**Phase – II**

**Practice Project – III**

**Add New Product**

**Sai Krishna Rajagopal**

**22-03-2022**

**Index.jsp:**

<html>

<link rel=*"stylesheet"*

href=*"https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"*

integrity=*"sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"*

crossorigin=*"anonymous"*>

</head>

<body>

<h3>Search for a product by ID</h3>

<div style="width: *40%*; height: *30%*; margin: *20px*;">

<form class=*"form-inline"* action=*"getProduct"*>

<div class=*"form-group mx-sm-3 mb-2"*>

<label class=*"sr-only"*>Password</label>

<!-- <input type="text" name="pid" class="form-control" placeholder="ID number"> -->

<input type=*"number"* id=*"replyNumber"* name=*"pid"* min=*"0"*

data-bind=*"value:replyNumber"* Placeholder=*"Product ID"* />

</div>

<button type=*"submit"* class=*"btn btn-warning mb-2"*>Find</button>

</form>

</div>

<!-- Show the user if the product saved successfully or not -->

<%

Object strMessage = request.getAttribute("message");

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

%>

<h3>

<%

out.println(strMessage.toString());

%>

</h3>

<%

}

%>

<!-- Alert the client that the id should be number not text -->

<%

Object idmessage = request.getAttribute("idMessage");

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

%>

<h3>

<%

out.println(idmessage.toString());

%>

</h3>

<%

}

%>

<h3>Add a New product:</h3>

<div style="width: *40%*; height: *50%*; margin: *30px*;">

<form class=*"form-inline"* action=*"saveProduct"*>

<div class=*"form-group mx-sm-3 mb-2"*>

<label class=*"sr-only"*>Product ID</label> <input type=*"number"*

id=*"replyNumber"* name=*"sid"* min=*"0"* data-bind=*"value:replyNumber"*

placeholder=*"Product ID"* />

</div>

<br/><br/>

<div class=*"form-group mx-sm-3 mb-2"*>

<label class=*"sr-only"*>Product Name</label> <input type=*"text"*

class=*"form-control"* name=*"sname"* placeholder=*"Product name"*>

</div>

<br/><br/>

<div class=*"form-group mx-sm-3 mb-2"*>

<label class=*"sr-only"*>Product Price</label> <input type=*"text"*

class=*"form-control"* name=*"scompany"* placeholder=*"Product Price"*>

</div>

<br/><br/>

<button type=*"submit"* class=*"btn btn-warning mb-2"*>submit</button>

</form>

</div>

</body>

</html>

<!-- <input type="number" id="replyNumber" name="pid" min="0"

data-bind="value:replyNumber" />

-->

**ShowProducts.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>

<h3>Add a New product:</h3>

<%

Object strMessage = request.getAttribute("message");

**if**( strMessage != **null**) { %>

<h3><% out.println(strMessage.toString()); %></h3>

<% }%>

<div style="width: *40%*; height: *100%*; margin: *20px*;">

<form class=*"form-inline"* action=*"saveProduct"*>

<div class=*"form-group mx-sm-3 mb-2"*>

<label class=*"sr-only"*>Product ID</label>

<input type=*"number"* id=*"replyNumber"* name=*"sid"* min=*"0"*

data-bind=*"value:replyNumber"* />

</div>

<div class=*"form-group mx-sm-3 mb-2"*>

<label class=*"sr-only"*>Product Name</label> <input type=*"text"*

class=*"form-control"* name=*"sname"* placeholder=*"Product Name"*>

</div>

<div class=*"form-group mx-sm-3 mb-2"*>

<label class=*"sr-only"*>Product Price</label> <input type=*"text"*

class=*"form-control"* name=*"scompany"* placeholder=*"Product Price"*>

</div>

<button type=*"submit"* class=*"btn btn-warning mb-2"*>submit</button>

</form>

</div>

</body>

</html>

**Servlet:**

**Product Controller:**

package com.file.web;

import java.io.IOException;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.file.web.dao.ProductDAO;

import com.file.web.model.Product;

/\*\*

\* Servlet implementation class ProductController

\*/

public class ProductController extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

// Create a dispatcher and get the Product id from index.js

// Create Product DAO to get the data from the database

// Inside Product DAO Create Product object and save the data from the Database in it

// Pass the Product object to showProduct.jsp to the user to see it.

try {

RequestDispatcher reqDis = request.getRequestDispatcher("showProducts.jsp");

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

// Create Product DAO to get the data from the database

// Inside Product DAO Create Product object and save the data from the Database in it

ProductDAO pDAO = new ProductDAO();

Product prd = pDAO.getProduct(pid);

// Pass the Product object to showProduct.jsp to the user to see it.

request.setAttribute("product", prd);

reqDis.forward(request, response);

} catch (Exception e) {

// TODO: handle exception

RequestDispatcher reqDisforID = request.getRequestDispatcher("index.jsp");

request.setAttribute("idMessage", "Please Product ID must be a posative number only.");

reqDisforID.forward(request, response);

}

}

}

**Save Product Controller:**

package com.file.web;

import java.io.IOException;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.file.web.dao.ProductDAO;

import com.file.web.model.Product;

/\*\*

\* Servlet implementation class ProductController

\*/

public class SaveProductController extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

// Create a dispatcher and get the Product id,Product name, Product company from user (index.js)

// Create Product DAO to save the entered data into the database

// Inside Product DAO Create Product object and save the data from the user in it

// Notify the user if the product saved successfully/not, exist/not

// Alert the user, if the Product ID entered was a letters instead of positive numbers.

RequestDispatcher reqDisforID = request.getRequestDispatcher("index.jsp");

try {

// Create a dispatcher and get the Product id,Product name, Product company from

// user (index.js)

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

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

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

// Create Product DAO to save the entered data into the database

ProductDAO pDAO = new ProductDAO();

// Inside Product DAO Create Product object and save the data from the user in

// it

Product checkresult = pDAO.saveProduct(sid, sname, scompany);

if (checkresult == null) {

// Notify the user if the product saved successfully/not, exist/not

RequestDispatcher reqDis = request.getRequestDispatcher("index.jsp");

request.setAttribute("message", "The Product added successfully");

reqDis.forward(request, response);

} else {

// Notify the user if the product exist/not

System.out.println("it is already exist");

RequestDispatcher reqDis = request.getRequestDispatcher("index.jsp");

request.setAttribute("message", "The Product already exists! ");

reqDis.forward(request, response);

}

} catch (Exception e) {

// Alert the user, if the Product ID entered was a letters instead of positive

// numbers.

request.setAttribute("idMessage", "Please Product id must be a posative a number only.");

reqDisforID.forward(request, response);

}

}

}

**Java Class:**

**Product Dao.java**

package com.file.web.dao;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import com.file.web.model.Product;

public class ProductDAO {

// Create product method :

// 1-Connect to the database

// 2-Execute a query to get data from the database

// 3-Save the data in a new product object.

// 4-Return the product to be used in the (Product Controller)

public Product getProduct(int pid) {

Product product = new Product();

try {

// 1-Connect to the database

Class.forName("com.mysql.cj.jdbc.Driver");

Connection connectionToDB = DriverManager.getConnection(

"jdbc:mysql://localhost:3306/ecommerce?createDatabaseIfNotExist=true", "root", "poiuy@098");

Statement st = connectionToDB.createStatement();

// 2-Execute a query to get data from the database

ResultSet resultData = st.executeQuery("select \* from product where id=" + pid + ";");

if (resultData.next()) {

// 3-Save the data in a new product object.

product.setId(resultData.getInt("id"));

product.setName(resultData.getString("name"));

product.setCompany(resultData.getString("company"));

}

} catch (SQLException e) {

System.out.println("sql problem:" + e);

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

// 4-Return the product to be used in the (Product Controller)

return product;

}

// Save new product:

// 1-Get the data from the product object (Created in the saveProductController)

// 2-Connect to the database

// 3-Execute a query to see if the project exist already in database or not

// 4-If product exist, return the product and notify the user(Product exist)

// 5-If not, then save the data in a new product object.

public Product saveProduct(int id, String sname, String scompany) {

Product product = new Product(id, sname, scompany);

// 1-Get the data from the product object (Created in the saveProductController)

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

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

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

try {

// 2-Connect to the database

Class.forName("com.mysql.cj.jdbc.Driver");

Connection connection = DriverManager.getConnection(

"jdbc:mysql://localhost:3306/ecommerce?createDatabaseIfNotExist=true", "root", "Abc@123");

Statement st = connection.createStatement();

// 3-Execute a query to see if the project exist already in database or not

ResultSet resultData = st.executeQuery("select \* from eproduct where id=" + product.getId() + ";");

// 4-If product exist, return the product and notify the user(Product exist)

if (resultData.next()) {

return product;

} else {

// 5-If not, then save the data in a new product object.

int result = st.executeUpdate("INSERT INTO eproduct(id,name,company) VALUES (" + product.getId() + ",'"

+ product.getName() + "','" + product.getPrice() + "');");

}

} catch (SQLException e) {

System.out.println("Sorry the is a problem with database" + e);

} catch (ClassNotFoundException ee) {

// TODO: handle exception

System.out.println("Sorry class not found" + ee);

}

return null;

}

}

**Product.java**

**package** com.file.web.model;

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

@Override

**public** String toString() {

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

}

**int** id;

String name;

**int** 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** **int** getPrice() {

**return** price;

}

**public** **void** setCompany(String company) {

**this**.price = price;

}

**public** Product(**int** id, String name, String company) {

**super**();

**this**.id = id;

**this**.name = name;

**this**.price = price;

}

**public** Product() {

**super**();

}

}

**Web.XML**

<!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>

<servlet>

<servlet-name>GetProductController</servlet-name>

<display-name>GetProductController</display-name>

<description></description>

<servlet-class>com.file.web.ProductController</servlet-class>

</servlet>

<servlet>

<servlet-name>SaveProductController</servlet-name>

<display-name>SaveProductController</display-name>

<description></description>

<servlet-class> com.file.web.SaveProductController</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>GetProductController</servlet-name>

<url-pattern>/getProduct</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>SaveProductController</servlet-name>

<url-pattern>/saveProduct</url-pattern>

</servlet-mapping>

</web-app>

**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>

<groupId>com.hcl</groupId>

<artifactId>WepProducts</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>war</packaging>

<name>WepProducts Maven Webapp</name>

<!-- FIXME change it to the project's website -->

<url>http://www.example.com</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.7</maven.compiler.source>

<maven.compiler.target>1.7</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.11</version>

<scope>test</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->

<dependency>

<groupId>mysql</groupId>

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

<version>8.0.21</version>

</dependency>

</dependencies>

<build>

<finalName>WepProducts</finalName>

<pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->

<plugins>

<plugin>

<artifactId>maven-clean-plugin</artifactId>

<version>3.1.0</version>

</plugin>

<!-- see http://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin\_bindings\_for\_war\_packaging -->

<plugin>

<artifactId>maven-resources-plugin</artifactId>

<version>3.0.2</version>

</plugin>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.0</version>

</plugin>

<plugin>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.22.1</version>

</plugin>

<plugin>

<artifactId>maven-war-plugin</artifactId>

<version>3.2.2</version>

</plugin>

<plugin>

<artifactId>maven-install-plugin</artifactId>

<version>2.5.2</version>

</plugin>

<plugin>

<artifactId>maven-deploy-plugin</artifactId>

<version>2.8.2</version>

</plugin>

</plugins>

</pluginManagement>

</build>

</project>