springframework.org/schema/tx*

*http://www.springframework.org/schema/tx/spring-tx.xsd*

*"*>

<tx:annotation-driven />

<!-- Step 7 Configure the bean of internal view resolver -->

<context:component-scan base-package=*"productcrudapp"* />

<!-- This is step 7(b) of view resolver which helps to create the name of path at the run time -->

<bean name=*"viewResolver"* class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<property name=*"prefix"* value=*"/WEB-INF/views/"* />

<property name=*"suffix"* value=*".jsp"* />

</bean>

<!-- Step 8 Configure the bean : connection with database -->

<!-- Step 8 (a) Configure database -DriverManagerDateSource -->

<bean name=*"ds"* class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>

<property name=*"driverClassName"* value=*"com.mysql.cj.jdbc.Driver"* />

<property name=*"url"* value=*"jdbc:mysql://localhost:3306/shivampractice"* />

<property name=*"username"* value=*"root"* />

<property name=*"password"* value=*"root"* />

</bean>

<!-- Step 8 (b) Configure database LocalSessionFactoryBean -->

<bean name=*"factory"* class=*"org.springframework.orm.hibernate5.LocalSessionFactoryBean"* >

<property name=*"dataSource"* ref=*"ds"* />

<property name=*"hibernateProperties"* >

<props>

<prop key=*"hibernate.dialect"*>org.hibernate.dialect.MySQL57Dialect</prop>

<prop key=*"hibernate.show\_sql"*>true</prop>

<prop key=*"hibernate.hbm2ddl.auto"*>update</prop>

</props>

</property>

<property name=*"annotatedClasses"*>

<list>

<!— <value>productcrudapp.model.Product</value>

comment this value initially otherwise error will come -->

</list>

</property>

</bean>

<!-- Step 8 (c) Configure database HibernateTemplate -->

<bean name=*"hibernateTemplate"* class=*"org.springframework.orm.hibernate5.HibernateTemplate"* >

<property name=*"sessionFactory"* ref=*"factory"* />

</bean>

<!-- Step 9 Configure database HibernateTransactionManager also enable tx annotaion-driver at top Needed for writing operation in DataBasee-->

<bean name=*"transactionManager"* class=*"org.springframework.orm.hibernate5.HibernateTransactionManager"* >

<property name=*"sessionFactory"* ref=*"factory"*/>

</bean>

</beans>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**HomeController.java File**

**package** productcrudapp.controller;

**import** java.util.List;

**import** javax.servlet.http.HttpServletRequest;

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

**import** org.springframework.stereotype.Controller;

**import** org.springframework.ui.Model;

**import** org.springframework.web.bind.annotation.ModelAttribute;

**import** org.springframework.web.bind.annotation.PathVariable;

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

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

**import** org.springframework.web.servlet.view.RedirectView;

**import** productcrudapp.dao.ProductDao;

**import** productcrudapp.model.Product;

@Controller

**public** **class** HomeController {

//STEP 16 : save data using ProductDao class

@Autowired

ProductDao productDao;

/\*Used it just for the trial

\* @RequestMapping("/") public String home() { return "index";

\*

\* }

\*/

// Step 12 : add create handler and create a base.jsp (step 13)

@RequestMapping("/add-product")

**public** String addProduct(Model m) {

m.addAttribute("title","ADD Product runtime Page");

**return** "add-product-form";

}

//STEP 15 : handle form submission and redirect to other page using HttpServletRequest : context\_path , also taking the ModelAttrbute to save data in Product Object , as well receive it as whole & bind it to send to Next Page

@RequestMapping(value="/handle-product" , method=RequestMethod.***POST***)

**public** RedirectView handleProduct(@ModelAttribute Product product ,HttpServletRequest request)

{

System.***out***.println(product);

RedirectView redirectView= **new** RedirectView();

// save the details of form in database using save method of ProductDao class

//Step 16 //ALso step 24 of Save and Update

productDao.createProduct(product); //(this internally uses the save method using hibernateTemplate obj)

redirectView.setUrl(request.getContextPath()+"/");

**return** redirectView;

}

//STEP 16 : DISPLAY ALL RECORDS ; in same home url : using the getProductsMethod of DAo class (which uses loadAll method via hibernate template obj)

//also next step 18 to get the obj in indexx.jsp

@RequestMapping("/")

**public** String home( Model m)

{

List<Product> products= productDao.getProducts();

m.addAttribute("products",products);

**return** "index";

}

//STEP 19 : Delete handler to handler when delete button give href : /delete/{id} So need to use RedirectView

@RequestMapping(value="/delete/{productid}")

**public** RedirectView deleteProduct(@PathVariable("productid") **int** pid,HttpServletRequest request)

{

RedirectView redirectView= **new** RedirectView();

//call the delete method of DAO (step 20) which internally use hibernate template obj to delete product obj with given id

**this**.productDao.deleteProduct(pid);

redirectView.setUrl(request.getContextPath()+"/");

**return** redirectView;

}

//STEP 21 : Updating handler

@RequestMapping("/update/{product\_id}")

**public** String updateForm(@PathVariable("product\_id") **int** pid , Model m)

{

Product temProduct= **this**.productDao.getProduct(pid); //step22

m.addAttribute("product",temProduct);

//used model attribute to send the details to update form handler to display details of previous product description ,

//step 23 in jsp of update\_form // same as add-product html ,

// in step 24

//IMPORTANT : since we are handling update and save at same controller handleproduct when url "/handler-product" so need to change that method of ProductDao to "saveandUpdate "form

**return** "update\_form";

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ProductDao.java File**

**package** productcrudapp.dao;

**import** java.util.List;

**import** javax.transaction.Transactional;

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

**import** org.springframework.orm.hibernate5.HibernateTemplate;

**import** org.springframework.stereotype.Component;

**import** productcrudapp.model.Product;

@Component

**public** **class** ProductDao { //Step 10. create DAO class

@Autowired

HibernateTemplate hibernateTemplate;

//Step 12. create method

@Transactional

**public** **void** createProduct(Product p) //Step 11 : create product class

{

**this**.hibernateTemplate.saveOrUpdate(p); // STEP 24 :handle both saving and updateing step : last

}

//Step 17: create getAll products Method using getAll method()

**public** List<Product> getProducts()

{

List<Product> products = **this**.hibernateTemplate.loadAll(Product.**class**);

**return** products;

}

//Step 20 : create delete method

@Transactional

**public** **void** deleteProduct(**int** pid)

{

// get the product obj using the pid then use hibernatetemplate to delete

Product tempproduct= **this**.hibernateTemplate.load(Product.**class**,pid);

//now delete

**this**.hibernateTemplate.delete(tempproduct);

}

// Step 22 get the single product details to be displayed on update page

**public** Product getProduct(**int** pid)

{

**return** **this**.hibernateTemplate.get(Product.**class**,pid);

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Product.java File**

**package** productcrudapp.model;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

//Step 11 : create product class

@Entity // since we have used @Entity it will create table in database , so need to give

// it in annotated classes property of sessionFactory in .xml file

**public** **class** Product {

@Id

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

**private** **int** id;

**private** String name;

**private** String description;

**private** **long** price;

**public** Product() {

**super**();

}

**public** Product(**int** id, String name, String description, **long** price) {

**super**();

**this**.id = id;

**this**.name = name;

**this**.description = description;

**this**.price = price;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getDescription() {

**return** description;

}

**public** **void** setDescription(String description) {

**this**.description = description;

}

**public** **long** getPrice() {

**return** price;

}

**public** **void** setPrice(**long** price) {

**this**.price = price;

}

@Override

**public** String toString() {

**return** "Product [id=" + id + ", name=" + name + ", description=" + description + ", price=" + price + "]";

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Base.jsp File**

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

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

<%@ page isELIgnored=*"false"* %>

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

<!--STEP 13 Create a base.jsp which provide css/ javascript to all other jsp pages when included -->

<!doctype html>

<html lang=*"en"*>

<head>

<!-- Required meta tags : view port makes website responsive -->

<meta charset=*"utf-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1, shrink-to-fit=no"*>

<!-- Bootstrap CSS -->

<link rel=*"stylesheet"* href=*"https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css"* integrity=*"sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"* crossorigin=*"anonymous"*>

<!-- -->

<link rel=*"stylesheet"* href=*"https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css"* integrity=*"sha512-KfkfwYDsLkIlwQp6LFnl8zNdLGxu9YAA1QvwINks4PhcElQSvqcyVLLD9aMhXd13uQjoXtEKNosOWaZqXgel0g=="* crossorigin=*"anonymous"* referrerpolicy=*"no-referrer"* />

<!-- \*\*\*\*\*\*\*\*\*\* We have include jstl taglib for this c:out -->

<title> <c:out value=*"*${title}*"*> Product CRUD APP </c:out></title>

</head>

<body>

<!-- <h1>Hello, world!</h1> -->

<!-- Optional JavaScript -->

<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src=*"https://code.jquery.com/jquery-3.2.1.slim.min.js"* integrity=*"sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"* crossorigin=*"anonymous"*></script>

<script src=*"https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"* integrity=*"sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"* crossorigin=*"anonymous"*></script>

<script src=*"https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js"* integrity=*"sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"* crossorigin=*"anonymous"*></script>

</body>

</html>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**index.jsp File**

<html>

<head>

<%@ include file=*"./base.jsp"*%>

<%@page isELIgnored=*"false"*%>

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

</head>

<!-- STEP 18 : Display the details send from display all (Home Controller ) using table

using the jstl (c:out loops ) and expression language(${p.name}) for the loop to display table rows

Also two button are also given (delete , update ) which takes to suitable handler method through href to perfrom the given task

Next step is to make the handler for delete and update (step 19)

-->

<body>

<div class=*"container mt-3"*>

<div class=*"row"*>

<div class=*"col-md-12"*>

<h1 class=*"text-center mb-3"*>Welcome to Product App</h1>

<table class=*"table"*>

<thead class=*"thead-dark"*>

<tr>

<th scope=*"col"*>S\_NO</th>

<th scope=*"col"*>Product Name</th>

<th scope=*"col"*>Description</th>

<th scope=*"col"*>Price</th>

<th scope=*"col"*>Action</th>

</tr>

</thead>

<tbody>

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

<tr>

<th scope=*"row"*>Learn Id ${p.id}</th>

<td>${p.name}</td>

<td>${p.description}</td>

<td>&#8377 ${p.price}</td>

<td>

<a href=*"delete/*${p.id}*"*><i class=*"fas fa-trash text-danger"* style="font-size:*30px*;" aria-hidden=*"true"*></i></a>

<a href=*"update/*${p.id}*"*><i class=*"fas fa-pen-nib text-primary"* style="font-size:*30px*;" aria-hidden=*"true"*></i></a>

</td>

</tr>

</c:forEach>

</tbody>

</table>

<div class=*"container text-center"*>

<a href=*"add-product"*

class=*"btn btn-outline-success"*>Add Product</a>

</div>

</div>

</div>

</div>

</body>

</html>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Add-product.jsp File**

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

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

<%@ include file=*"./base.jsp"* %>

<!DOCTYPE html>

<html>

<head>

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

<!--

The title will come from another file

<title>Insert title here</title> -->

</head>

<body>

<!-- STEP 14 : DEsingned the HTML form (copied) and also added functionality of back button

Also need to handle submission of form in handle-product handler method in Home Controller(STep 15)

>

<!-- <h1>This is add product form</h1> -->

<div class=*"container mt-3"*>

<div class=*"row"*>

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

<h1 class=*"text-center mb-3"*>Fill the product detail</h1>

<form action=*"handle-product"* method=*"post"*>

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

<label for=*"name"*>Product Name</label>

<input type=*"text"*

class=*"form-control"*

id=*"name"*

aria-describedby=*"emailHelp"*

name=*"name"*

placeholder=*"Enter the product name here"*>

</div>

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

<label for=*"description"*>Product Description</label>

<textarea

class=*"form-control"*

id=*"description"*

rows=*"5"*

name=*"description"*

placeholder=*"Enter the product description"*>

</textarea>

</div>

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

<label for=*"price"*>Product Price</label>

<input type=*"text"*

placeholder=*"Enter Product Price"*

name=*"price"*

class=*"form-control"* id=*"price"*>

</div>

<div class=*"container text-center"*>

<a href=*"*${pageContext.request.contextPath }*/"*

class=*"btn btn-outline-danger"*>Back</a> <!-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

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

</div>

</form>

</div>

</div>

</div>

</body>

</html>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Update-form.jsp File**

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

pageEncoding=*"ISO-8859-1"*

isELIgnored=*"false"* %>

<!DOCTYPE html>

<html>

<head>

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

<!--

The title will come from another file

<title>Insert title here</title> -->

<%@include file=*"./base.jsp"* %>

</head>

<body>

<!-- STEP 23 ::

<form action="${pageContext.request.contextPath }/handle-product" method="post">

\*In form action we have used the context so path is not lost by clicling update button

\*Also we have displayed value of previous filling usnig

value="${product.name}">

\* also in functionality of back button used contextPAth to return to home page

LAst step 23

We are handling the save and update request at same method of ProductDao class, so follow on this :

while updateing we are also passing the previous id to that method , so it will update only if present previously.

we make type of id as hidden so not displayed on page

-->

<!-- <h1>This is add product form</h1> -->

<div class=*"container mt-3"*>

<div class=*"row"*>

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

<h1 class=*"text-center mb-3"*>Update product Details</h1>

<form action=*"*${pageContext.request.contextPath }*/handle-product"* method=*"post"*>

<!-- STEP 24 -->

<input type=*"hidden"* value=*"*${product.id}*"* name=*"id"*>

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

<label for=*"name"*>Product Name</label>

<input type=*"text"*

class=*"form-control"*

id=*"name"*

aria-describedby=*"emailHelp"*

name=*"name"*

placeholder=*"Enter the product name here"*

value=*"*${product.name}*"*>

</div>

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

<label for=*"description"*>Product Description</label>

<textarea

class=*"form-control"*

id=*"description"*

rows=*"5"*

name=*"description"*

placeholder=*"Enter the product description"*>${product.description}

</textarea>

</div>

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

<label for=*"price"*>Product Price</label>

<input type=*"text"*

placeholder=*"Enter Product Price"* name=*"price"*

class=*"form-control"* id=*"price"*

value=*"*${product.price}*"*

>

</div>

<div class=*"container text-center"*>

<a href=*"*${pageContext.request.contextPath }*/"*

class=*"btn btn-outline-danger"*>Back</a>

<button type=*"submit"* class=*"btn btn-warning"*>Update </button>

</div>

</form>

</div>

</div>

</div>

</body>

</html>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**File**