

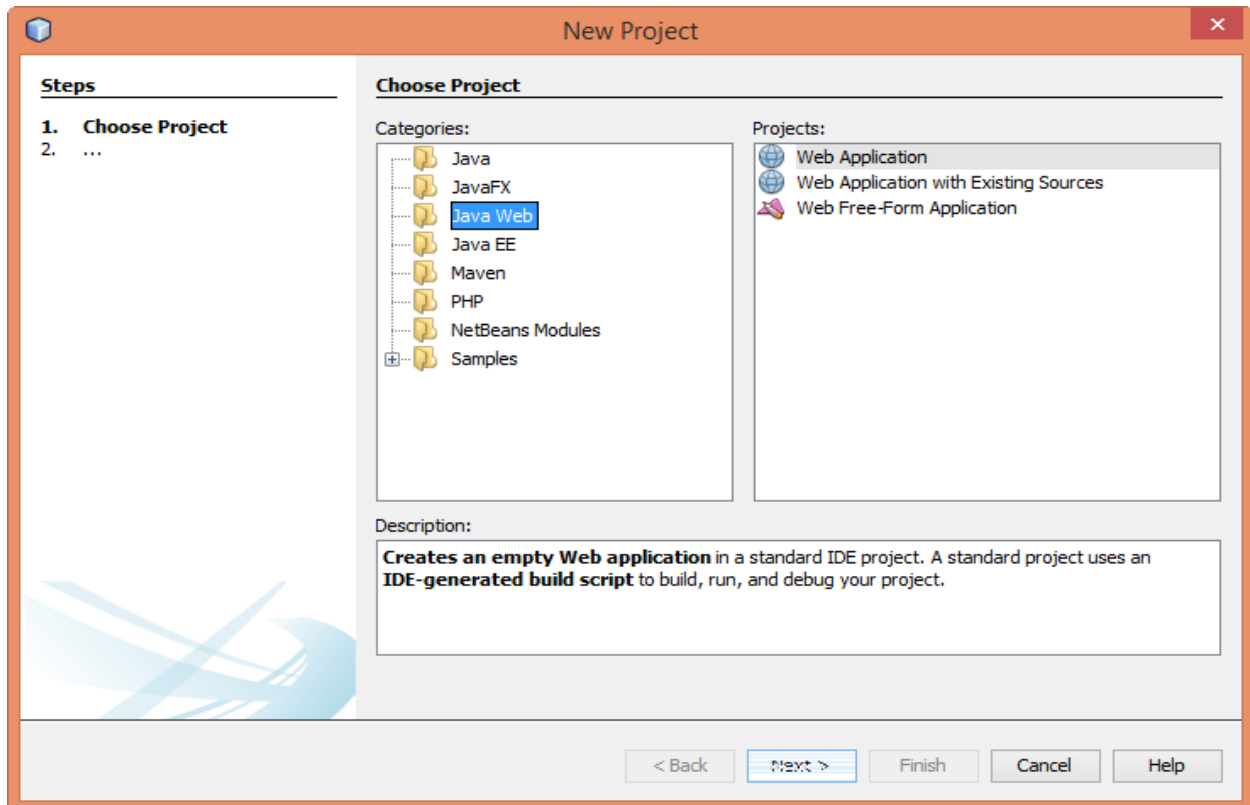
Hướng dẫn các bước phát triển CRUD 3-layer với JSP Servlet

Giáo viên: Nguyễn Hùng Cường

Phần I: Hướng dẫn các bước liệt kê dữ liệu

Bước 1: Tạo web application

Mở NetBeans IDE, tạo mới một ứng dụng web.



Đặt tên và chọn đường dẫn lưu ứng dụng.

The screenshot shows the 'New Web Application' wizard in an IDE. The title bar is orange with a close button. The left sidebar, titled 'Steps', lists four steps: 1. Choose Project, 2. Name and Location (highlighted), 3. Server and Settings, and 4. Frameworks. The main area is titled 'Name and Location' and contains the following fields: 'Project Name' with the value 'DemoCRUDJSPServlet', 'Project Location' with the value 'G:\DL giang day\Tai lieu giang day\' and a 'Browse...' button, and 'Project Folder' with the value 'G:\DL giang day\Tai lieu giang day\DemoCRUDJSPServlet'. Below these is a checked checkbox 'Use Dedicated Folder for Storing Libraries' and a 'Libraries Folder' field with the value '.\\lib' and a 'Browse...' button. A note states: 'Different users and projects can share the same compilation libraries (see Help for details)'. At the bottom are buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

Steps

1. Choose Project
- 2. Name and Location**
3. Server and Settings
4. Frameworks

Name and Location

Project Name: DemoCRUDJSPServlet

Project Location: G:\DL giang day\Tai lieu giang day\ Browse...

Project Folder: G:\DL giang day\Tai lieu giang day\DemoCRUDJSPServlet

☒ Use Dedicated Folder for Storing Libraries

Libraries Folder: .\\lib Browse...

Different users and projects can share the same compilation libraries (see Help for details).

< Back Next > Finish Cancel Help

Sau đó chọn server và phiên bản JavaEE cho ứng dụng.

The screenshot shows the 'New Web Application' wizard in an IDE, now at Step 3: 'Server and Settings'. The left sidebar highlights step 3. The main area is titled 'Server and Settings' and contains the following fields: 'Add to Enterprise Application' with a dropdown menu set to '<None>', 'Server' with a dropdown menu set to 'Apache Tomcat 7.0.27.0' and an 'Add...' button, 'Java EE Version' with a dropdown menu set to 'Java EE 5', a checked checkbox 'Set Source Level to 1.5' with a recommendation note 'Recommendation: Source Level 1.5 should be used in Java EE 5 projects.', and 'Context Path' with the value '/DemoCRUDJSPServlet'. At the bottom are buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

Steps

1. Choose Project
2. Name and Location
- 3. Server and Settings**
4. Frameworks

Server and Settings

Add to Enterprise Application: <None>

Server: Apache Tomcat 7.0.27.0 Add...

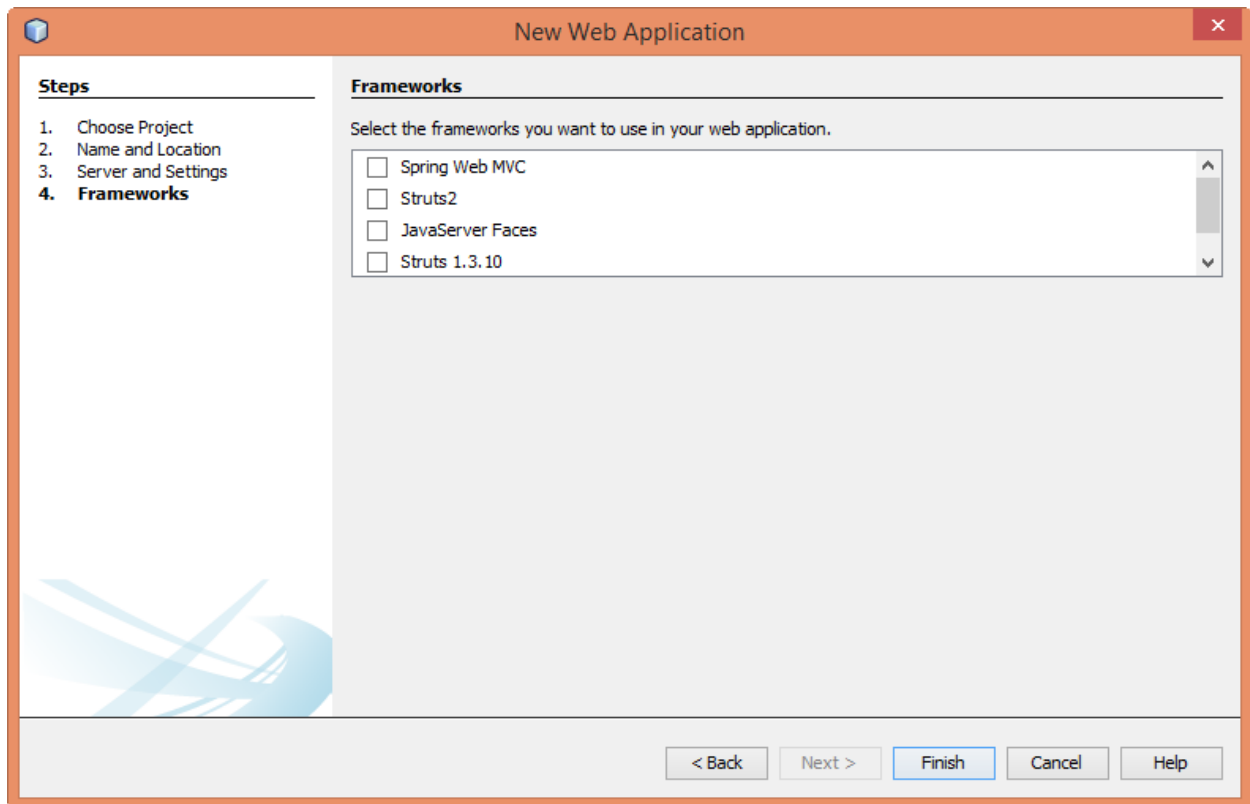
Java EE Version: Java EE 5

☒ Set Source Level to 1.5
Recommendation: Source Level 1.5 should be used in Java EE 5 projects.

Context Path: /DemoCRUDJSPServlet

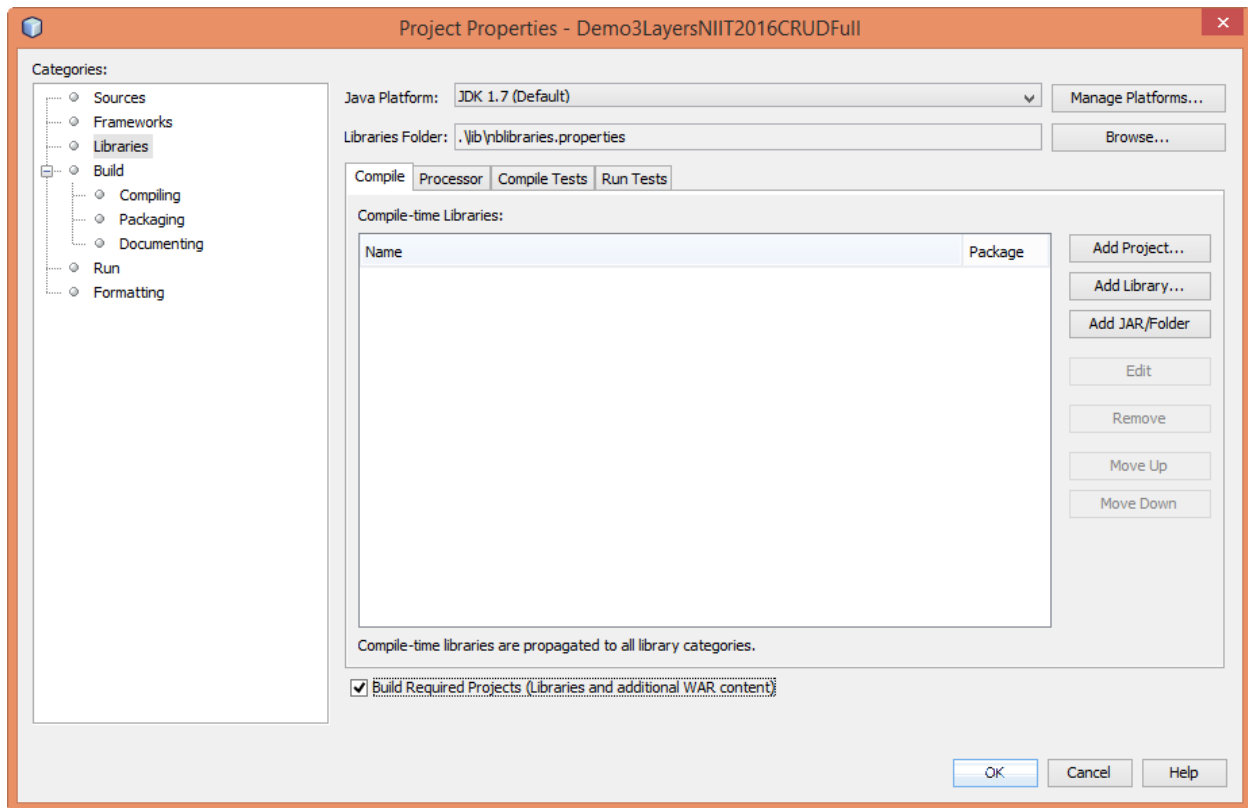
< Back Next > Finish Cancel Help

Sau đó ta có thể chọn Framework cho ứng dụng. Trong tài liệu này, ta không sử dụng Framework nào bên ngoài nên ta không chọn mục nào.

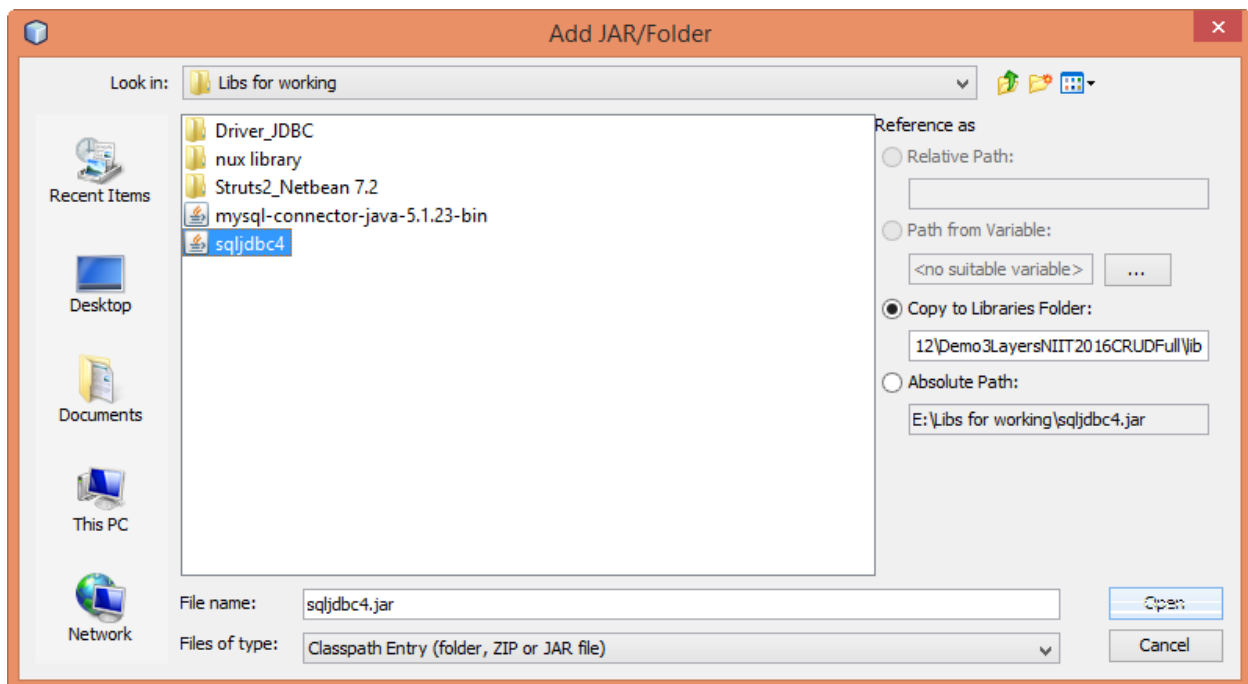


Bước 2: Add JDBC Library

Để có thể kết nối với database, ta cần add thư viện JDBC vào Project. Để add thư viện, click phải chuột vào Project, chọn mục Libraries, chọn Add JAR/Folder để thêm thư viện mới.



Tiếp theo hãy chọn đường dẫn thư mục chứa thư viện, chọn thư viện để add vào Project.



Bước 3: Tạo database

Mở một RDBMS (ở đây là SQL Server). Tạo mới database (ở đây tên là quanlyhaisan), sau đó tạo table (trong tài liệu này ta đặt tên là haisan) và chèn một số bản ghi vào trong bảng.

```
create database quanlyhaisan
go
use quanlyhaisan
go
```

```
create table haisan (
id int primary key identity(1, 1),
name nvarchar(50),
price int,
address nvarchar(50)
)
```

```
select * from haisan
```

Chèn dữ liệu vào bảng, sau đó có thể tạo một số StoredProcedure để cho phép thao tác với dữ liệu trong bảng.

```
insert into haisan (name, price, address) values ('Cua bien', 400, 'Bien Dong')
insert into haisan (name, price, address) values ('Nhim bien - Cau gai', 300,
'Bien Caribe')
insert into haisan (name, price, address) values ('Muc ong', 250, 'Bien Thai Binh Duong')
insert into haisan (name, price, address) values ('Bach tuoc', 150, 'Bien Dai Tay Duong')
insert into haisan (name, price, address) values ('Ca chim', 400, 'Bien Den')

create proc InsertSeaFood
@name nvarchar(50),
@price int,
@address nvarchar(50)
AS
    insert into haisan (name, price, address) values(@name, @price, @address)
```

```

create proc UpdateSeaFood
@id int,
@name nvarchar(50),
@price int,
@address nvarchar(50)
AS
    update haisan set name = @name, price = @price, address = @address where id = @id

```

```

create proc DeleteSeaFood
@id int
AS
    delete haisan where id = @id

```

```

create proc ListSeaFood
AS
    select * from haisan

```

```

create proc GetSeaFoodById
@id int
AS
    select * from haisan where id = @id

```

Results		Messages		
	id	name	price	address
1	3	Muc ong	5000000	Bien Thai Binh Duong
2	4	Bach tuoc	5000000	Bien Dai Tay Duong
3	5	Ca chim	5000000	Bien Den
4	6	Cua hoang de	6000000	Bien Baltic
5	7	Ca thu	200000	Bien Dong
6	8	Ca thu	-200000	Bien Dong

Bước 4: Viết mã tầng data access layer

a) Viết mã kết nối với database

- Tạo class có tên DatabaseUtils (nằm trong package com.qlhs.util), trong class này ta định nghĩa một method cho phép kết nối đến database.

```

package com.qlhs.util;
import java.sql.Connection;
import java.sql.DriverManager;
/**
 * @author Admin
 */
public class DatabaseUtils {
    public static Connection getConnection() {
        Connection con = null;
        try {
            //load driver
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
            //mo ket noi den CSDL
            con = DriverManager.
                getConnection("jdbc:sqlserver://localhost;databaseName=quanlyhaisan", "sa", "1234$");
        } catch (Exception e) {
            e.printStackTrace();
        }
        return con;
    }
}

```

b) Tạo DTO

Tiếp theo, tạo package com.qlhs.dto, trong package này tạo một class có tên là SeaFood, class này sẽ tương ứng với bảng SeaFood trong database. Trong class này ta tạo các thuộc tính là private, rồi tạo các cặp method setter/getter tương ứng với từng thuộc tính này để thao tác.

```

package com.qlhs.dto;
/**
 * Data Transfer Object - JavaBean
 * @author Admin
 */
public class SeaFood {
    private int id;
    private int price;
    private String name;
    private String address;
    public SeaFood() {
    }
    public SeaFood(int price, String name, String address) {
        this.price = price;
        this.name = name;
        this.address = address;
    }
}

```

```

public SeaFood(int id, int price, String name, String address) {
    this.id = id;
    this.price = price;
    this.name = name;
    this.address = address;
}
public int getId() {
    return id;
}
public void setId(int id) {
    this.id = id;
}
public int getPrice() {
    return price;
}
public void setPrice(int price) {
    this.price = price;
}

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
}

```

c) Tạo Data Access Object

Tạo mới package com.qlhs.dao, trong đó tạo một class có tên là HaiSanDAO, class

này cung cấp các method cho phép kết nối CSDL và tương tác với bảng seafood (các thao tác insert – update – delete – list).

```
package com.qlhs.dao;

import com.qlhs.dto.SeaFood;
import com.qlhs.util.DatabaseUtils;
import java.sql.*;
import java.util.ArrayList;

/**
 * Data Access Object
 * @author Admin
 */
public class HaiSanDAO {
    public static ArrayList<SeaFood> getListSeaFood() {
        ArrayList<SeaFood> listSeaFoods = null;

        try {
            //mo ket noi den CSDL
            Connection con = DatabaseUtils.getConnection();
            //goi thu tuc trong CSDL
            CallableStatement pre = con.prepareCall("{call ListSeaFood}");

            //truy van du lieu trong CSDL
            ResultSet rs = pre.executeQuery();

            //kiem tra xem rs co null hay khong
            if(rs != null) {
                listSeaFoods = new ArrayList<SeaFood>();
            }
        }
    }
}
```

```

        //vong lap nay dung de duyet qua tung ban ghi trong ResultSet
        //sau do chuyen tung ban ghi nay thanh tung doi tuong SeaFood
        //roi add doi tuong SeaFood vao ArrayList
        while(rs.next()) {
            //khoi tao doi tuong SeaFood, sau do lay du lieu trong cac cot
            //de thiet lap cho SeaFood
            SeaFood tmp = new SeaFood();
            tmp.setId(rs.getInt("id"));
            tmp.setName(rs.getString("name"));
            tmp.setPrice(rs.getInt("price"));
            tmp.setAddress(rs.getString("address"));

            //them doi tuong SeaFood vao ArrayList
            listSeaFoods.add(tmp);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return listSeaFoods;
}
}

```

Bước 5: Viết mã servlet

Tiếp theo tạo package có tên là com.qlhs.controller, sau đó tạo servlet có tên là ListSeaFoodServlet. Trong servlet này, ta viết mã để kết nối với CSDL, rồi truyền dữ liệu sang cho trang hiển thị danh sách, rồi forward đến trang này.

```

protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
        //goi DAO de lay du lieu tu CSDL
        ArrayList<SeaFood> listSeaFood = HaiSanDAO.getListSeaFood();

        //truyen attribute sang trang ke tiep
        request.setAttribute("danhsachhaisan", listSeaFood);

        //forward den trang JSP co ten la ListSeaFood.jsp,
        //dong thoi truyen gia tri sang trang do
        RequestDispatcher dispatcher = getServletContext().
            getRequestDispatcher("/ListSeaFood.jsp");
        dispatcher.forward(request, response);
    }
}

```

Bước 6: Tạo View

a) Tạo trang chủ

Tạo mới trang JSP cho phép hiển thị hyperlink cung cấp liên kết để cho phép gọi servlet rồi chuyển đến trang danh sách.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>Hello World!</h1>

        <h3><a href="ListSeaFoodServlet">View list seafood</a></h3>

    </body>
</html>
```

b) Tạo trang danh sách

Sau đó ta tạo trang danh sách, trang này sẽ lấy dữ liệu được truyền từ servlet sang, rồi viết mã java để duyệt qua mảng (ở đây là ArrayList các đối tượng SeaFood) và hiển thị dữ liệu lên trang web.

```
<%@page contentType="text/html" pageEncoding="UTF-8"
import="java.util.ArrayList, com.qlhs.dto.SeaFood"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>List seafood</h1>
```

```
<%
    //lay attribute chua danh sach du lieu duoc truyen sang tu Servlet
    ArrayList<SeaFood> listSeaFood = (ArrayList<SeaFood>)request.
        getAttribute("danhsachhaisan");
%>
```

```

<form action="" method="post">
<table width="500" border="1">
    <tr>
        <th>ID</th>
        <th>Name</th>
        <th>Price</th>
        <th>Address</th>
        <th>Update</th>
        <th>Delete</th>
        <th></th>
    </tr>

```

```

<%
    for(int i = 0; i < listSeaFood.size(); i++) {
        SeaFood f = listSeaFood.get(i);
    %>

```

```

<tr>
    <td><%= f.getId() %></td>
    <td><%= f.getName() %></td>
    <td><%= f.getPrice() %></td>
    <td><%= f.getAddress() %></td>
    <td><a href="EditSeaFoodServlet?id=<%= f.getId() %>">Edit</a></td>
    <td><a href="DeleteSeaFoodServlet?id=<%= f.getId() %>"
        onclick="return confirm('Ban co thuc su muon xoa ?');">Delete</a></td>
    <td><input type="checkbox" name="checkid" value="<%= f.getId() %>" /></td>
</tr>

```

```

<%
    }
%>
<tr>
    <td colspan="7" align="right"><input type="submit" value="Delete multi"/></td>
</tr>

```

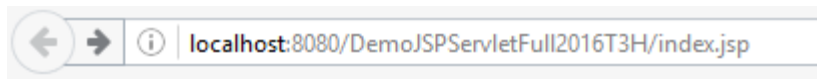
```
</table>
</form>
<BR><BR>

<a href="FormAddSeaFood.jsp">Add seafood</a>

</body>
</html>
```

Bước 7: Thực thi ứng dụng và xem kết quả

Sau khi đã viết mã xong, hãy click phải chuột lên ứng dụng, chọn Build để biên dịch. Sau đó thực thi ứng dụng và xem kết quả.



Hello World!

[View list seafood](#)

List seafood

ID	Name	Price	Address	Update	Delete	
3	Muc ong	5000000	Bien Thai Binh Duong	Edit	Delete	<input type="checkbox"/>
4	Bach tuoc	5000000	Bien Dai Tay Duong	Edit	Delete	<input type="checkbox"/>
5	Ca chim	5000000	Bien Den	Edit	Delete	<input type="checkbox"/>
6	Cua hoang de	6000000	Bien Baltic	Edit	Delete	<input type="checkbox"/>
7	Ca thu	200000	Bien Dong	Edit	Delete	<input type="checkbox"/>
8	Ca thu	-200000	Bien Dong	Edit	Delete	<input type="checkbox"/>
					Delete multi	

[Add seafood](#)

Phần II – Hướng dẫn các bước thêm mới dữ liệu

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Thêm loại tin</h1>
    <form action="AddCategoryServlet" method="get">
      <table border="0">
        <tr>
          <td>Nội dung:</td>
          <td><input type="text" name="txtcontent" /> </td>
        </tr>
        <tr>
          <td><td valign="top">Mô tả:</td>
          <td><textarea cols="20" rows="10" name="txtdescription" /></td>
        </tr>
      </table>
    </form>
  </body>
</html>

```

```
        <tr>
            <td colspan="2" align="center"><input type="submit" value="Thêm mới" /></td>
        </tr>
    </table>
</form>
</body>
</html>
```

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
        //lay gia tri tham so gui len tu trang JSP
        String content = request.getParameter("txtcontent");
        String description = request.getParameter("txtdescription");

        //goi dao de insert du lieu vao database
        NewsCategoryDAO.insert(content, description);

        //chuyen tiep den view tiep theo
        RequestDispatcher dispatch = getServletContext().
            getRequestDispatcher("/ListCategoryServlet");
        dispatch.forward(request, response);
    } finally {
        out.close();
    }
}
```

```
//method nay dung de insert ban ghi vao trong CSDL
public static int insert(String content, String description) {
    int records = 0;
    try {
        //mo ket noi den CSDL
        Connection con = DatabaseUtils.getConnection();
        String sql = "insert into newscategory (content, description) values(?, ?)";
        //tao doi tuong CallableStatement de goi thu tuc
        PreparedStatement pre = con.prepareStatement(sql);
        //truyen gia tri vao thu tuc
        pre.setString(1, content);
        pre.setString(2, description);

        //goi thu tuc trong CSDL
        records = pre.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return records;
}
```

Danh sách loại tin

Mã	Nội dung	Mô tả	Sửa	Xóa
7	Tin thể thao	Barca thắng ManCity 3-0	Sửa	Xóa
8	Tin giao thông	Khai trương đường cao tốc HN - LC	Sửa	Xóa
9	Tin kinh tế	Thành lập doanh nghiệp startup	Sửa	Xóa
10	Tin giáo dục	Phát minh ra phương pháp đào tạo mới	Sửa	Xóa

[Thêm mới loại tin](#)

Thêm loại tin

Nội dung:

Mô tả:

Doan khao co da phat
hien di tích Hoang
thanh Thang Long

Thêm mới

Danh sách loại tin

Mã	Nội dung	Mô tả	Sửa	Xóa
7	Tin thể thao	Barca thắng ManCity 3-0	Sửa	Xóa
8	Tin giao thông	Khai trương đường cao tốc HN - LC	Sửa	Xóa
9	Tin kinh tế	Thành lập doanh nghiệp startup	Sửa	Xóa
10	Tin giáo dục	Phát minh ra phương pháp đào tạo mới	Sửa	Xóa
17	Tin lịch sử	Doan khao co da phat hien di tích Hoang thanh Thang Long	Sửa	Xóa

[Thêm mới loại tin](#)

Phần III – Hướng dẫn các bước cập nhật dữ liệu

Bước 1: Viết mã tầng Data Access Layer

Mở DAO, viết mã để cung cấp 2 method phục vụ cho việc cập nhật dữ liệu.

Method `getSeaFoodById(int id)`: Truy vấn dữ liệu trong CSDL, rồi trả về đối tượng DTO tương ứng với bản ghi trong CSDL theo id.

Method `update()`: Cập nhật dữ liệu trong database (ở đây ta gọi thủ tục)

Viết mã cho method lấy thông tin bản ghi theo id

```
//method này dùng để lấy một bản ghi trong CSDL theo id
public static SeaFood getSeaFoodById(int id) {
    SeaFood food = new SeaFood();
    try {
        //mở kết nối đến CSDL
        Connection con = DatabaseUtils.getConnection();
        //gọi thủ tục
        CallableStatement pre = con.prepareCall("{call GetSeaFoodById (?)}");
        pre.setInt(1, id);

        //truy vấn dữ liệu trong CSDL
        ResultSet rs = pre.executeQuery();

        if(rs.next()) {
            food.setId(rs.getInt("id"));
            food.setName(rs.getString("name"));
            food.setPrice(rs.getInt("price"));
            food.setAddress(rs.getString("address"));
        }

    } catch (Exception e) {
        e.printStackTrace();
    }
    return food;
}
```

Tiếp theo ta viết mã để cung cấp method cho phép cập nhật dữ liệu trong CSDL.

```

//method nay cho phép gọi StoredProcedure để thực hiện
//cập nhật dữ liệu trong bảng
public static int update(SeaFood food) {
    int records = 0;
    int index = 0;
    try {
        Connection con = DatabaseUtils.getConnection();
        //gọi StoredProcedure để cập nhật dữ liệu
        CallableStatement pre = con.prepareCall("{call UpdateSeaFood (?, ?, ?, ?)}");
        //truyền giá trị cho thủ tục
        pre.setInt(++index, food.getId());
        pre.setString(++index, food.getName());
        pre.setInt(++index, food.getPrice());
        pre.setString(++index, food.getAddress());
        //thực thi việc cập nhật dữ liệu
        records = pre.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return records;
}

```

Bước 2: Viết mã cho tầng giao diện

a) Cập nhật file danh sách

Mở file hiển thị danh sách ListSeaFood.jsp, trong vòng lặp hãy cung cấp một hyperlink trên mỗi dòng trong bảng, mỗi khi click vào link này thì sẽ chuyển đến EditSeaFoodServlet để hiển thị form cập nhật, đồng thời truyền id của bản ghi đến cho servlet.

```

<td><a href="EditSeaFoodServlet?id=<%= f.getId() %>">Edit</a></td>

```

b) Xây dựng form cập nhật

Sau đó ta tạo ra trang JSP chứa form cập nhật. Form này sẽ hiển thị thông tin chi tiết về bản ghi trong CSDL theo id (để người dùng nhập dữ liệu mới và cập nhật).

```

<%@page contentType="text/html" pageEncoding="UTF-8" import="com.qlhs.dto.SeaFood"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Update seafood</h1>

    <%
      //lay gia tri attribute duoc gui tu Servlet sang
      SeaFood food = (SeaFood)request.getAttribute("haisan");
    %>

    <form id="form1" name="form1" method="post" action="UpdateSeaFoodServlet">
    <table width="500" border="1">
      <tr>
        <td colspan="2" align="center"><strong>Update seafood</strong></td>
      </tr>
      <tr>
        <td width="112">Name:</td>
        <td width="372">
          <input type="text" name="txtname" id="txtname" value="<%= food.getName() %>">
          <input type="hidden" name="txtid" value="<%= food.getId() %>">
        </td>
      </tr>
      <tr>
        <td>Price</td>
        <td><input type="text" name="txtprice" id="txtprice"
          value="<%= food.getPrice() %>"></td>
      </tr>

      <tr>
        <td colspan="2" align="center">
          <input name="button" type="submit" id="button" form="form1" value="Update">
          <input name="btnReset" type="button" id="btnReset" form="form1" value="Nhập lại">
        </td>
      </tr>
    </table>
    </form>
  </body>
</html>

```

Bước 3: Viết mã servlet

Sau đó ta cần tạo ra các servlet (controller) để cung cấp các mã xử lý cho việc cập nhật thông tin. Các servlet này nằm trong package com.qlhs.controller.

a) Tạo EditSeaFoodServlet

Servlet này làm nhiệm vụ gọi DAO để lấy ra thông tin về bản ghi tương ứng trong CSDL theo id, sau đó truyền đối tượng này và chuyển tiếp đến form cập nhật.

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
        //lay id duoc gui tu trang list sang
        int id = Integer.parseInt(request.getParameter("id"));

        //goi model de lay ban ghi tu CSDL theo id
        SeaFood food = HaiSanDAO.getSeaFoodByID(id);

        //truyen gia tri sang trang ke tiep
        request.setAttribute("haisan", food);

        //chuyen den trang ke tiep
        RequestDispatcher dispatch = getServletContext().
            getRequestDispatcher("/FormEditSeaFood.jsp");
        dispatch.forward(request, response);
    }
}
```

b) Tạo UpdateSeaFoodServlet

Servlet này sẽ chịu trách nhiệm lấy dữ liệu được gửi từ form cập nhật, sau đó truyền dữ liệu cho DAO để cập nhật dữ liệu, sau đó forward quay lại trang danh sách.

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
        //lay gia tri tu form gui len theo request
        String name = request.getParameter("txtname");
        int id = Integer.parseInt(request.getParameter("txtid"));
        int price = Integer.parseInt(request.getParameter("txtprice"));
        String address = request.getParameter("txtaddress");

        //khởi tạo một đối tượng của lớp SeaFood
        SeaFood food = new SeaFood(id, price, name, address);

        //gọi dao để update trong CSDL
        HaiSanDAO.update(food);
    }
}
```

```

        //chuyen den trang ke tiep
        RequestDispatcher dispatch = getServletContext().
            getRequestDispatcher("/ListSeaFoodServlet");
        dispatch.forward(request, response);
    }
}

```

Bước 4: Thực thi ứng dụng và xem kết quả

Sau khi đã viết mã xong, hãy chọn Build để biên dịch ứng dụng. Tiếp theo hãy thực thi ứng dụng để xem kết quả của việc cập nhật dữ liệu.

List seafood

ID	Name	Price	Address	Update	Delete	
3	Muc ong	5000000	Bien Thai Binh Duong	Edit	Delete	<input type="checkbox"/>
4	Bach tuoc	5000000	Bien Dai Tay Duong	Edit	Delete	<input type="checkbox"/>
5	Ca chim	5000000	Bien Den	Edit	Delete	<input type="checkbox"/>
6	Cua hoang de	6000000	Bien Baltic	Edit	Delete	<input type="checkbox"/>
7	Ca thu	200000	Bien Dong	Edit	Delete	<input type="checkbox"/>
8	Ca thu	-200000	Bien Dong	Edit	Delete	<input type="checkbox"/>
						<input type="button" value="Delete multi"/>

[Add seafood](#)

Từ trang danh sách, khi người dùng click vào hyperlink, sẽ chuyển đến trang edit. Người dùng nhập dữ liệu mới lên form, rồi click submit form.

Update seafood

Update seafood	
Name:	<input type="text" value="Ca chim bien"/>
Price	<input type="text" value="5500000"/>
Address:	<input type="text" value="Bien Den"/>
<input type="button" value="Update"/> <input type="button" value="Nhap lai"/>	

Sau khi dữ liệu đã được cập nhật, ta có thể thấy dữ liệu mới đã được hiển thị trong danh sách như hình bên dưới.

List seafood

ID	Name	Price	Address	Update	Delete	
3	Muc ong	5000000	Bien Thai Binh Duong	Edit	Delete	<input type="checkbox"/>
4	Bach tuoc	5000000	Bien Dai Tay Duong	Edit	Delete	<input type="checkbox"/>
5	Ca chim bien	5500000	Bien Den	Edit	Delete	<input type="checkbox"/>
6	Cua hoang de	6000000	Bien Baltic	Edit	Delete	<input type="checkbox"/>
7	Ca thu	200000	Bien Dong	Edit	Delete	<input type="checkbox"/>
8	Ca thu	-200000	Bien Dong	Edit	Delete	<input type="checkbox"/>
						<input type="button" value="Delete multi"/>

[Add seafood](#)

Phần IV: Hướng dẫn các bước xóa dữ liệu

Bước 1: Chỉnh sửa view

Mở trang danh sách, trong vòng lặp ta hãy tạo ra một hyperlink, mỗi lần lặp ta sẽ tạo ra một hyperlink để thực hiện việc xóa. Mỗi lần người dùng click vào hyperlink này thì sẽ chuyển đến file DeleteCategoryServlet, đồng thời truyền id của bản ghi đến servlet tương ứng.

```
<td><a class="link" href="DeleteCategoryServlet?id=<%= tmp.getId() %>"
onclick="return confirm('Bạn có thực sự muốn xóa ?');">Xóa</a></td>
```

Bước 2: Viết mã cho model

Vào phần source package, tạo mới một class DAO có tên là NewsCategoryDAO (hoặc một tên bất kỳ do mình tự định nghĩa). Trong class này hãy cung cấp một method cho phép thực hiện xóa dữ liệu trong bảng theo id như bên dưới.

```
//method này cho phép xóa bản ghi theo id
public static int delete(int id) {
    int records = 0;
    try {
        Connection con = DatabaseUtils.getConnection();
        String sql = "delete newscategory where id = ?";

        //tạo đối tượng CallableStatement để gọi thủ tục
        PreparedStatement pre = con.prepareStatement(sql);

        //truyền giá trị vào thủ tục
        pre.setInt(1, id);

        //gọi thủ tục trong CSDL
        records = pre.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return records;
}
```

Bước 3: Cài đặt controller

Tạo mới một Servlet có tên DeleteCategoryServlet (hoặc tên tùy chọn), servlet này

sẽ nhận id từ trang danh sách, rồi sau đó gọi DAO để thực hiện việc xóa trong CSDL theo id, rồi forward lại trang danh sách.

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
        //lay ma loai tin duoc gui tu trang danh sach sang
        int id = Integer.parseInt(request.getParameter("id"));

        //goi DAO de thuc thi viec xoa du lieu
        NewsCategoryDAO.delete(id);

        //chuyen tiep den view tiep theo
        RequestDispatcher dispatch = getServletContext().
            getRequestDispatcher("/ListCategoryServlet");
        dispatch.forward(request, response);
    } finally {
        out.close();
    }
}
```

Bước 4: Thực thi chương trình và xem kết quả

Sau khi đã viết mã xong, ta hãy biên dịch rồi thực thi chương trình để xem kết quả.

Danh sách loại tin

Mã	Nội dung	Mô tả	Sửa	Xóa
7	Tin thể thao	Barca thắng ManCity 3-0	Sửa	Xóa
8	Tin giao thông	Khai trương đường cao tốc HN - LC	Sửa	Xóa
9	Tin kinh tế	Thành lập doanh nghiệp startup	Sửa	Xóa
10	Tin giáo dục	Phát minh ra phương pháp đào tạo mới	Sửa	Xóa
16	Demo	Tin mới nhận	Sửa	Xóa

[Thêm mới loại tin](#)

Sau khi người dùng chọn xóa một bản ghi, hệ thống sẽ hiển thị hộp thoại xác nhận.

Danh sách loại tin

Mã	Nội dung	Mô tả	Sửa	Xóa
7	Tin thể thao	Barca thắng ManCity 3-0	Sửa	Xóa
8	Tin giao thông	Khai trương đường cao tốc HN - LC	Sửa	Xóa
9	Tin kinh tế	Thành lập doanh nghiệp startup	Sửa	Xóa
10	Tin giáo dục	Phát minh ra phương pháp đào tạo mới	Sửa	Xóa
16	Demo	Tin mới nhận	Sửa	Xóa

[Thêm mới loại tin](#)

Bạn có thực sự muốn xóa ?

OKHủy bỏ

Nếu người dùng chọn OK, hệ thống sẽ xóa bản ghi tương ứng theo id.

Danh sách loại tin

Mã	Nội dung	Mô tả	Sửa	Xóa
7	Tin thể thao	Barca thắng ManCity 3-0	Sửa	Xóa
8	Tin giao thông	Khai trương đường cao tốc HN - LC	Sửa	Xóa
9	Tin kinh tế	Thành lập doanh nghiệp startup	Sửa	Xóa
10	Tin giáo dục	Phát minh ra phương pháp đào tạo mới	Sửa	Xóa

[Thêm mới loại tin](#)

Đây là toàn bộ cấu trúc thư mục của Project như sau:

