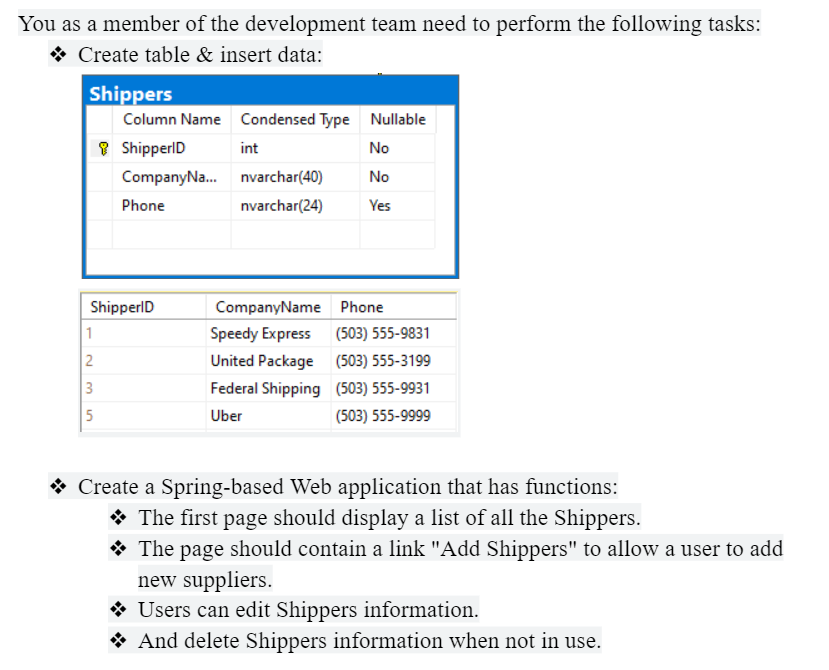
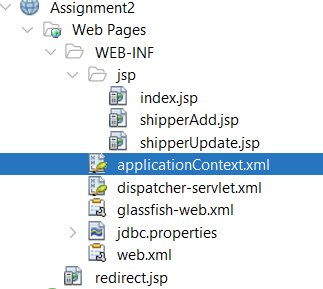
Assignment2



**Cấu trúc thư mục phải giống như hình**

**SQL:**USE master

GO

CREATE DATABASE A21131\_Assignment2;

GO

USE A21131\_Assignment2;

GO

CREATE TABLE Shippers (

ShipperID int PRIMARY KEY IDENTITY(1, 1),

CompanyName nvarchar(40) NOT NULL,

Phone nvarchar(24) NULL

);

GO

INSERT INTO Shippers(CompanyName, Phone) VALUES (N'Speedy Express', N'(503) 555-9831');

INSERT INTO Shippers(CompanyName, Phone) VALUES (N'United Package', N'(503) 555-3199');

INSERT INTO Shippers(CompanyName, Phone) VALUES (N'Federal Shipping', N'(503) 555-9931');

INSERT INTO Shippers(CompanyName, Phone) VALUES (N'Speedy Express', N'(503) 555-9831');

DELETE FROM Shippers

WHERE ShipperID = 4;

INSERT INTO Shippers(CompanyName, Phone) VALUES (N'Uber', N'(503) 555-9999');

**WEB-INF/jdbc.properties**

jdbc.driverClassName=com.microsoft.sqlserver.jdbc.SQLServerDriver

jdbc.url=jdbc:sqlserver://localhost:1433;databaseName=A21131\_Assignment2;encrypt=true;trustServerCertificate=true;

jdbc.username=sa

jdbc.password=sa

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

xmlns:aop="http://www.springframework.org/schema/aop"

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

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.3.xsd

http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.3.xsd

http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.3.xsd">

<bean id="propertyConfigurer"

class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"

p:location="/WEB-INF/jdbc.properties" />

<bean id="dataSource"

class="org.springframework.jdbc.datasource.DriverManagerDataSource"

p:driverClassName="${jdbc.driverClassName}"

p:url="${jdbc.url}"

p:username="${jdbc.username}"

p:password="${jdbc.password}" />

<!-- ADD PERSISTENCE SUPPORT HERE (jpa, hibernate, etc) -->

<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">

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

</bean>

<bean id="rdao" class="com.models.ShipperImplement">

<property name="jdbcTemplate" ref="jdbcTemplate"></property>

</bean>

</beans>

**jsp/index.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

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

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>List Of Shipper</title>

</head>

<body>

<h1>MANAGE SHIPPER</h1>

<a href="addForm.htm">Add Shipper</a>

<form action="findShipper.htm">

<table>

<tbody>

<tr>

<td><label for="keyword">Search:</label></td>

<td><input type="text" id="keyword" name="keyword" value=""></td>

<td><button type="submit">Search</button></td>

</tr>

</tbody>

</table>

</form>

<table>

<thead>

<tr>

<th>Shipper ID</th>

<th>Company Name</th>

<th>Phone</th>

<th>Edit</th>

<th>Delete</th>

</tr>

</thead>

<tbody>

<c:forEach var="shipper" items="${listShipper}" varStatus="status">

<tr>

<td><c:out value="${shipper.shipperID}"/></td>

<td><c:out value="${shipper.companyName}"/></td>

<td><c:out value="${shipper.phone}"/></td>

<td><a href="showUpdateForm.htm?id=<c:out

value="${shipper.shipperID}"/>">Edit</a></td>

<td><a href="deleteShipper.htm?id=<c:out

value="${shipper.shipperID}"/>">Delete</a></td>

</tr>

</c:forEach>

</tbody>

</table>

</body>

</html>

**jsp/shipperAdd.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Add New Shipper</title>

</head>

<body>

<form:form action="addShipper.htm" modelAttribute="shipper" class="needs-validation">

<h1>Shipper Information</h1>

<table>

<tbody>

<tr>

<td>

<label for="companyName" class="form-label">Company Name:</label>

</td>

<td><form:input path="companyName" value="${companyName}"/> <br>

<form:errors path="companyName" cssStyle="color:red"/>

</td>

</tr>

<tr>

<td>

<label for="phone" class="form-label">Phone:</label>

</td>

<td><form:input path="phone" value="${phone}"/> <br>

<form:errors path="phone" cssStyle="color:red"/>

</td>

</tr>

<tr>

<td><input type="submit" value="Add Shipper"/></td>

<td><button type="reset">Reset</button></td>

</tr>

</tbody>

</table>

</form:form>

</body>

</html>

**jsp/shipperUpdate.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

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

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Update Information Shipper</title>

</head>

<body>

<form:form action="updateShipper.htm" modelAttribute="shipper">

<h1>Shipper Information</h1>

<table>

<tbody>

<tr>

<td><label for="shipperID">Shipper ID:</label></td>

<td><form:input path="shipperID" value="${shipperID}" readonly="true"/></td>

</tr>

<tr>

<td><label for="companyName">Company Name:</label></td>

<td><form:input path="companyName" value="${companyName}"/> <br>

<form:errors path="companyName" cssStyle="color:red"/></td>

</tr>

<tr>

<td><label for="phone">Phone:</label></td>

<td><form:input path="phone" value="${phone}"/> <br>

<form:errors path="phone" cssStyle="color:red"/></td>

</tr>

<tr>

<td><input type="submit" value="Update"/></td>

<td><button type="reset">Reset</button></td>

</tr>

</tbody>

</table>

</form:form>

</body>

</html>

**com.controllers**

**ShipperController.java**

package com.controllers;

import com.models.Shipper;

import com.models.ShipperDAO;

import java.util.List;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

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

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

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

@Controller

@RequestMapping("/")

public class ShipperController {

@Autowired

private ShipperDAO shipperDAO;

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

public String showListShipper(ModelMap model) {

model.addAttribute("listShipper", shipperDAO.findAll());

return "index";

}

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

public String findShipper(String keyword, ModelMap model) {

List<Shipper> listShipper = shipperDAO.find(keyword);

model.addAttribute("listShipper", listShipper);

return "index";

}

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

public String showAddShipperForm() {

return "shipperAdd";

}

@ModelAttribute("shipper")

public Shipper setShipperForm() {

return new Shipper();

}

@RequestMapping(value = "addShipper", method = RequestMethod.POST)

public String addShipper(@ModelAttribute("shipper") Shipper shipper, ModelMap model) {

shipperDAO.add(shipper);

return showListShipper(model);

}

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

public String showUpdateShipperForm(int id, ModelMap model) {

Shipper shipper = shipperDAO.get(id);

model.addAttribute("shipperID", shipper.getShipperID());

model.addAttribute("companyName", shipper.getCompanyName());

model.addAttribute("phone", shipper.getPhone());

return "shipperUpdate";

}

@RequestMapping(value = "updateShipper", method = RequestMethod.POST)

public String updateShipper(@ModelAttribute("shipper") Shipper shipper, ModelMap model) {

shipperDAO.update(shipper);

return showListShipper(model);

}

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

public String deleteShipper(int id, ModelMap model) {

shipperDAO.delete(id);

return showListShipper(model);

}

}

**com.models**

**Shipper.java**

package com.models;

import javax.validation.constraints.NotNull;

public class Shipper {

private String shipperID;

@NotNull(message = "Company Name can't be empty!!!")

private String companyName;

private String phone;

/\*Getter and setter - constructor\*/

}

**ShipperDAO.java (interface)**

package com.models;

import java.util.List;

public interface ShipperDAO {

public List<Shipper> findAll();

public void add(Shipper shipper);

public void update(Shipper shipper);

public Shipper get(int id);

public void delete(int id);

public List<Shipper> find(String keyword);

}

**ShipperImplement.java**

package com.models;

import java.util.ArrayList;

import java.util.List;

import java.util.Map;

import javax.sql.DataSource;

import org.springframework.jdbc.core.BeanPropertyRowMapper;

import org.springframework.jdbc.core.JdbcTemplate;

public class ShipperImplement implements ShipperDAO {

private JdbcTemplate jdbcTemplate;

public ShipperImplement() {

}

public ShipperImplement(DataSource ds) {

jdbcTemplate = new JdbcTemplate(ds);

}

@Override

public List<Shipper> findAll() {

String sql = "SELECT TOP 100 \* FROM Shippers";

List<Shipper> shipper = new ArrayList<>();

List<Map<String, Object>> rows = jdbcTemplate.queryForList(sql);

for (Map row : rows) {

Shipper shipperObject = new Shipper();

shipperObject.setShipperID(row.get("ShipperID").toString());

shipperObject.setCompanyName(row.get("CompanyName").toString());

shipperObject.setPhone(row.get("Phone").toString());

shipper.add(shipperObject);

}

return shipper;

}

@Override

public void add(Shipper shipper) {

String sql = "INSERT INTO Shippers VALUES (?,?)";

jdbcTemplate.update(sql, shipper.getCompanyName(), shipper.getPhone());

}

@Override

public void update(Shipper shipper) {

String sql = "UPDATE Shippers SET CompanyName = ?, Phone = ? WHERE ShipperID = ?";

jdbcTemplate.update(sql, shipper.getCompanyName(), shipper.getPhone(), shipper.getShipperID());

}

@Override

public Shipper get(int id) {

String sql = "SELECT TOP 100 \* FROM Shippers WHERE ShipperID = ?";

return (Shipper) jdbcTemplate.queryForObject(sql, new Object[]{id}, new BeanPropertyRowMapper(Shipper.class));

}

@Override

public void delete(int id) {

String sql = "DELETE FROM Shippers WHERE ShipperID = ?";

jdbcTemplate.update(sql, id);

}

@Override

public List<Shipper> find(String keyword) {

StringBuilder sql = new StringBuilder("SELECT TOP 100 \* FROM Shippers ");

// Check if keyword is empty or null

if (keyword == null || keyword.isEmpty()) {

return jdbcTemplate.query(sql.toString(), BeanPropertyRowMapper.newInstance(Shipper.class));

}

sql.append("WHERE (CompanyName LIKE ? OR Phone LIKE ?)");

keyword = "%" + keyword + "%";

return jdbcTemplate.query(sql.toString(), new Object[]{keyword, keyword},

BeanPropertyRowMapper.newInstance(Shipper.class));

}

public JdbcTemplate getJdbcTemplate() {

return jdbcTemplate;

}

public void setJdbcTemplate(JdbcTemplate jdbcTemplate) {

this.jdbcTemplate = jdbcTemplate;

}

}