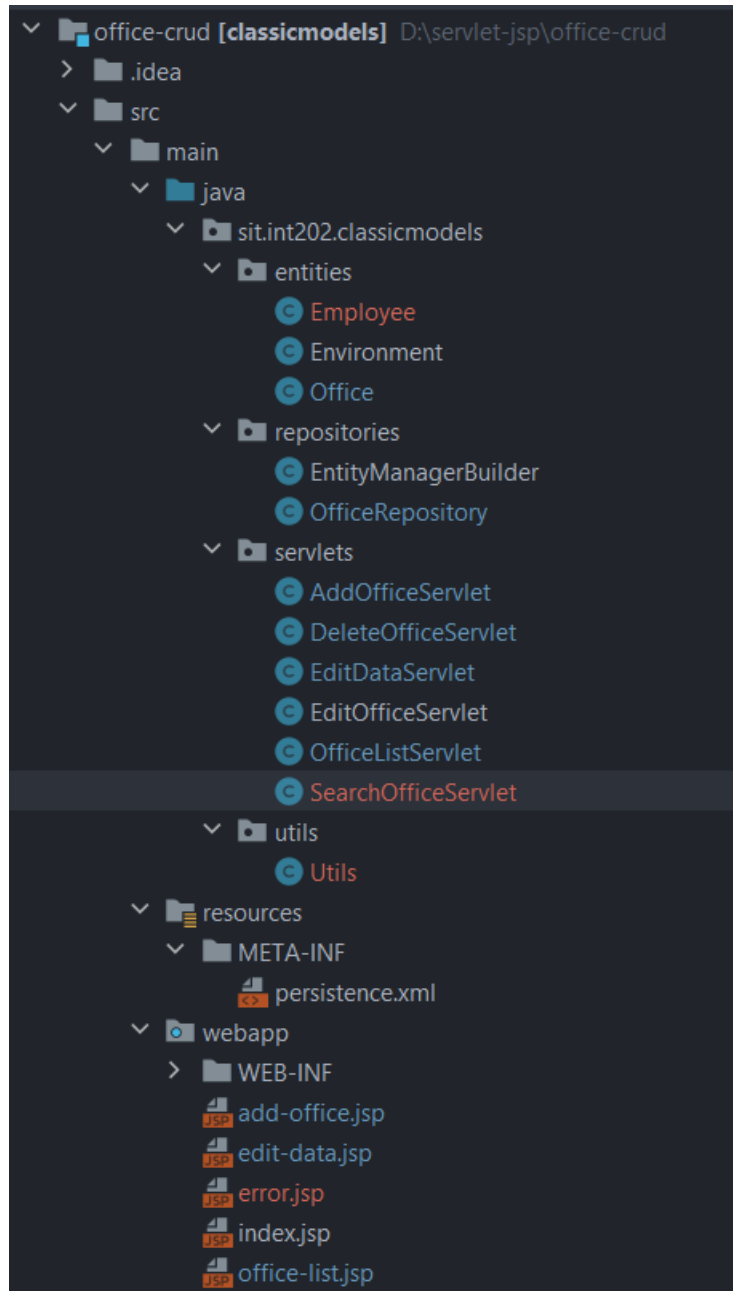


Office CRUD

65130500067 นายมณฑล มักคั่น

ก่อนอื่นเลยขอให้ดู project structure ที่ผมวางไว้คือมีอะไรบ้าง



1.entities เป็น model ของ application ซึ่งในระบบนี้มี model 2 ตัวคือ Office และ Employee
แต่ยังมี Environment คือกำหนดชื่อของ persistence unit เพื่อที่จะให้เขาไปใช้ต่อใน class
EntityManagerBuilder ตอนที่เรสร้าง EntityManager

2.repositories เป็น class ที่เอาไว้ติดต่อกับ database หรือเขานิยมเรียกว่าเป็น Data Access Object Layer ซึ่งก็จะเอาไว้ในการ Create, Read, Update, Delete ข้อมูลใน database

แต่ยังมี EntityManagerBuilder เอาไว้ createEntityManager ไว้ให้ Repository นำไปใช้

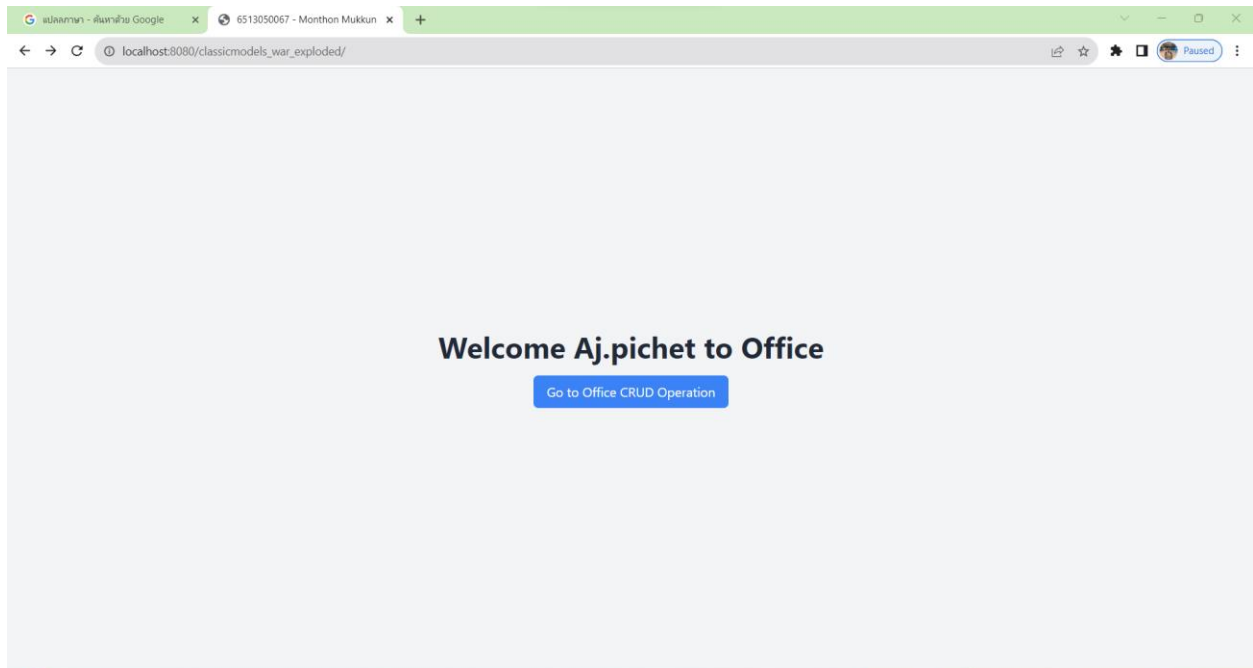
3.servlets เป็น controller ที่เอาไว้จัดการกับ request ที่ส่งมา รวมถึงทำ url mapping พร้อมกันไป เรียกใช้ Data Access Object Layer มาแล้วทำ business logic หรือ validate input ใบบางอย่าง และ set พวก attributes ส่งไปให้ jsp (view layer)

4.util คือเป็นสิ่งที่ผมเอาไว้เก็บ method ที่เป็นประโยชน์นั้นก็คือการ validate input ต่างๆที่ส่งเข้ามาจาก jsp เช่น validateNotEmpty, validateInteger เป็นต้น

5.webapp เป็นส่วนของ view ที่เอาไว้แสดงผลหน้าจอ UI ให้ user เห็น มีทั้งการลึงไปหน้าอื่น (GET) มีแบบฟอร์มให้ add ข้อมูล, update ข้อมูล, delete ข้อมูล (POST) เป็นต้น

Workflow web app + code

Workflow คือ เมื่อผมได้ทำการ deploy web app ของเราขึ้นมา ก็จะเข้าไปที่ context path url ซึ่งจะไม่ path ย่อยต่อท้ายจะทำการไปเรียก default ซึ่งก็คือ index.jsp ดังนั้นก็จะ return หน้า html ที่ทำไว้ออกมา



ตัวอย่างของหน้า web index.jsp

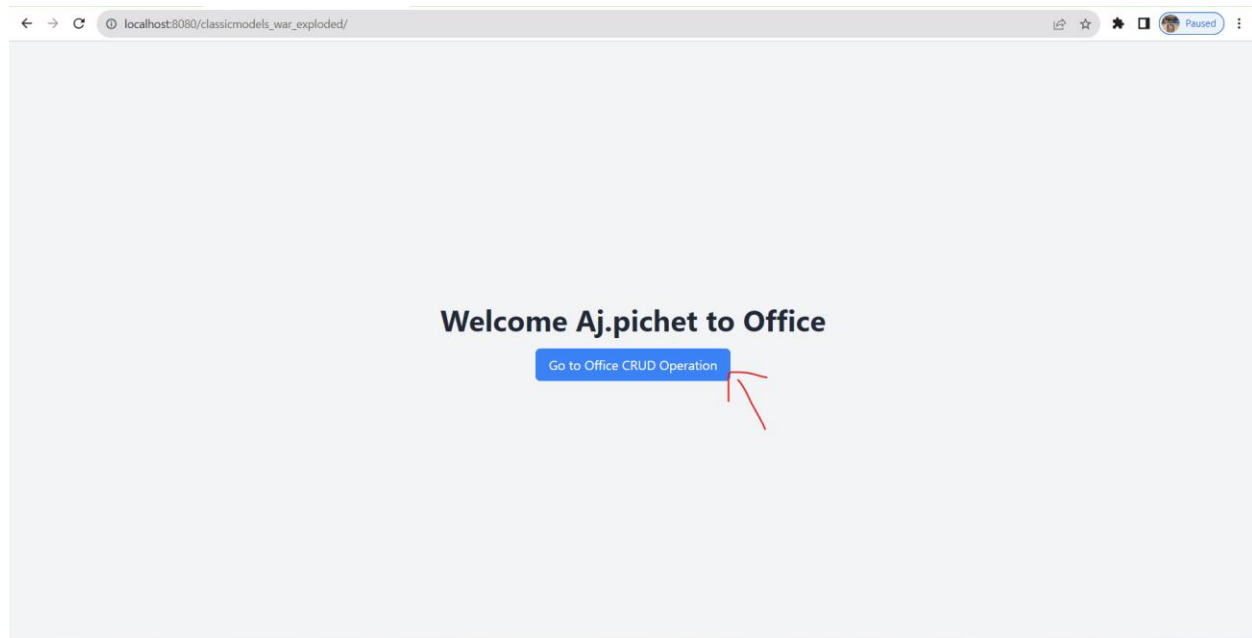
```

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
<head>
  <title>6513050067 - Monthon Mukkun</title>
  <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="stylesheet">
</head>
<body class="flex items-center justify-center h-screen bg-gray-100">
  <div class="text-center">
    <h1 class="text-4xl font-bold mb-4 text-gray-800">Welcome Aj.pichet to Office</h1>
    <a class="inline-block px-4 py-2 bg-blue-500 text-white rounded-md" href="office-list">Go to Office
  </div>
</body>
</html>

```

index.jsp

อธิบายcode คือผมได้สร้างหน้า html พร้อมกับตกแต่งด้วย css โดยใช้ tailwind css แล้วได้มีการใช้ div ทำให้มันอยู่ตรงกลางของ website แล้วมี tag <h1> ซึ่งก็มีข้อความ Welcome Aj.pichet to Office และมี tag <a> มีข้อความ Go to Office ซึ่งจะมี href เพื่อที่จะ link ไปที่ path office-list ซึ่งในส่วนนี้มันก็คือ url mapping ที่ผมได้ทำใน officeListServlet



เมื่อมีการ click ที่ button ตรงนี้ก็จะไปเรียก url mapping ของ OfficeListServlet ที่แสดงไว้ด้านล่าง

```

@WebServlet(name = "OfficeListServlet", value = "/office-list")
public class OfficeListServlet extends HttpServlet {

    3 usages carrynong2 *

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException {
        HttpSession session = request.getSession(b: true);
        OfficeRepository officeRepository = new OfficeRepository();
        session.setAttribute(s: "offices", officeRepository.findAll());
        getServletContext().getRequestDispatcher(s: "/office-list.jsp").forward(request, response);
    }

    2 usages carrynong2 *

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException {
        doGet(request, response);
    }
}

```

OfficeListServlet

อธิบายcode คือถ้ามีการ click tag <a> เมื่อกี้ก็คือการเรียกโดย url mapping /office-list ผ่าน method GET ก็จะทำ method doGet() ซึ่งใน method doGet() ผมได้ทำการทำสร้าง session ไว้เพื่อเก็บ List<Office> ทั้งหมดไว้ที่มาจาก method findAll() ใน officeRepository ซึ่งก็ทำการ setAttribute ที่ชื่อว่า "offices" ถ้าถามว่าทำไมถึงใช้ session เก็บเพราะจะทำให้ข้อมูลที่ผมได้ทำการ add, delete เมื่อมีการ click ไปหน้าอื่นหรือทำ request ไรบางอย่างไม่ให้ข้อมูลผิดไปจาก logic ทั่วไปเช่น add ก็ควรเพิ่มเข้าไปไม่ใช่เพิ่มเข้าไปแล้ว click หน้าอื่นหรือทำ request ไรบางอย่างแล้วหาย หรือ delete ลบข้อมูลไปแล้วพอ click หน้าอื่น หรือทำ request ไรบางอย่าง แล้วมันไม่ลบ (ดังนั้นเป็นเหตุผลว่าผมใช้ session scope ในการ setAttribute ไม่ใช่ request scope)

จากนั้นผมก็ทำการใช้ getServletContext แล้วทำการ request dispatcher ไปที่หน้า office-list.jsp พร้อมกับ forward(request, response) ไปทำให้หน้า web ตอนนี้จะแสดง html ของ office-list.jsp

OfficeCode	City	Phone	AddressLine1	Country	PostalCode	Territory	Actions
1	San Francisco	+1 650 219 4782	100 Market Street	USA	94080	NA	Edit Delete
2	Boston	+1 215 837 0825	1550 Court Place	USA	02107	NA	Edit Delete
3	USA	+1 212 555 3000	523 East 53rd Street	USA	10022	NA	Edit Delete
4	Paris	+33 14 723 4404	43 Rue Jouffroy D'abbans	France	75017	EMEA	Edit Delete
5	Tokyo	+81 33 224 5000	4-1 Kioicho	Japan	102-8578	Japan	Edit Delete
6	Sydney	+61 2 9264 2451	5-11 Wentworth Avenue	Australia	NSW 2010	APAC	Edit Delete
7	UK	+44 20 7877 2041	25 Old Broad Street	UK	EC2N 1HN	EMEA	Edit Delete
8	Sanfransico	0639042389	120 Sukhumwit	Thailand	92170	NA	Edit Delete

ตัวอย่างหน้า web ของ office-list.jsp

```

<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
<head>
  <title>6513050067 - Monthon Mukkun</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" %>
</head>
<body>
<div class="container mx-auto">
  <h2 class="bg-primary p-2 text-white">Classic Model Offices ::</h2>
  <table class="table-auto w-full">
    <thead>
      <tr>
        <th class="px-4 py-2">OfficeCode</th>
        <th class="px-4 py-2">City</th>
        <th class="px-4 py-2">Phone</th>
        <th class="px-4 py-2">AddressLine1</th>
        <th class="px-4 py-2">Country</th>
        <th class="px-4 py-2">PostalCode</th>
        <th class="px-4 py-2">Territory</th>
        <th class="px-4 py-2">Actions</th>
      </tr>
    </thead>

```

office-list.jsp (1)

code อธิบายคือผมได้มีการใช้ css framework ของ bootstrap รวมถึงมีการใช้ jstl เลยต้องทำการ import taglib jstl core เข้ามา จากนั้นก็ทำการสร้าง div ครอบที่เหลือเอาไว้ แล้วมี tag <h2> มีข้อความ Classic Model Offices :: แล้วได้ทำการสร้าง table ที่มีหัวของตารางอยู่ 8 column แต่ว่าจะมีข้อมูลของ offices อยู่แค่ 7 column (ผมเลือก 7 column นี้เพราะว่ามันคือ attribute ที่เป็น Not null ได้ db) และมีปุ่ม Actions ซึ่งจะมี edit กับ delete ข้อมูลในของ office ของ id นั้น

```

<tbody>
  <c:forEach items="{offices}" var="office">
    <tr>
      <td class="border px-4 py-2">${office.officeCode}</td>
      <td class="border px-4 py-2">${office.city}</td>
      <td class="border px-4 py-2">${office.phone}</td>
      <td class="border px-4 py-2">${office.addressLine1}</td>
      <td class="border px-4 py-2">${office.country}</td>
      <td class="border px-4 py-2">${office.postalCode}</td>
      <td class="border px-4 py-2">${office.territory}</td>
      <td class="border px-4 py-2">
        <a href="editOffice?officeCode=${office.officeCode}" class="text-blue-500">Edit</a>
        <a href="deleteOffice?officeCode=${office.officeCode}" class="text-red-500 ml-2">Delete</a>
      </td>
    </tr>
  </c:forEach>
  <div class="container mt-3">
    <div class="row">
      <div class="col-md-6 d-flex justify-content-start">
        <a href="addOffice" class="text-blue-500">Add</a>
      </div>
      <div class="col-md-6 d-flex justify-content-end w-50">
        <form action="searchOffice" method="post">
          <input name="searchByCityOrCountry" style="..." type="text" placeholder="Search By City or Country">
        </form>
      </div>
    </div>
  </div>

```

office-list.jsp (2)

code อธิบายคือผมได้ทำการใช้ forEach ที่มีใน jstl ซึ่งผมได้ทำการ loop “offices” ที่มีการ setAttribute ไว้ซึ่งได้ตั้งเพื่อแสดงข้อมูลของแต่ละ rows รวมถึงมีการตั้งชื่อตัวแปรของแต่ละ object ว่า office จากนั้นก็ดึงค่าของแต่ละ attribute ของ office มาแสดงด้วย syntax \${office.attribute} ก็จะแสดงว่าตรงกับ column <thead> ที่เคยเขียนไว้ในตอนแรกนั่นเอง

อีก 2 ส่วนที่สำคัญที่ Actions ผมได้ทำการสร้างปุ่ม 2 ปุ่มคือ edit และ delete ซึ่งได้ใช้ tag <a> เพื่อที่จะได้ใช้ href เพราะผมจะทำการ link ไปยัง url mapping ของ servlet ที่ชื่อว่า editOffice กับ deleteOffice พร้อมทั้งกำหนด query string ของทั้ง 2 ตัวคือ officeCode (key) และดึง officeCode (value) ของแต่ละ object ที่ loop มานั่นเองทำให้สามารถที่จะ update หรือ delete ได้ถูกต้องตาม id

```
</c:forEach>
<div class="container mt-3">
  <div class="row">
    <div class="col-md-6 d-flex justify-content-start">
      <a href="addOffice" class="text-blue-500">Add</a>
    </div>
    <div class="col-md-6 d-flex justify-content-end w-50">
      <form action="searchOffice" method="post">
        <input name="searchByCityOrCountry" style="..." type="text" placeholder="Search By City or Country" />
        <button type="submit" class="btn btn-primary">Search</button>
      </form>
    </div>
  </div>
</div>
</tbody>
</table>
```

office-list.jsp (3)

code อธิบายคือผมได้ทำการสร้าง div และตกแต่งโดยใช้ bootstrap แบ่งซ้ายขวาด้วยการใช้ flex ซึ่งส่วนแรก (ด้านซ้าย) คือปุ่ม Add ได้ใช้ tag <a> พร้อมกับระบุ path ที่ต้องการ link ไปในที่นี้คือ addOffice ซึ่งจะไป map กับ url mapping ของ AddOfficeServlet ซึ่งส่วนที่สอง (ด้านขวา) คือมี form (method post) พร้อมกับทำ action (searchOffice) ในการส่งข้อมูลใน form ไปที่ url mapping searchByCityOrCountry ซึ่งมี 2 tag คือ input เอาไว้ให้ระบุตอนที่ผมจะทำการ searchByCityOrCountry ซึ่งจะส่งผ่าน attribute name และปุ่ม button ในการกด submit

```

<c:choose>
  <c:when test="${statusAdd}">
    <h5 class="text-success">${messageAdd}</h5>
  </c:when>
  <c:otherwise>
    <h5 class="text-danger">${messageAdd}</h5>
  </c:otherwise>
</c:choose>
<c:choose>
  <c:when test="${statusUpdate}">
    <h5 class="text-success">${messageUpdate}</h5>
  </c:when>
  <c:otherwise>
    <h5 class="text-danger">${messageUpdate}</h5>
  </c:otherwise>
</c:choose>
<c:choose>
  <c:when test="${statusDelete}">
    <h5 class="text-success">${messageDelete}</h5>
  </c:when>
  <c:otherwise>
    <h5 class="text-danger">${messageDelete}</h5>
  </c:otherwise>
</c:choose>

```

office-list.jsp (4)

code อธิบายคือส่วนสุดท้ายในหน้านี้จะทำการ check status attribute ที่ส่งมาจากตัวของ servlet ว่ามีการ add, update, delete สำเร็จไหม ด้วยการใช้ jstl tag `<c:choose>` ซึ่งจะมี 2 ส่วนของ `<c:when test=condition>` ถ้าเงื่อนไขนี้เป็นจริงให้ทำคำสั่งนี้ ซึ่งคำสั่งที่ผมให้ทำคือ tag `<h5>` ที่มี class text-success (text สีเขียว) หมายความว่าได้มีการ add, update, delete ข้อมูลจาก database สำเร็จและอีกส่วนของ `<c:otherwise>` คือส่วนของ false หมายความว่า add, update, delete ข้อมูลจาก database ไม่สำเร็จ (แต่ในส่วนของการ delete จะสำเร็จเสมอเพราะว่าผมไม่ได้เขียน validate ดักอะไรไว้เหมือนกับ add, update) จะแสดง tag `<h5>` ที่ class text-danger (text สีแดง)

```

if (Utils.validateIsEmpty(officelCode, city, phone, addressLine1, country, postalCode, territory)
    && Utils.validateIsInteger(officelCode) && Utils.validateIsLetter(city) &&
    && Utils.validateIsPhone(phone) && !Utils.validateHaveSomeSpecial(addressLine1) && Utils.validateIsLetter(country)
    && !Utils.validateHaveSomeSpecial(postalCode) && Utils.validateIsLetter(territory)) {
    boolean statusAdd = officeRepository.insert(new Office(officelCode, city, phone, addressLine1,
    if (statusAdd) {
        request.setAttribute("messageAdd", "Add office successfully");
        request.setAttribute("statusAdd", true);
    } else {
        request.setAttribute("messageAdd", "Add office not successfully");
        request.setAttribute("statusAdd", false);
    }
} else {
    request.setAttribute("messageAdd", "Add office not successfully");
    request.setAttribute("statusAdd", false);
}
}

```

อันนี้เป็นตัวอย่างที่ผมส่งค่า attribute มาจาก servlet

Classic Model Offices ::

[Add](#)

OfficeCode	City	Phone	AddressLine1	Country	PostalCode	Territory	Actions
1	San Francisco	+1 650 219 4782	100 Market Street	USA	94080	NA	Edit Delete
2	Boston	+1 215 837 0825	1550 Court Place	USA	02107	NA	Edit Delete
3	USA	+1 212 555 3000	523 East 53rd Street	USA	10022	NA	Edit Delete
4	Paris	+33 14 723 4404	43 Rue Jouffroy D'abbans	France	75017	EMEA	Edit Delete
5	Tokyo	+81 33 224 5000	4-1 Kioicho	Japan	102-8578	Japan	Edit Delete
6	Sydney	+61 2 9264 2451	5-11 Wentworth Avenue	Australia	NSW 2010	APAC	Edit Delete
7	UK	+44 20 7877 2041	25 Old Broad Street	UK	EC2N 1HN	EMEA	Edit Delete
8	Sanfransico	0639042389	120 Sukhumvit	Thailand	92170	NA	Edit Delete

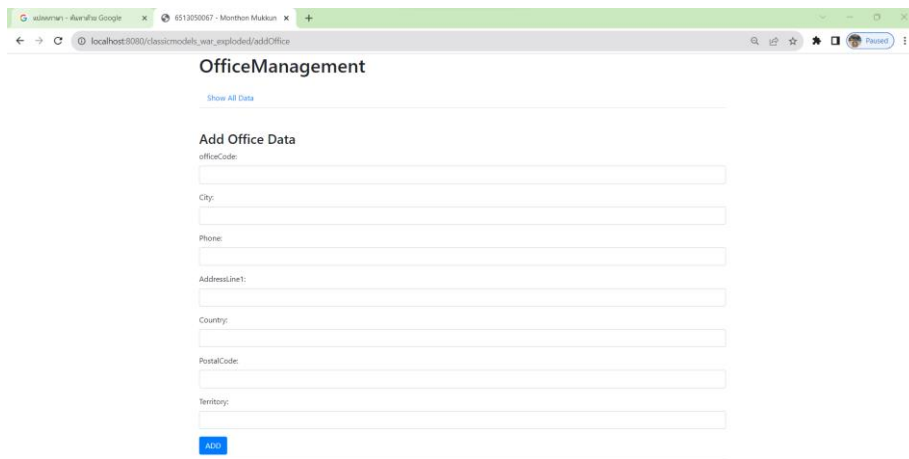
ต่อไปนี่ก็มีหลาย option ให้ผมเลือกที่จะเล่นคือ add Office, Update Office (edit), search Office, delete Office ซึ่งผมจะทำการ Add ให้ดูก่อน

```
<div class="col-md-6 d-flex justify-content-start">
  <a href="addOffice" class="text-blue-500">Add</a>
</div>
```

จาก code คือเมื่อผม click ปุ่ม add ก็จะทำให้การ link ไปที่ path addOffice ซึ่งจะไปการผ่าน method GET

```
@WebServlet(name = "AddOfficeServlet", value = "/addOffice")
public class AddOfficeServlet extends HttpServlet {
    3 usages carrynong2
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException {
        request.getRequestDispatcher(s: "/add-office.jsp").forward(request, response);
    }
    2 usages carrynong2 *
```

ดังนั้นผมจึงทำการสร้าง AddOfficeServlet ที่มี url mapping /addOffice และได้สร้าง method doGet() มาเพื่อรับ request พร้อมกับใช้ getRequestDispatcher() เพื่อทำการ forward ไปที่หน้า add-office.jsp เพื่อนำมาแสดงในหน้า website



ตัวอย่างของหน้า add-office.jsp

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <title>651305067 - Monthon Mukkun</title>
</head>
<div class="container">
  <h1 class="mb-4">OfficeManagement</h1>
  <ul class="nav nav-tabs">
    <li class="nav-item">
      <a class="nav-link" href="office-list">Show All Data</a>
    </li>
  </ul>
  <h3 class="mt-5">Add Office Data</h3>
  <form action="addOffice" method="POST">
    <div class="form-group">
      <label for="officeCode">officeCode:</label>
      <input type="text" class="form-control" name="officeCode" >
    </div>
    <div class="form-group">
      <label for="city">City:</label>
      <input type="text" class="form-control" name="city" >
    </div>
  </form>
</div>
```

อธิบาย code คือผมได้มีการใช้ bootstrap เข้ามาในการตกแต่ง website ซึ่งได้ทำการสร้าง div จัดไว้ตรงกลาง ที่ครอบ tag <h1> ที่มีข้อความ OfficeMangement อีกทั้งยังมีการสร้าง tag <a> ที่มีข้อความว่า Show All Data ซึ่งจะทำให้ link ไปยัง path office-list ซึ่ง path นี้ก็ได้มีการทำ url mapping office-list ในส่วนของ OfficeListServlet อย่างที่เคยเขียนไว้ข้างบน

```

<label for="city">City:</label>
<input type="text" class="form-control" name="city" >
</div>
<div class="form-group">
<label for="phone">Phone:</label>
<input type="text" class="form-control" name="phone">
</div>
<div class="form-group">
<label for="addressLine1">AddressLine1:</label>
<input type="text" class="form-control" name="addressLine1">
</div>
<div class="form-group">
<label for="country">Country:</label>
<input type="text" class="form-control" name="country">
</div>
<div class="form-group">
<label for="postalCode">PostalCode:</label>
<input type="text" class="form-control" name="postalCode">
</div>
<div class="form-group">
<label for="territory">Territory:</label>
<input type="text" class="form-control" name="territory">
</div>
<button type="submit" class="btn btn-primary">ADD</button>
</form>

```

และก็จะมีส่วนคือแบบฟอร์มที่ใช้สำหรับการ add office คือผมได้ทำการสร้าง form ที่มี method POST แล้วทำการส่ง action ไปที่ path addOffice ซึ่งก็คือ AddOfficeServlet ในส่วนของ doPost() นั้นเองเดี่ยวผมจะมาอธิบายโค้ด ใน doPost() อีกทีขอมาอธิบาย input ที่ส่งไปก่อน ซึ่ง input ที่ผมจะมี label เขียนอธิบายไว้มี 7 ตัว คือ OfficeCode, City, Phone, AddressLine1, Country, PostalCode, Territory ซึ่งเป็น Pascal Case ทั้งหมดเลยแต่ว่าตอนส่ง input ใน attribute name คือส่งไปแบบ camelCase

The screenshot shows a web browser window with the URL `localhost:8080/classicmodels_war_exploded/addOffice`. The page displays a form titled "Add Office Data". The form has the following fields and values:

- officeCode: 8
- City: Sanfransico
- Phone: 0639042389
- AddressLine1: 120 Sukhumwit
- Country: Thailand
- PostalCode: 92170
- Territory: NA

At the bottom of the form is a blue button labeled "ADD". A red arrow points to this button.

ดังนั้นเมื่อผมทำการ click ที่ button ADD ก็จะทำให้การไปเรียก /addOffice ใน AddOfficeServlet

```
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException {
    HttpSession session = request.getSession(b: true);
    String officeCode = request.getParameter(s: "officeCode");
    String city = request.getParameter(s: "city");
    String phone = request.getParameter(s: "phone");
    String addressLine1 = request.getParameter(s: "addressLine1");
    String country = request.getParameter(s: "country");
    String postalCode = request.getParameter(s: "postalCode");
    String territory = request.getParameter(s: "territory");
    OfficeRepository officeRepository = new OfficeRepository();
    if (Utils.validateIsEmpty(officeCode, city, phone, addressLine1, country, postalCode, territory)
        && Utils.validateIsInteger(officeCode) && Utils.validateIsLetter(city) &&
        Utils.validateIsPhone(phone) && Utils.validateHaveSomeLetter(addressLine1) &&
        Utils.validateHaveSomeInteger(addressLine1) && Utils.validateIsLetter(country)
        && !Utils.validateHaveSomeSpecial(postalCode) && Utils.validateIsLetter(territory)) {
        boolean statusAdd = officeRepository.insert(new Office(officeCode, city, phone, addressLine1,
        if (statusAdd) {
            request.setAttribute(s: "messageAdd", o: "Add office successfully");
            request.setAttribute(s: "statusAdd", o: true);
        } else {
            request.setAttribute(s: "messageAdd", o: "Add office not successfully");
            request.setAttribute(s: "statusAdd", o: false);
        }
    }
}

} else {
    request.setAttribute(s: "messageAdd", o: "Add office not successfully");
    request.setAttribute(s: "statusAdd", o: false);
}
session.setAttribute(s: "offices", officeRepository.findAll());
request.getRequestDispatcher(s: "/office-list.jsp").forward(request, response);
}
```

เมื่อมี method POST request เข้ามานั้นก็จะเข้าการเรียกใช้ method doPost() ซึ่งผมได้ทำการสร้าง session ไว้ก่อนเดี๋ยวมานอธิบายทีหลัง ซึ่งจากที่ใน jsp ส่งข้อมูลมา 7 input ผมก็ทำการรับ request.getParameter() 7 ตัวที่ส่งเข้ามาพร้อมกับตั้งชื่อตัวแปร จากนั้นผมก็ทำการสร้าง OfficeRepository เพื่อที่จะมา insert object Office เข้า database

แต่ก่อนจะ insert object Office เข้า database นั้นผมก็ได้ทำการ validate input ที่ส่งเข้ามาคือจะมีอยู่ด้วยกัน 7 method ซึ่งผมได้ทำการสร้าง package utils แยกเอาไว้แล้วตั้งชื่อ class ว่าเป็น Utils พร้อมกับทำให้ทั้ง 7 method เป็น static เพื่อที่จะให้เรียกใช้ได้ง่าย

```

package sit.int202.classicmodels.utils;

20 usages
public class Utils {

    2 usages
    public static boolean validateIsNotEmpty(String officeCode, String city, String phone, String addressLine1,
                                             String country, String postalCode, String territory) {
        if (officeCode != null && city != null && phone != null && addressLine1 != null &&
            country != null && postalCode != null && territory != null && !officeCode.isEmpty() &&
            !city.isEmpty() && !phone.isEmpty() && !addressLine1.isEmpty() && !country.isEmpty() &&
            !postalCode.isEmpty() && !territory.isEmpty()) {
            return true;
        }
        return false;
    }

    2 usages
    public static boolean validateIsInteger(String input) {
        boolean flag=true;
        for (int i = 0; i < input.length(); i++) {
            if(!Character.isDigit(input.charAt(i)))
                flag=false;
        }
        return flag;
    }
}

```

นี่คือ 2 method ใน utils

Method ที่ 1 คือ validateIsNotEmpty ซึ่งรับ parameter เข้ามา 7 ตัวมีการ check ว่าทั้ง 7 input ที่ส่งเข้ามานั้นไม่ใช่ null และ ไม่เป็น empty ถึงจะ return true ไม่งั้นถ้ามีอย่างน้อย 1 ตัวไม่ผ่าน case ก็จะมี return false

Method ที่ 2 คือ validateIsInteger ซึ่งรับ parameter เข้ามา 1 ตัวซึ่ง logic คือการสร้างค่า Boolean flag = true ไว้ก่อน จากนั้นก็ทำการ for loop เพื่อ check ว่าถ้ามีอย่างน้อย 1 ตัวที่ไม่ใช่ integer ให้ทำการ set flag เป็น false จากนั้นก็ return ค่า flag

```

public static boolean validateIsLetter(String input) {
    boolean flag = true;
    for (int i = 0; i < input.length(); i++) {
        if (input.charAt(i) == ' ') continue;
        if (!Character.isLetter(input.charAt(i))) {
            flag = false;
        }
    }
    return flag;
}

2 usages
public static boolean validateHaveSomeSpecial(String postalCode) {
    boolean flag = false;
    for (int i = 0; i < postalCode.length(); i++) {
        if (postalCode.charAt(i) == ' ' || postalCode.charAt(i) == '-') continue;
        if (!Character.isLetter(postalCode.charAt(i)) && !Character.isDigit(postalCode.charAt(i))) {
            flag = true;
        }
    }
    return flag;
}

```

นี่คืออีก 2 method ใน utils

Method ที่ 3 คือ validateIsInteger ซึ่งรับ parameter 1 ตัวซึ่ง logic ได้สร้าง Boolean flag = true ไว้ก่อนจากนั้นก็ for loop เข้าไปทำการ check ว่าเป็นตัวเลขหรือไม่ให้ข้ามไปรอบต่อไป และอีกเงื่อนไขหนึ่งคือ check ถ้ามีอย่างน้อย 1 ตัวที่ไม่ใช่ตัวอักษรก็ให้ set ให้มันเป็น false ถ้าเป็นตัวอักษรหมดก็ return true

Method ที่ 4 คือ validateHaveSomeSpecial ซึ่งรับ parameter 1 ตัวซึ่ง logic ได้สร้าง Boolean flag = false ไว้ก่อน จากนั้นก็ for loop เข้าไป check ว่าเป็นตัวเลขหรือเป็น - (เหตุผลที่มี - เพราะว่าผมเอามาใช้ validate input postalCode โดยเฉพาะซึ่ง postalCode จะมีกรณีที่มี - ถ้าสามารถ update ได้) ก็ให้ข้ามไปรอบต่อไป และอีกเงื่อนไขหนึ่งคือ check ถ้าไม่ใช่ตัวอักษรและตัวเลข อย่างน้อย 1 ตัวให้ set ค่า flag เป็น true จากนั้นก็ return ค่า flag ที่เป็น Boolean

```

2 usages
public static boolean validateIsPhone(String phone) {
    boolean flag=true;
    for (int i = 0; i < phone.length(); i++) {
        if (phone.charAt(i) == '+' || phone.charAt(i) == '-' || phone.charAt(i) == ' ') {
            continue;
        }
        if(!Character.isDigit(phone.charAt(i)))
            flag=false;
    }
    return flag;
}

```

นี่คืออีก 1 method ใน utils

Method ที่ 5 คือ validateIsPhone ซึ่งรับ parameter 1 ตัวจากนั้นก็สร้าง Boolean flag = true แล้วทำการ for loop check ว่าถ้ามี +, -, เว้นวรรค ให้ข้ามไปรอบถัดไปเพราะว่าผมอยากให้ phone มีเว้นวรรคได้, มี + ได้และมี - ได้

```

public static boolean validateHaveSomeLetter(String addressLine1) {
    for (int i = 0; i < addressLine1.length(); i++) {
        if(Character.isLetter(addressLine1.charAt(i)) || addressLine1.charAt(i) == '\'' ||
            addressLine1.charAt(i) == '-') {
            return true;
        }
    }
    return false;
}

2 usages
public static boolean validateHaveSomeInteger(String addressLine1) {
    for (int i = 0; i < addressLine1.length(); i++) {
        if(Character.isDigit(addressLine1.charAt(i)) || addressLine1.charAt(i) == '\'' ||
            addressLine1.charAt(i) == '-') {
            return true;
        }
    }
    return false;
}

```

นี่คืออีก 2 method ใน utils

Method ที่ 6 คือ validateHaveSomeLetter ซึ่งรับ parameter 1 ตัว โดยมีการใช้ for loop แล้ว check ว่าถ้าเป็นตัวอักษรหรือ ' หรือ - ให้ return true (เหตุผลที่เพิ่ม ' กับ - มาใน case เพราะว่าผมเอามาใช้แค่กับ validate addressLine1 โดยเฉพาะซึ่งจะมีกรณีที่มี ' กับ - สามารถ update ได้)

Method ที่ 7 คือ validateHaveSomeInteger ซึ่งรับ parameter 1 ตัว โดยมีการใช้ for loop แล้ว check ว่าถ้าเป็นตัวเลขหรือ ' หรือ - ให้ return true (เหตุผลที่เพิ่ม ' กับ - มาใน case

เพราะว่าผมเอามาใช้แค่กับ validate addressLine1 โดยเฉพาะซึ่งจะมีกรณีที่มี ' กับ – สามารถ update ได้

ตัวอย่างของ TestCase ที่ผ่าน (ADD) ได้

OfficeCode : 8, City: USA, Phone: 0639041212, AddressLine1: 120 Sukhumwit, Country: Thailand, PostalCode: 92170, Territory: NA

เพราะทุก field ผ่านเงื่อนไขที่ผม validate ไว้ทั้งหมด

ตัวอย่างของ TestCase ที่ไม่ผ่าน (ADD) ไม่ได้

OfficeCode : a9, City: USA, Phone: 0639041212, AddressLine1: 120 Sukhumwit, Country: Thailand, PostalCode: 92170, Territory: NA

เพราะ field OfficeCode ไม่เป็นตัวเลขทั้งหมด (มีตัวอักษรหรืออักขระพิเศษก็ได้ไม่ได้)

OfficeCode : 9, City: USA, Phone: 0639041212za, AddressLine1: 120 Sukhumwit, Country: Thailand, PostalCode: 92170, Territory: NA

เพราะ field Phone ไม่เป็นตัวเลขทั้งหมด (มีตัวอักษรไม่ได้ มีอักขระพิเศษได้แค่บางตัว คือ +, -)

OfficeCode : 9, City: USA, Phone: 0639041212, AddressLine1: 120, Country: Thailand, PostalCode: 92170, Territory: NA

เพราะ field AddressLine1 เป็นตัวเลขทั้งหมด ไม่มีตัวตัวอักษรอย่างน้อย 1 ตัว

OfficeCode : 9, City: USA, Phone: 0639041212, AddressLine1: 120 Sukhumwit, Country: Thailand555, PostalCode: 92170, Territory: NA

เพราะ field Country ไม่เป็นตัวอักษรทั้งหมด มีตัวเลขอย่างน้อย 1 ตัวก็ไม่ได้

OfficeCode : 9, City: USA, Phone: 0639041212, AddressLine1: 120 Sukhumwit, Country: Thailand, PostalCode: 92170@@@, Territory: NA

เพราะ field PostalCode มีตัวอักขระพิเศษอย่างน้อย 1 ตัว

OfficeCode : 9, City: USA, Phone: 0639041212, AddressLine1: 120 Sukhumwit, Country: Thailand, PostalCode: 92170, Territory: NA555

เพราะ field Territory ไม่เป็นตัวอักษรทั้งหมด มีตัวเลขอย่างน้อย 1 ตัวก็ไม่ได้

```

    && !Utils.validateHaveSomeSpecial(postalCode) && Utils.validateIsLetter(territory)) {
        boolean statusAdd = officeRepository.insert(new Office(officeCode, city, phone, addressLine1, cou
        if (statusAdd) {
            request.setAttribute(s: "messageAdd", o: "Add office successfully");
            request.setAttribute(s: "statusAdd", o: true);
        } else {
            request.setAttribute(s: "messageAdd", o: "Add office not successfully");
            request.setAttribute(s: "statusAdd", o: false);
        }
    } else {
        request.setAttribute(s: "messageAdd", o: "Add office not successfully");
        request.setAttribute(s: "statusAdd", o: false);
    }
}
session.setAttribute(s: "offices", officeRepository.findAll());
request.getRequestDispatcher(s: "/office-list.jsp").forward(request, response);
}

```

จากที่ผ่านมามี input ที่ส่งเข้ามาตรงตามเงื่อนไขผมก็ทำการ insert object office เข้าไปใน db แล้วสร้างตัวแปร statusAdd เพื่อเก็บค่าว่า insert object เข้า db สำเร็จไหมถ้าสำเร็จก็ให้ทำการ setAttribute แบบ requestScope (เมื่อมีการ click หรือ refresh attribute ก็จะหายไป) โดยผมทำการ setAttribute อยู่ 2 ตัวตามเงื่อนไขคือถ้า statusAdd เป็น true ให้ messageAdd = Add office successfully กับ statusAdd = true และที่เหลือก็ทำกลับกัน

อีกส่วนหนึ่งถ้า input ที่ส่งเข้ามาไม่ผ่านการ validate ก็ให้ไปทำเงื่อนไข else คือการ setAttribute ซึ่ง อยู่ 2 ตัวคือ

messageAdd = Add office not successfully กับ statusAdd = false

และมีการใช้ session setAttribute offices (key) กับ officeRepository.findAll() List<Object>

พร้อมกับ forward ไปที่หน้า office-list.jsp เพื่อนำข้อมูลไปแสดงในหน้า web

Classic Model Offices ::							
Add		Search By City or Country			<input type="button" value="Search"/>		
OfficeCode	City	Phone	AddressLine1	Country	PostalCode	Territory	Actions
1	San Francisco	+1 650 219 4782	100 Market Street	USA	94080	NA	Edit Delete
2	Boston	+1 215 837 0825	1550 Court Place	USA	02107	NA	Edit Delete
3	USA	+1 212 555 3000	523 East 53rd Street	USA	10022	NA	Edit Delete
4	Paris	+33 14 723 4404	43 Rue Jouffroy D'abbans	France	75017	EMEA	Edit Delete
5	Tokyo	+81 33 224 5000	4-1 Kioicho	Japan	102-8578	Japan	Edit Delete
6	Sydney	+61 2 9264 2451	5-11 Wentworth Avenue	Australia	NSW 2010	APAC	Edit Delete
7	UK zaza	+44 20 7877 2041	25 Old Broad Street	UK	5555	EMEA	Edit Delete
8	Sanfransico	0639042389	120 Sukhumwit	Thailand	92170	NA	Edit Delete

Add office successfully

ตัวอย่างเมื่อ กด click Add สำเร็จ

Classic Model Offices ::

[Add](#)

Search By City or Country

Search

OfficeCode	City	Phone	AddressLine1	Country	PostalCode	Territory	Actions
1	San Francisco	+1 650 219 4782	100 Market Street	USA	94080	NA	Edit Delete
2	Boston	+1 215 837 0825	1550 Court Place	USA	02107	NA	Edit Delete
3	USA	+1 212 555 3000	523 East 53rd Street	USA	10022	NA	Edit Delete
4	Paris	+33 14 723 4404	43 Rue Jouffroy D'abbans	France	75017	EMEA	Edit Delete
5	Tokyo	+81 33 224 5000	4-1 Kioicho	Japan	102-8578	Japan	Edit Delete
6	Sydney	+61 2 9264 2451	5-11 Wentworth Avenue	Australia	NSW 2010	APAC	Edit Delete
7	UK zaza	+44 20 7877 2041	25 Old Broad Street	UK	5555	EMEA	Edit Delete
8	Sanfransico	0639042389	120 Sukhumwit	Thailand	92170	NA	Edit Delete

Add office not successfully

ตัวอย่างเมื่อกด click add ไม่สำเร็จ

Classic Model Offices ::

[Add](#)

Search By City or Country

Search

OfficeCode	City	Phone	AddressLine1	Country	PostalCode	Territory	Actions
1	San Francisco	+1 650 219 4782	100 Market Street	USA	94080	NA	Edit Delete
2	Boston	+1 215 837 0825	1550 Court Place	USA	02107	NA	Edit Delete
3	USA	+1 212 555 3000	523 East 53rd Street	USA	10022	NA	Edit Delete
4	Paris	+33 14 723 4404	43 Rue Jouffroy D'abbans	France	75017	EMEA	Edit Delete
5	Tokyo	+81 33 224 5000	4-1 Kioicho	Japan	102-8578	Japan	Edit Delete
6	Sydney	+61 2 9264 2451	5-11 Wentworth Avenue	Australia	NSW 2010	APAC	Edit Delete
7	UK zaza	+44 20 7877 2041	25 Old Broad Street	UK	5555	EMEA	Edit Delete
8	Sanfransico	0639042389	120 Sukhumwit	Thailand	92170	NA	Edit Delete

ต่อไปก็จะมาลองใช้ edit (update)

```
<td class="border px-4 py-2">  
  <a href="editOffice?officeCode=${office.officeCode}" class="text-blue-500">Edit</a>  
  <a href="deleteOffice?officeCode=${office.officeCode}" class="text-red-500 ml-2">Delete</a>  
</td>
```

ซึ่งในส่วนของ code edit (update) ก็จะใช้ tag <a> แล้ว link path ไปที่ editOffice ซึ่งจะทำให้การไปดึง officeCode ของ object ตัวนั้นมาต่อท้ายเป็น query string ส่งไปแบบ method GET

```

@WebServlet(name = "EditOfficeServlet", value = "/editOffice")
public class EditOfficeServlet extends HttpServlet {
    3 usages carrynong2

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException {
        HttpSession session = request.getSession(b: true);
        String officeCode = request.getParameter(s: "officeCode");

        if (officeCode != null) {
            OfficeRepository officeRepository = new OfficeRepository();
            Office office = officeRepository.find(officeCode);
            session.setAttribute(s: "office", office);
        }
        request.getRequestDispatcher(s: "/edit-data.jsp").forward(request, response);
    }

    2 usages carrynong2

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException {
        doGet(request, response);
    }
}

```

จากนั้นใน EditOfficeServlet ก็รับ request ที่ url mapping /editOffice พร้อมกับทำ method doGet() ซึ่งก็จะรับ officeCode เข้ามาพร้อมผ่าน getParamter เก็บเข้าในตัวแปร และมีการสร้าง object OfficeRepository และใช้ method find ของ OfficeRepository ตาม officeCode ซึ่งก็จะได้ Object ของ Office มา

สุดท้ายก็ใช้ session ในการ setAttribute ชื่อ office (key) กับ office object (value) และก็มี forward ไปที่หน้า edit-data.jsp

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/classicmodels_war_exploded/editOffice?officeCode=8'. The page content includes a header 'OfficeManagement' and a link 'Show All Data'. The main section is titled 'Update Office Data' and contains several text input fields. The 'City' field contains 'Sanfransico', 'Phone' contains '0639042389', 'AddressLine1' contains '120 Sukhumwit', 'Country' contains 'Thailand', 'PostalCode' contains '92170', and 'Territory' contains 'NA'. At the bottom of the form are two buttons: 'UPDATE' and 'RESET'.

นี่คือตัวอย่างของหน้า web edit-data.jsp ที่มี การไปดึง attribute object ของ office ที่ setAttribute มาแสดงใน form ของการ update

```

<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@tagLib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
<head>
  <title>6513050067 - Monthon Mukkun</title>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
</head>
<body>
<div class="container">
  <h1 class="mb-4">OfficeManagement</h1>
  <ul class="nav nav-tabs">
    <li class="nav-item">
      <a class="nav-link" href="office-list">Show All Data</a>
    </li>
  </ul>

  <h3 class="mt-5">Update Office Data</h3>
  <form action="editData?officeCode=${office.officeCode}" method="POST">
    <div class="form-group">
      <label for="city">City:</label>
      <input type="text" class="form-control" name="city" value="${office.city}" >
    </div>
    <div class="form-group">
      <label for="phone">Phone:</label>

```

อธิบาย code คือได้ใช้ bootstrap ในการตกแต่ง website มีการสร้าง div กรอบ tag ทั้งหมดไว้มีการสร้าง tag <h1> กับ tag <a> เพื่อที่จะ link path ไป office-list ของ OfficeListServlet และมีการสร้าง tag <h3> มีข้อความ Update Office Data

พร้อมกับการสร้าง form ที่มี action ไปที่ editData ตามด้วย query string ของ officeCode ของ attribute object ที่ส่งมาใน servlet และ retrieve data ของ object attribute city มาใส่ไว้ใน input ที่ value office attribute city (คือการ show ข้อมูลเก่าของ object office)

และมีการ ใส่ attribute name = city เพื่อส่ง body parameter ไปให้ EditDataServlet

```

<div class="form-group">
  <label for="phone">Phone:</label>
  <input type="text" class="form-control" name="phone" value="${office.phone}" >
</div>
<div class="form-group">
  <label for="addressLine1">AddressLine1:</label>
  <input type="text" class="form-control" name="addressLine1" value="${office.addressLine1}">
</div>
<div class="form-group">
  <label for="country">Country:</label>
  <input type="text" class="form-control" name="country" value="${office.country}" >
</div>
<div class="form-group">
  <label for="postalCode">PostalCode:</label>
  <input type="text" class="form-control" name="postalCode" value="${office.postalCode}" >
</div>
<div class="form-group">
  <label for="territory">Territory:</label>
  <input type="text" class="form-control" name="territory" value="${office.territory}">
</div>
<input type="submit" class="btn btn-primary" value="UPDATE">
<input type="reset" class="btn btn-primary" value="RESET">
</form>
<hr>

```

จากนั้นก็ทำเหมือนเดิมกับด้านบนและมีปุ่ม submit และ reset (reset จะล้างข้อมูลที่เราพิมพ์ก่อนจะ update)

```

@WebServlet(name = "EditDataServlet", value = "/editData")
public class EditDataServlet extends HttpServlet {
    3 usages  carrynong2
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException {
        doPost(request, response);
    }

    2 usages  carrynong2
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException {
        HttpSession session = request.getSession(b: true);
        String officeCode = request.getParameter(s: "officeCode");
        String city = request.getParameter(s: "city");
        String phone = request.getParameter(s: "phone");
        String addressLine1 = request.getParameter(s: "addressLine1");
        String country = request.getParameter(s: "country");
        String postalCode = request.getParameter(s: "postalCode");
        String territory = request.getParameter(s: "territory");
        OfficeRepository officeRepository = new OfficeRepository();
    }

```

ดังนั้นใน EditDataServlet ผมก็ได้ทำ url mapping /editData มารับ request ไว้พร้อมกับทำ doPost() เพื่อรับ action method POST ที่ส่งมาซึ่งก็ได้มีการ สร้าง session เอาไว้เพื่อจะเก็บ session ของ offices

```

if (Utils.validateIsNotEmpty(officeCode, city, phone, addressLine1, country, postalCode, territory)
    && Utils.validateIsInteger(officeCode) && Utils.validateIsLetter(city) &&
    Utils.validateIsPhone(phone) && Utils.validateHaveSomeLetter(addressLine1) &&
    Utils.validateHaveSomeInteger(addressLine1) && Utils.validateIsLetter(country) &&
    !Utils.validateHaveSomeSpecial(postalCode) && Utils.validateIsLetter(territory)) {
    Office newOffice = new Office(officeCode, city, phone, addressLine1, country, postalCode, territory);
    boolean statusUpdate = officeRepository.update(newOffice);
    if (statusUpdate) {
        request.setAttribute(s: "messageUpdate", o: "Update office successfully");
        request.setAttribute(s: "statusUpdate", o: true);
    } else {
        request.setAttribute(s: "messageUpdate", o: "Update office not successfully");
        request.setAttribute(s: "statusUpdate", o: false);
    }
} else {
    request.setAttribute(s: "messageUpdate", o: "Update office not successfully");
    request.setAttribute(s: "statusUpdate", o: false);
}

session.setAttribute(s: "offices", officeRepository.findAll());
request.getRequestDispatcher(s: "/office-list.jsp").forward(request, response);

```

และก็ได้มีการ validate input แบบที่ทำเหมือนกับ AddOffice ซึ่งก็ได้ใช้ method update ของ officeRepository setAttribute 2 ตัวเหมือนเดิม พร้อมกับการ setAttribute offices (key) กับ officeRepository.findAll() List<Office> แบบ session และได้ forward ไปที่หน้า officec-list.jsp

Classic Model Offices ::

[Add](#)

OfficeCode	City	Phone	AddressLine1	Country	PostalCode	Territory	Actions
1	San Francisco	+1 650 219 4782	100 Market Street	USA	94080	NA	Edit Delete
2	Boston	+1 215 837 0825	1550 Court Place	USA	02107	NA	Edit Delete
3	USA	+1 212 555 3000	523 East 53rd Street	USA	10022	NA	Edit Delete
4	Paris	+33 14 723 4404	43 Rue Jouffroy D'abbans	France	75017	EMEA	Edit Delete
5	Tokyo	+81 33 224 5000	4-1 Kioicho	Japan	102-8578	Japan	Edit Delete
6	Sydney	+61 2 9264 2451	5-11 Wentworth Avenue	Australia	NSW 2010	APAC	Edit Delete
7	UK zaza	+44 20 7877 2041	25 Old Broad Street	UK	5555	EMEA	Edit Delete
8	Sanfransico	06390423895	120 Sukhumvit	Thailand	92170	NA	Edit Delete

Update office successfully ←

นี่คือตัวอย่างของการ update ข้อมูลลงใน database สำเร็จ

Classic Model Offices ::

[Add](#)

OfficeCode	City	Phone	AddressLine1	Country	PostalCode	Territory	Actions
1	San Francisco	+1 650 219 4782	100 Market Street	USA	94080	NA	Edit Delete
2	Boston	+1 215 837 0825	1550 Court Place	USA	02107	NA	Edit Delete
3	USA	+1 212 555 3000	523 East 53rd Street	USA	10022	NA	Edit Delete
4	Paris	+33 14 723 4404	43 Rue Jouffroy D'abbans	France	75017	EMEA	Edit Delete
5	Tokyo	+81 33 224 5000	4-1 Kioicho	Japan	102-8578	Japan	Edit Delete
6	Sydney	+61 2 9264 2451	5-11 Wentworth Avenue	Australia	NSW 2010	APAC	Edit Delete
7	UK zaza	+44 20 7877 2041	25 Old Broad Street	UK	5555	EMEA	Edit Delete
8	Sanfransico	06390423895	120 Sukhumvit	Thailand	92170	NA	Edit Delete

Update office not successfully ←

นี่คือตัวอย่างของการ update office ไม่สำเร็จ

```
<c:choose>
  <c:when test="{statusUpdate}">
    <h5 class="text-success">{messageUpdate}</h5>
  </c:when>
  <c:otherwise>
    <h5 class="text-danger">{messageUpdate}</h5>
  </c:otherwise>
</c:choose>
```

Code อธิบายคือก็เอา statusUpdate ที่ส่งมา check ถ้าเรา update Office สำเร็จก็แสดง messageUpdate แบบเขียว ใช้ class text-success ของ bootstrap ถ้าไม่สำเร็จก็ใช้ class text-danger ของ bootstrap

ตัวอย่างของ TestCase ที่ผ่าน (UPDATE) ได้

City: USA, Phone: 0639041212, AddressLine1: 120 Sukhumwit, Country: Thailand, PostalCode: 92170, Territory: NA

เพราะทุก field ผ่านเงื่อนไขที่ผม validate ไว้ทั้งหมด

ตัวอย่างของ TestCase ที่ไม่ผ่าน (UPDATE) ไม่ได้

City: USA, Phone: 0639041212za, AddressLine1: 120 Sukhumwit, Country: Thailand, PostalCode: 92170, Territory: NA

เพราะ field Phone ไม่เป็นตัวเลขทั้งหมด (มีตัวอักษรไม่ได้ มีอักขระพิเศษได้แค่บางตัว คือ +, -)

City: USA, Phone: 0639041212, AddressLine1: 120, Country: Thailand, PostalCode: 92170, Territory: NA

เพราะ field AddressLine1 เป็นตัวเลขทั้งหมด ไม่มีตัวตัวอักษรอย่างน้อย 1 ตัว

City: USA, Phone: 0639041212, AddressLine1: 120 Sukhumwit, Country: Thailand555, PostalCode: 92170, Territory: NA

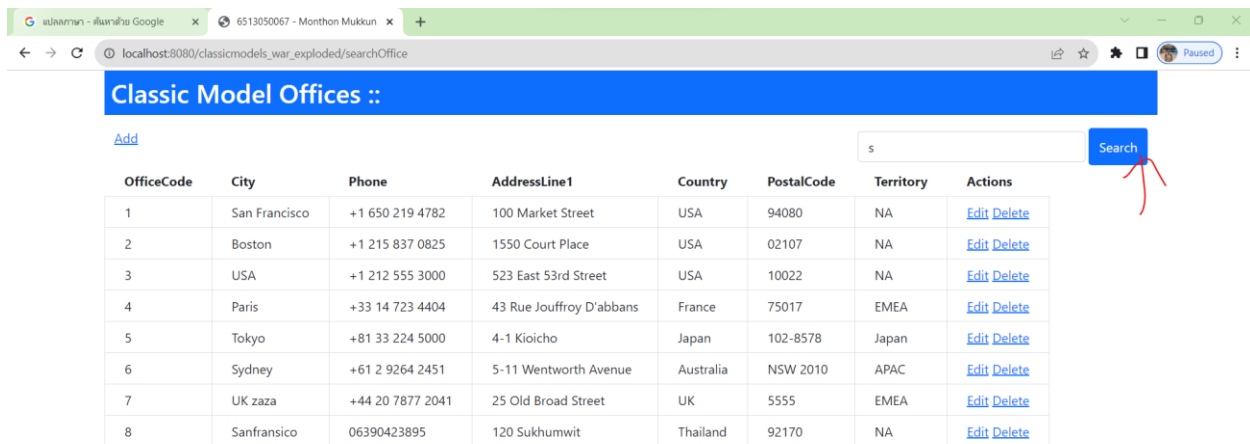
เพราะ field Country ไม่เป็นตัวอักษรทั้งหมด มีตัวเลขอย่างน้อย 1 ตัวก็ไม่ได้

City: USA, Phone: 0639041212, AddressLine1: 120 Sukhumwit, Country: Thailand, PostalCode: 92170@@@, Territory: NA

เพราะ field PostalCode มีตัวอักขระพิเศษอย่างน้อย 1 ตัว

City: USA, Phone: 0639041212, AddressLine1: 120 Sukhumwit, Country: Thailand, PostalCode: 92170, Territory: NA555

เพราะ field Territory ไม่เป็นตัวอักษรทั้งหมด มีตัวเลขอย่างน้อย 1 ตัวก็ไม่ได้



Option ในการ search ก็จะทำการ searchByCityOrCountry

```
<div class="col-md-6 d-flex justify-content-end w-50">
  <form action="searchOffice" method="post">
    <input name="searchByCityOrCountry" style="..." type="text" placeholder="Search By C
    <button type="submit" class="btn btn-primary">Search</button>
  </form>
</div>
iv>
```

เมื่อผมทำการพิมพ์ input แล้วทำการ click ก็จะไปทำ action seachOffice โดยใช้ method POST พร้อมกับส่ง body ไปที่มี attribute name searchByCityOrCountry

```
@WebServlet(name = "SearchOfficeServlet", value = "/searchOffice")
public class SearchOfficeServlet extends HttpServlet {
    3 usages
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException {
        doPost(request, response);
    }

    2 usages
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException {
        String searchByCityOrCountry = request.getParameter("searchByCityOrCountry");
        HttpSession session = request.getSession(true);
        if (searchByCityOrCountry != null) {
            OfficeRepository officeRepository = new OfficeRepository();
            List<Office> offices = officeRepository.findByCityOrCountry(searchByCityOrCountry);
            session.setAttribute("offices", offices);
        }
        request.getServletContext().getRequestDispatcher("/office-list.jsp").forward(request, response);
    }
}
```

ดังนั้นใน SearchOfficeServlet ผมจึงทำ url mapping ที่ชื่อ /searchOffice ซึ่งทำ method doPost() พร้อมกับ getParameter searchByCityOrCountry แล้วได้มีการสร้าง OfficeRepository กับการใช้ method findByCityOrCountry ซึ่งจะ return List<Office> ของ city หรือ country ที่ขึ้นต้นด้วยตัวอักษรนั้น และได้มีการ setAttribute offices (key) กับ List<Office> (value) และ forward ไปที่หน้า office-list.jsp

OfficeCode	City	Phone	AddressLine1	Country	PostalCode	Territory	Actions
1	San Francisco	+1 650 219 4782	100 Market Street	USA	94080	NA	Edit Delete
6	Sydney	+61 2 9264 2451	5-11 Wentworth Avenue	Australia	NSW 2010	APAC	Edit Delete
8	Sanfransico	06390423895	120 Sukhumwit	Thailand	92170	NA	Edit Delete

นี่คือตัวอย่างของการ search By City or Country ที่ขึ้นต้นด้วยตัวอักษร s ซึ่งก็จะ loop offices ที่มี office ตามที่ได้ setAttribute มา

OfficeCode	City	Phone	AddressLine1	Country	PostalCode	Territory	Actions
1	San Francisco	+1 650 219 4782	100 Market Street	USA	94080	NA	Edit Delete
2	Boston	+1 215 837 0825	1550 Court Place	USA	02107	NA	Edit Delete
3	USA	+1 212 555 3000	523 East 53rd Street	USA	10022	NA	Edit Delete
4	Paris	+33 14 723 4404	43 Rue Jouffroy D'abbans	France	75017	EMEA	Edit Delete
5	Tokyo	+81 33 224 5000	4-1 Kioicho	Japan	102-8578	Japan	Edit Delete
6	Sydney	+61 2 9264 2451	5-11 Wentworth Avenue	Australia	NSW 2010	APAC	Edit Delete
7	UK zaza	+44 20 7877 2041	25 Old Broad Street	UK	5555	EMEA	Edit Delete
8	Sanfransico	06390423895	120 Sukhumwit	Thailand	92170	NA	Edit Delete

มาดูในส่วนสุดท้ายคือ delete ซึ่งถ้าผม click ที่ปุ่ม delete ก็จะ delete object ของ office นั้นตาม officeCode

```
<td class="border px-4 py-2">${office.territory}</td>
<td class="border px-4 py-2">
  <a href="editOffice?officeCode=${office.officeCode}" class="text-blue-500">Edit</a>
  <a href="deleteOffice?officeCode=${office.officeCode}" class="text-red-500 ml-2">Delete</a>
</td>
```

อธิบาย code คือผมได้ทำการสร้าง tag <a> มี link path ไปที่ deleteOffice และตาม query string officeCode ของ object ที่แสดงตามนั้น ui


```

@WebServlet(name = "DeleteOfficeServlet", value = "/deleteOffice")
public class DeleteOfficeServlet extends HttpServlet {
    3 usages carrynong2 *
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException {
        HttpSession session = request.getSession(b: true);
        String officeCode = request.getParameter(s: "officeCode");
        OfficeRepository officeRepository = new OfficeRepository();
        if (officeCode != null) {
            try {
                boolean statusDelete = officeRepository.delete(officeCode);
                if (statusDelete) {
                    request.setAttribute(s: "messageDelete", o: "Delete office successfully");
                    request.setAttribute(s: "statusDelete", o: true);
                } else {
                    request.setAttribute(s: "messageDelete", o: "Delete office not successfully");
                    request.setAttribute(s: "statusDelete", o: false);
                }
            } catch (Exception e) {
                request.getServletContext().getRequestDispatcher(s: "/error.jsp").forward(request, response);
            }
        } else {
            request.setAttribute(s: "messageDelete", o: "Delete office not successfully");
            request.setAttribute(s: "statusDelete", o: false);
        }
    }
}

```

ใน DeleteOfficeServlet ก็ได้ทำการ สร้าง url mapping ที่ชื่อ /deleteOffice ตามที่ request ส่งมาซึ่งในตอนแรกก็ได้ทำการสร้าง session ไว้พร้อมกับดึง officeCode parameter ที่ส่งมาจากนั้นก็มีการ check ว่าที่ส่งมาไม่ใช่ null

ในส่วนของก่อนที่จะลบมีการใช้ try catch เพราะว่าใน db Office จะมีข้อมูลตั้งต้นไว้ 7 ตัวซึ่งจะไปติด foreign key ของ Employee ดังนั้นเมื่อลบ officeCode ที่ 1-7 thrown exception ออกมาผมเลยจัดการด้วยการ forward ไปที่หน้า error.jsp เพื่อแสดงว่าไม่สามารถลบได้ กลับกันถ้าเป็น case ที่สามารถลบแล้วไม่ thrown ออกมาก็จะมี 2 case คือลบได้กับลบไม่ได้ จากนั้นก็ setAttribute แยก case แบบเดิม

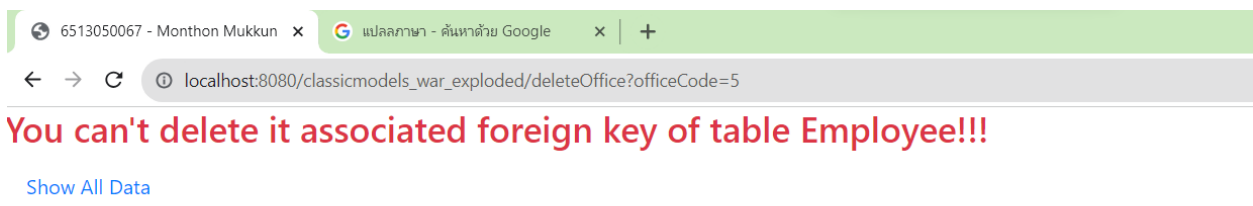
```

        request.getServletContext().getRequestDispatcher(s: "/error.jsp").forward(request, response);
    }
} else {
    request.setAttribute(s: "messageDelete", o: "Delete office not successfully");
    request.setAttribute(s: "statusDelete", o: false);
}
session.setAttribute(s: "offices", officeRepository.findAll());
request.getServletContext().getRequestDispatcher(s: "/office-list.jsp").forward(request, response);
}

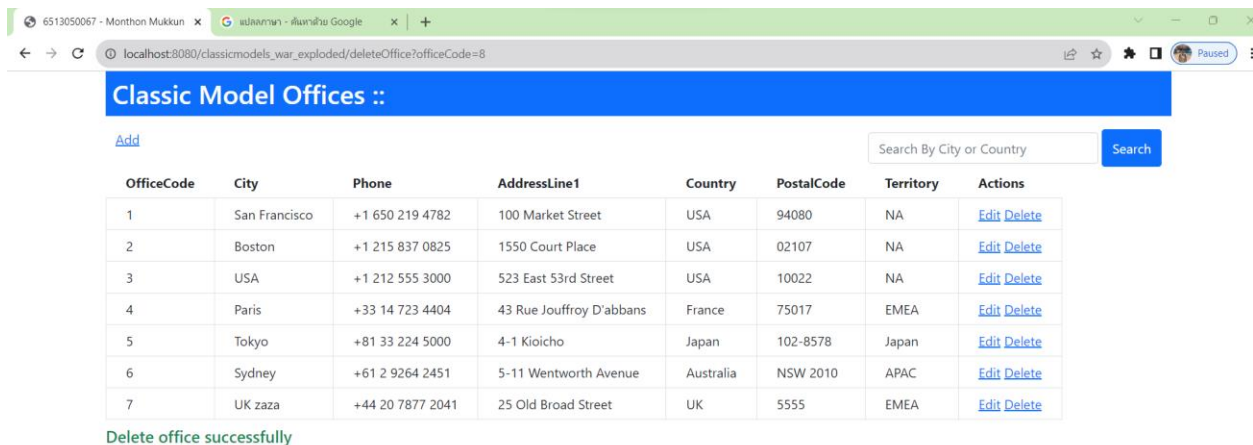
2 usages carrynong2
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException {
    doGet(request, response);
}

```

และก็ได้มีการ setAttribute offices (key) กับ List<Office> (value) ซึ่งใช้ method findAll() ของ officeRepository เหมือนเดิม



นี่คือ case เมื่อลบ officeCode 1 – 7



ส่วนนี้คือ case ที่ลบ officeCode ที่ 8+ ขึ้นไป