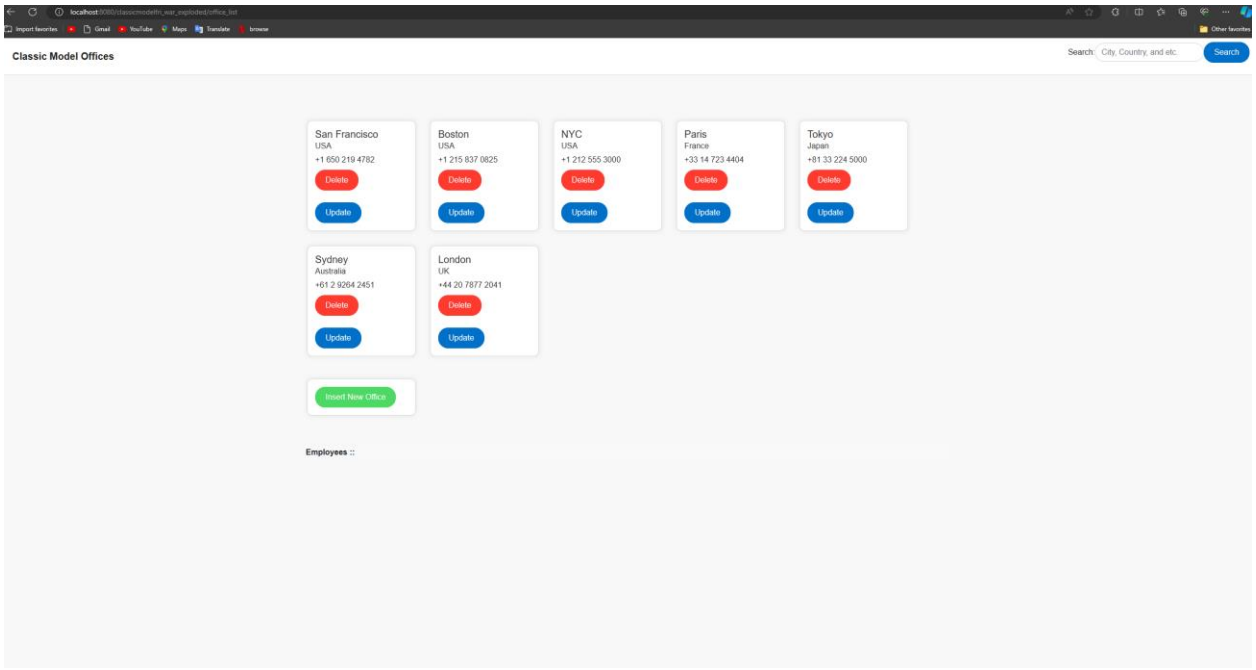
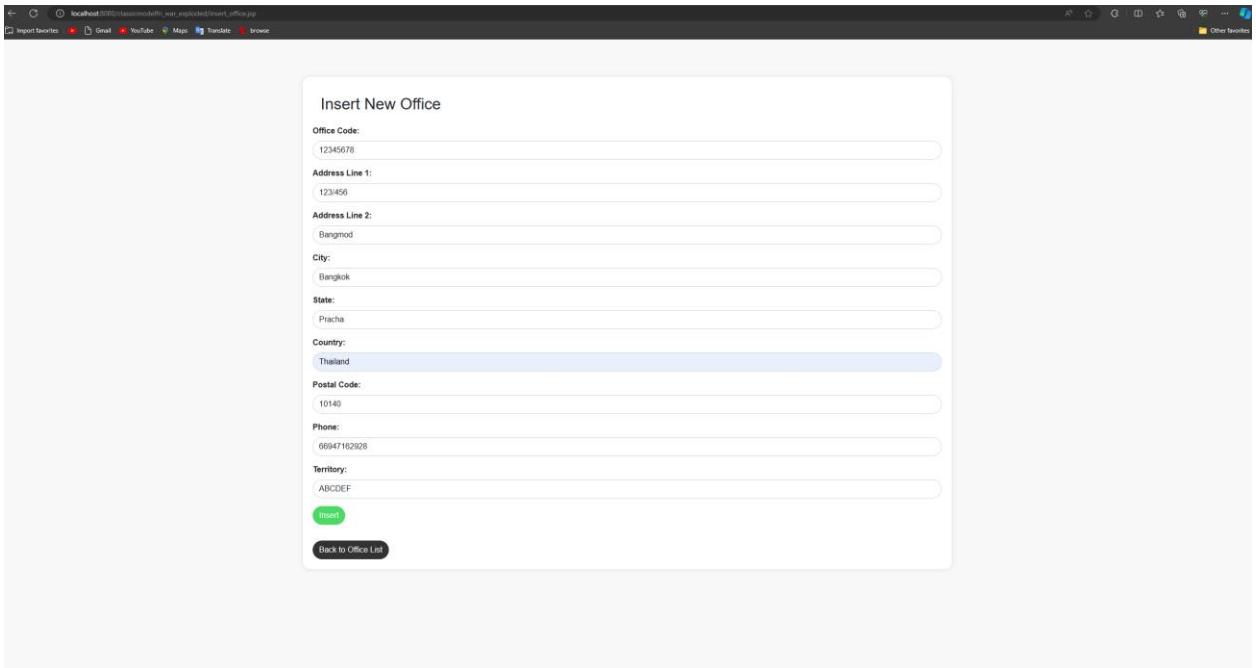
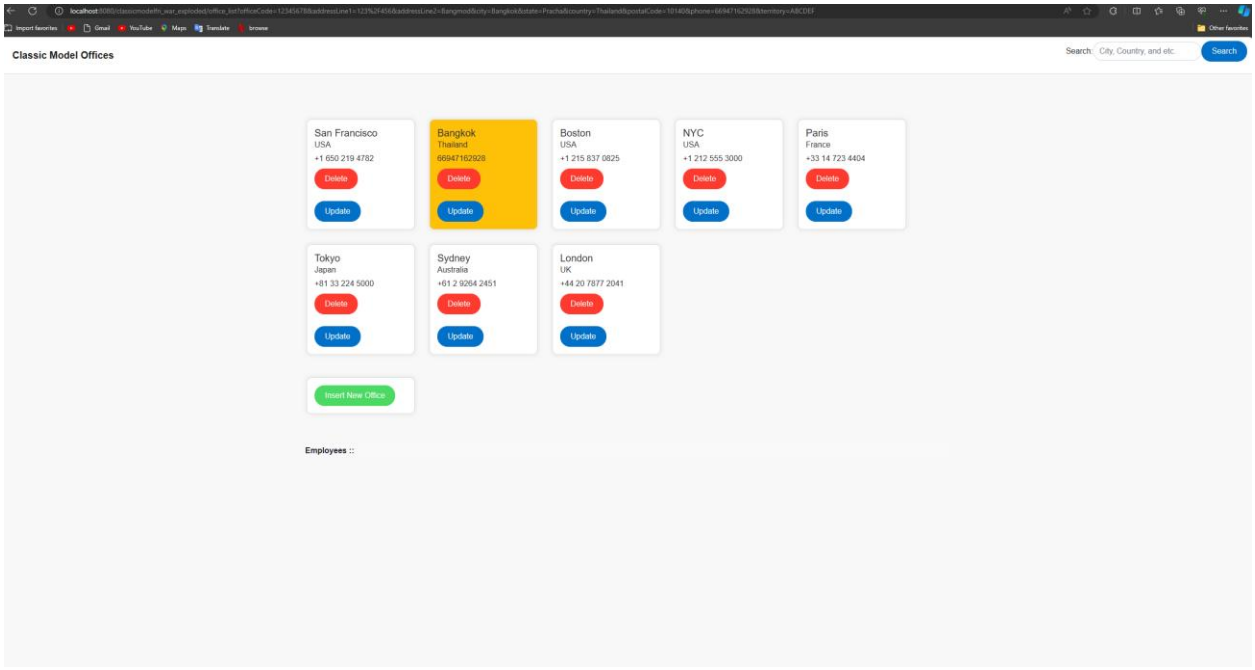
INSERT


Classic Model Offices
Search: City, Country, and etc.
Search
San Francisco
USA
+1 650 219 4782
Delete
Update
Boston
USA
+1 215 837 0825
Delete
Update
NYC
USA
+1 212 555 3000
Delete
Update
Paris
France
+33 14 723 4404
Delete
Update
Tokyo
Japan
+81 33 224 5000
Delete
Update
Sydney
Australia
+61 2 9264 2451
Delete
Update
London
UK
+44 20 7877 2041
Delete
Update
Insert New Office
Employees ::

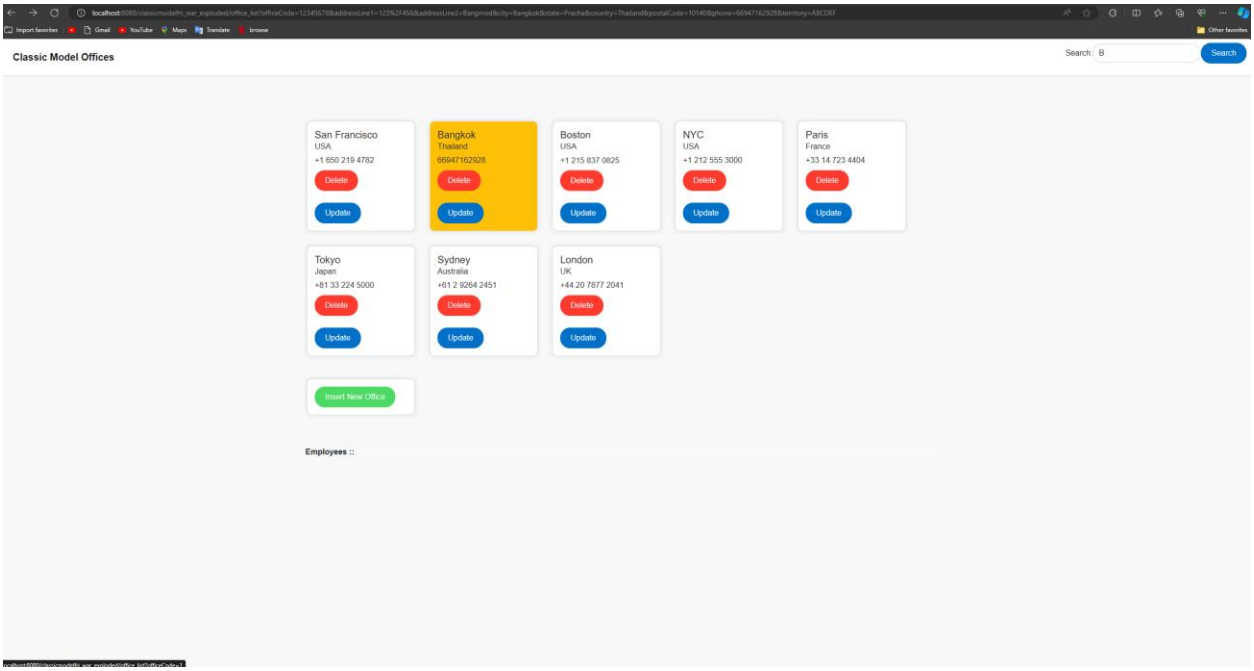

Insert New Office
Office Code:
12345678
Address Line 1:
123/456
Address Line 2:
Bangmod
City:
Bangkok
State:
Pracha
Country:
Thailand
Postal Code:
10140
Phone:
66947162928
Territory:
ABCDEF
Insert
Back to Office List

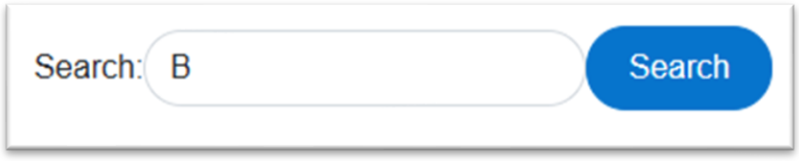


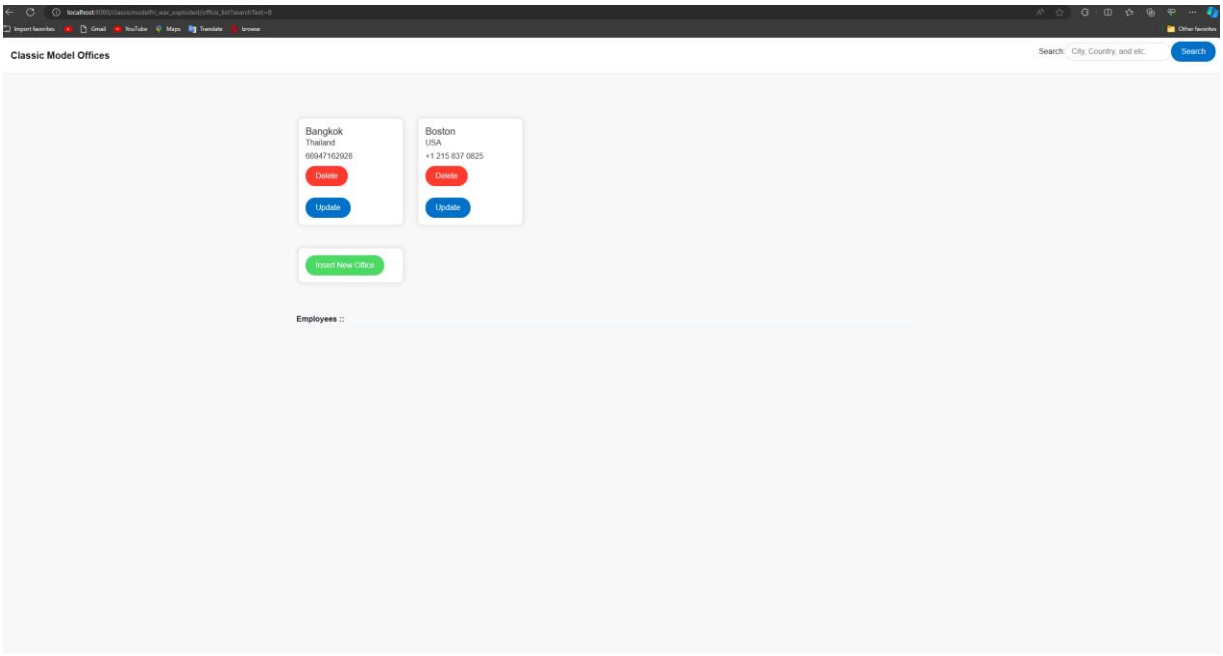

```java
import java.io.IOException;

@WebServlet("/insertOffice")
public class InsertOfficeServlet extends HttpServlet {
    1 usage
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        String officeCode = request.getParameter( name: "officeCode");
        String addressLine1 = request.getParameter( name: "addressLine1");
        String addressLine2 = request.getParameter( name: "addressLine2");
        String city = request.getParameter( name: "city");
        String state = request.getParameter( name: "state");
        String country = request.getParameter( name: "country");
        String postalCode = request.getParameter( name: "postalCode");
        String phone = request.getParameter( name: "phone");
        String territory = request.getParameter( name: "territory");

        Office newOffice = new Office();
        newOffice.setOfficeCode(officeCode);
        newOffice.setAddressLine1(addressLine1);
        newOffice.setAddressLine2(addressLine2);
        newOffice.setCity(city);
        newOffice.setState(state);
        newOffice.setCountry(country);
        newOffice.setPostalCode(postalCode);
        newOffice.setPhone(phone);
        newOffice.setTerritory(territory);


        OfficeRepository officeRepository = new OfficeRepository();
        boolean success = officeRepository.insert(newOffice);


        response.sendRedirect( location: request.getContextPath() + "/office_list.jsp");
    }
}
```

- InsertOfficeServlet
    - Servlet นี้ถูกเรียกจากหน้า JSP ที่ให้ผู้ใช้กรอกข้อมูลofficeใหม่.
    - รับข้อมูลจาก request parameters เช่น officeCode, addressLine1, addressLine2, city, และอื่นๆ.
    - สร้างอ็อบเจ็กต์ Office ใหม่และกำหนดค่าข้อมูลที่ได้รับจาก request parameters.
    - เรียกใช้ OfficeRepository เพื่อ insert ข้อมูลofficeใหม่.
    - Redirect ผู้ใช้กลับไปที่หน้า JSP ที่แสดงรายการofficeหลังจากทำการ insert.
- insert_office.jsp (หน้า JSP สำหรับกรอกข้อมูลofficeใหม่)
    - มีฟอร์มที่ผู้ใช้กรอกข้อมูลoffice เช่น officeCode, addressLine1, addressLine2, city, และอื่น ๆ.
    - เมื่อผู้ใช้กดปุ่ม "Insert" จะทำการ submit ข้อมูลไปยัง InsertOfficeServlet.
- Flow ในการ Insert Office:
    - ผู้ใช้เข้าถึงหน้า JSP (insert_office.jsp) และกรอกข้อมูล officeที่ต้องการเพิ่ม.
    - เมื่อกดปุ่ม "Insert", ข้อมูลจะถูก submit ไปยัง Servlet (InsertOfficeServlet).
    - Servlet จะรับข้อมูลจาก request parameters และสร้างอ็อบเจ็กต์ Office ใหม่.
    - Servlet เรียกใช้ OfficeRepository เพื่อ insert ข้อมูลในฐานข้อมูล.
    - หลังจากที่ insert เสร็จสิ้น, Servlet จะ redirect ผู้ใช้กลับไปที่หน้า JSP ที่แสดงรายการoffice.

Search


Classic Model Offices
Search: B
Search
San Francisco
USA
+1 650 219 4782
Delete
Update
Bangkok
Thailand
66947162928
Delete
Update
Boston
USA
+1 215 837 0825
Delete
Update
NYC
USA
+1 212 555 3000
Delete
Update
Paris
France
+33 14 723 4404
Delete
Update
Tokyo
Japan
+81 33 224 5000
Delete
Update
Sydney
Australia
+61 2 9264 2451
Delete
Update
London
UK
+44 20 7877 2041
Delete
Update
Insert New Office
Employees ::

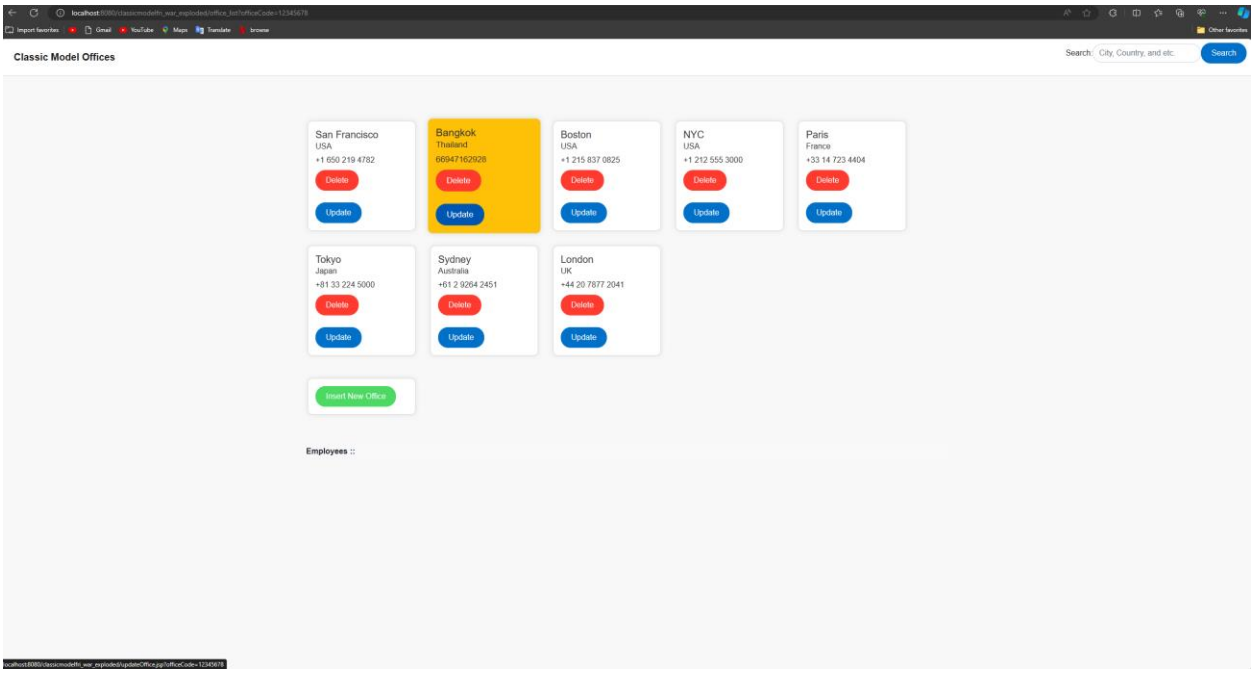

Search:
B
Search


Classic Model Offices
Search: City, Country, and etc.
Search
Bangkok
Thailand
66947162928
Delete
Update
Boston
USA
+1 215 837 0825
Delete
Update
Insert New Office
Employees ::

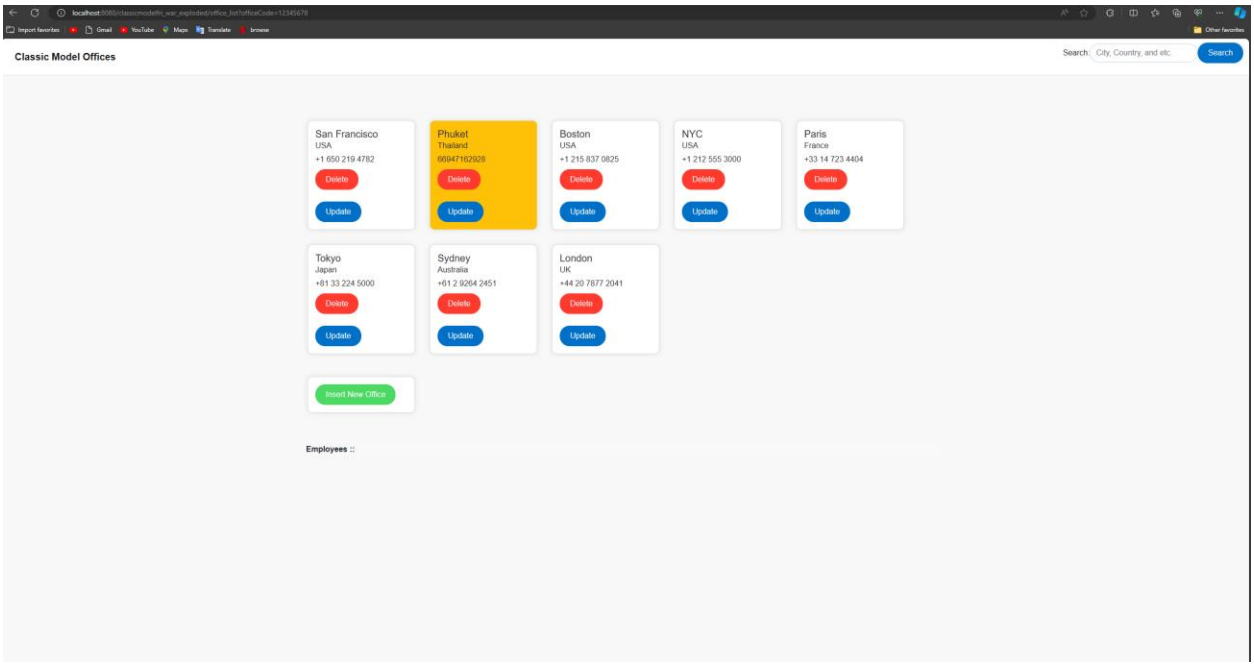
- OfficeListServlet
    - Servlet นี้ถูกเรียก: เมื่อผู้ใช้เข้าถึงหน้ารายการ office (/office_list) หรือทำการค้นหา (Search).
    - ขั้นตอนการทำงาน:
    - ตรวจสอบ Parameter: ตรวจสอบว่ามีการส่งพารามิเตอร์ searchText มาหรือไม่.
    - เรียกใช้ Repository: ถ้ามี searchText, จะเรียกใช้ OfficeRepository เพื่อค้นหา office โดยใช้เงื่อนไขที่ระบุใน SearchByCityOrCountry.
    - Set Attribute: นำผลลัพธ์มาเก็บใน attribute ของ request เพื่อนำไปแสดงในหน้า JSP.

- OfficeRepository
    - SearchByCityOrCountry: ในคลาสนี้มีเมธอดที่ใช้ Query ข้อมูล office โดยใช้เงื่อนไข Office.FIND_BY_CITY_OR_COUNTRY.
    - ใช้ Named Query ที่ถูกกำหนดใน Entity Class (Office) เพื่อค้นหา office ที่มี city หรือ country ที่ตรงกับ searchText.
- หน้า JSP (office_list.jsp)
    - แสดงผล: แสดงรายการ office ที่ได้รับมาจาก Servlet.
    - ฟอร์มค้นหา: มีฟอร์มที่ให้ผู้ใช้กรอกข้อความเพื่อค้นหา (City, Country, และอื่น ๆ).
    - เมื่อผู้ใช้กดปุ่ม "Search", ข้อมูลจะถูกส่งไปที่ OfficeListServlet.
- Flow การทำงานทั้งหมด:
    - ผู้ใช้ทำการกรอกข้อมูลค้นหา (Search) หรือเข้าถึงหน้ารายการ office
    - Servlet (OfficeListServlet) ตรวจสอบ Parameter:
    - ถ้ามี searchText, ก็จะทำการค้นหาด้วย OfficeRepository และนำผลลัพธ์ไปแสดง.
    - ถ้าไม่มี searchText, ก็จะแสดงรายการ office ทั้งหมด.
    - ผลลัพธ์ถูกแสดงในหน้า JSP (office_list.jsp):
    - ถ้ามีผลลัพธ์จากการค้นหา, รายการ office ที่เกี่ยวข้องจะถูกแสดง.
    - หากไม่มีการค้นหา, รายการ office ทั้งหมดจะถูกแสดง.
    - ผู้ใช้สามารถทำการค้นหาหรือดูรายการ office ทั้งหมดตามต้องการ.

Update


Classic Model Offices
Search:
City, Country, and etc.
Search
San Francisco
USA
+1 650 219 4782
Delete
Update
Bangkok
Thailand
66947162928
Delete
Update
Boston
USA
+1 215 837 0825
Delete
Update
NYC
USA
+1 212 555 3000
Delete
Update
Paris
France
+33 14 723 4404
Delete
Update
Tokyo
Japan
+81 33 224 5000
Delete
Update
Sydney
Australia
+61 2 9264 2451
Delete
Update
London
UK
+44 20 7877 2041
Delete
Update
Insert New Office
Employees ::


Update Office
Office Code:
12345678
Address Line 1:
123/456
Address Line 2:
Bangmod
City:
Phuket
State:
Pracha
Country:
Thailand
Postal Code:
55555
Phone:
66947162928
Territory:
ABCDEF
Update
Back to Office List

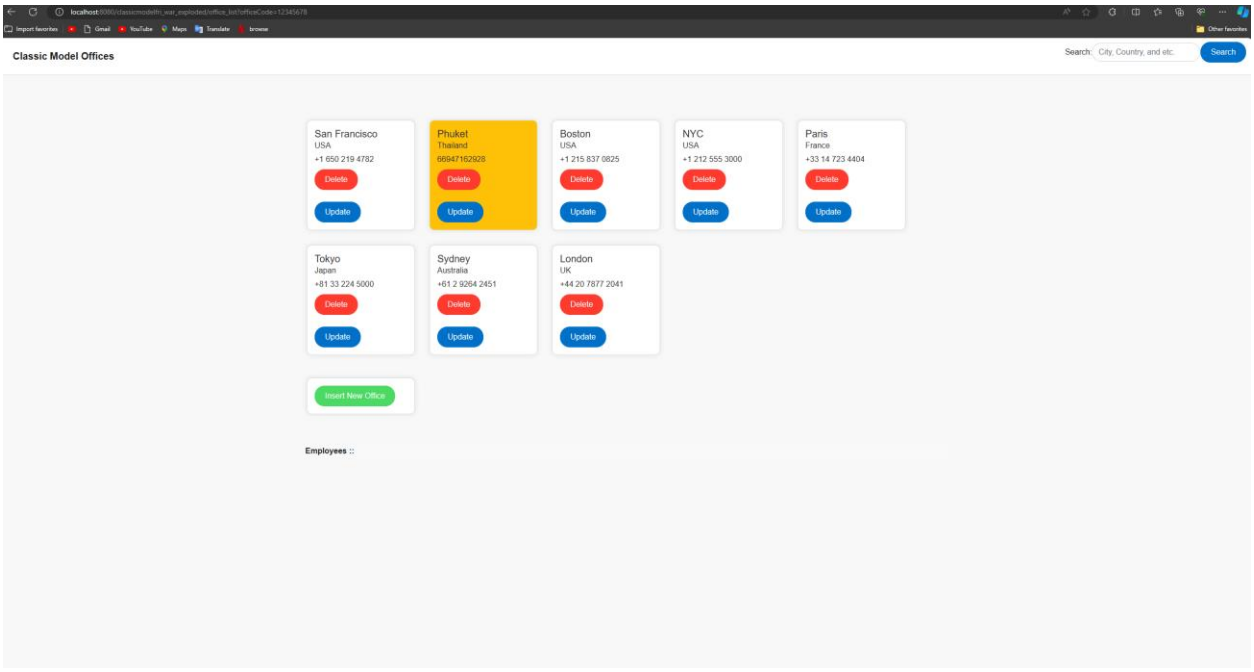


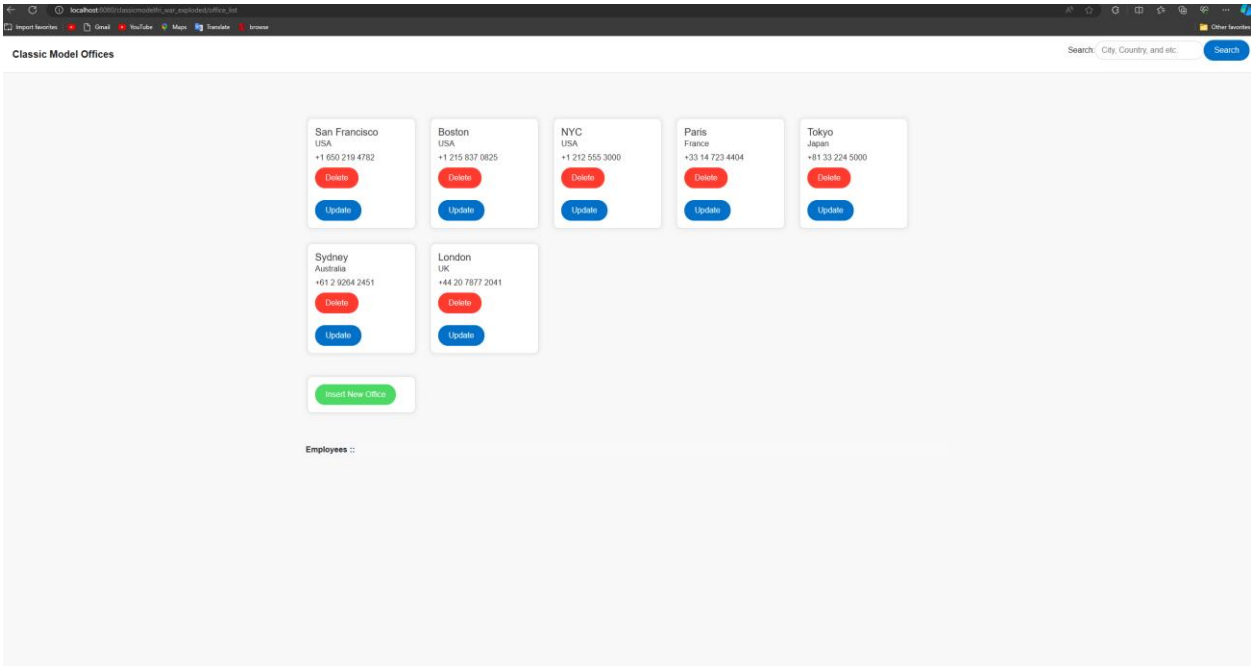

```java
@WebServlet(name = "UpdateOfficeServlet", value = "/UpdateOffice")
public class UpdateOfficeServlet extends HttpServlet {
    1 usage
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        String officeCode = request.getParameter( name: "officeCode");
        String addressLine1 = request.getParameter( name: "addressLine1");
        String addressLine2 = request.getParameter( name: "addressLine2");
        String city = request.getParameter( name: "city");
        String state = request.getParameter( name: "state");
        String country = request.getParameter( name: "country");
        String postalCode = request.getParameter( name: "postalCode");
        String phone = request.getParameter( name: "phone");
        String territory = request.getParameter( name: "territory");


        Office updatedOffice = new Office();
        updatedOffice.setOfficeCode(officeCode);
        updatedOffice.setAddressLine1(addressLine1);
        updatedOffice.setAddressLine2(addressLine2);
        updatedOffice.setCity(city);
        updatedOffice.setState(state);
        updatedOffice.setCountry(country);
        updatedOffice.setPostalCode(postalCode);
        updatedOffice.setPhone(phone);
        updatedOffice.setTerritory(territory);


        OfficeRepository officeRepository = new OfficeRepository();
        boolean success = officeRepository.update(updatedOffice);

        if (success) {

            response.sendRedirect( location: request.getContextPath() + "/office_list?officeCode=" + officeCode);
        } else {

            response.sendRedirect( location: request.getContextPath() + "/error.jsp");
        }
    }
}
```

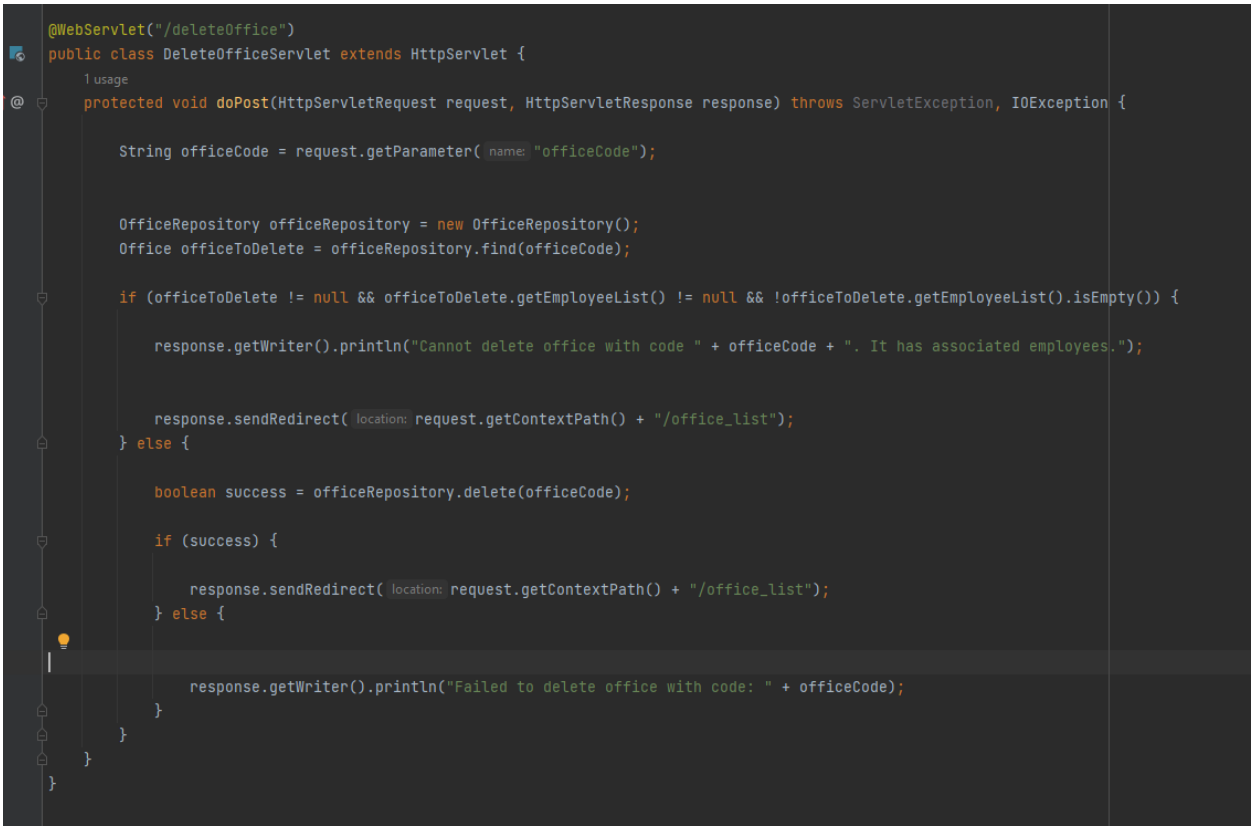
- OfficeEditServlet (Servlet สำหรับแก้ไขข้อมูลoffice)
    - Servlet นี้ถูกเรียก: เมื่อผู้ใช้กดปุ่ม "Edit" บนหน้ารายการ office (/office_edit) เพื่อแก้ไขข้อมูล office ที่มี id ที่ระบุ.
    - ขั้นตอนการทำงาน:
    - รับพารามิเตอร์: รับ id ของ office ที่ต้องการแก้ไขผ่าน URL Parameter.
    - โหลดข้อมูล: เรียกใช้ OfficeRepository เพื่อโหลดข้อมูล office ที่ต้องการแก้ไข.
    - Set Attribute: นำข้อมูล office ที่โหลดมาไว้ใน attribute ของ request เพื่อนำไปแสดงในหน้า JSP.
- OfficeUpdateServlet (Servlet สำหรับบันทึกการอัปเดตข้อมูล)
    - Servlet นี้ถูกเรียก: เมื่อผู้ใช้กดปุ่ม "Save" หลังจากแก้ไขข้อมูล office (/office_update).
    - ขั้นตอนการทำงาน:
    - รับข้อมูลที่แก้ไข: รับข้อมูลที่ผู้ใช้แก้ไขผ่าน parameter หรือ form data.
    - ตรวจสอบข้อมูล: ตรวจสอบความถูกต้องของข้อมูลที่รับมา.
    - อัปเดตข้อมูล: เรียกใช้ OfficeRepository เพื่อทำการอัปเดตข้อมูล office
    - Redirect: หลังจากที่ทำการอัปเดตเรียบร้อย, ทำการ redirect กลับไปที่หน้ารายละเอียดของ office ที่ถูกแก้ไข.
- OfficeRepository (คลาสที่ทำการจัดการกับข้อมูล office ในฐานข้อมูล)
    - updateOffice: เมธอดที่ใช้ในการอัปเดตข้อมูล office.
    - รับข้อมูล office ต้องการอัปเดต.
    - ใช้ EntityManager ใน JPA เพื่อทำการ merge ข้อมูล office
- หน้า JSP (office_edit.jsp)
    - แสดงฟอร์มแก้ไข: แสดงฟอร์มที่ผู้ใช้สามารถแก้ไขข้อมูล office ได้.
    - ปุ่ม Save: เมื่อกดปุ่ม "Save", ข้อมูลจะถูกส่งไปที่ OfficeUpdateServlet เพื่อทำการอัปเดต.

- Flow การทำงานทั้งหมด:
    - ผู้ใช้คลิกที่ปุ่ม "Edit" บนหน้ารายการ office
    - Servlet (OfficeEditServlet) รับ id ของ office ที่ต้องการแก้ไข.
    - โหลดข้อมูล office นี้จากฐานข้อมูล.
    - Set Attribute ใน request เพื่อนำไปแสดงในหน้า JSP (office_edit.jsp).
    - ผู้ใช้แก้ไขข้อมูลในหน้า JSP (office_edit.jsp).
    - ผู้ใช้กดปุ่ม "Save".
    - Servlet (OfficeUpdateServlet) รับข้อมูลที่แก้ไข.
    - ตรวจสอบความถูกต้องของข้อมูล.
    - เรียกใช้ OfficeRepository เพื่อทำการอัปเดตข้อมูล office
    - อัปเดตเรียบร้อย:
    - Redirect ไปที่หน้ารายละเอียดของ office ที่ถูกแก้ไข.
    - ผู้ใช้สามารถแก้ไขข้อมูล office และบันทึกการเปลี่ยนแปลงได้ตามกระบวนการข้างต้น.

Delete


Classic Model Offices
Search:
City, Country, and etc.
Search
San Francisco
USA
+1 650 219 4782
Delete
Update
Phuket
Thailand
66947162928
Delete
Update
Boston
USA
+1 215 837 0825
Delete
Update
NYC
USA
+1 212 555 3000
Delete
Update
Paris
France
+33 14 723 4404
Delete
Update
Tokyo
Japan
+81 33 224 5000
Delete
Update
Sydney
Australia
+61 2 9264 2451
Delete
Update
London
UK
+44 20 7877 2041
Delete
Update
Insert New Office
Employees ::


Classic Model Offices
Search:
City, Country, and etc.
Search
San Francisco
USA
+1 650 219 4782
Delete
Update
Boston
USA
+1 215 837 0825
Delete
Update
NYC
USA
+1 212 555 3000
Delete
Update
Paris
France
+33 14 723 4404
Delete
Update
Tokyo
Japan
+81 33 224 5000
Delete
Update
Sydney
Australia
+61 2 9264 2451
Delete
Update
London
UK
+44 20 7877 2041
Delete
Update
Insert New Office
Employees ::

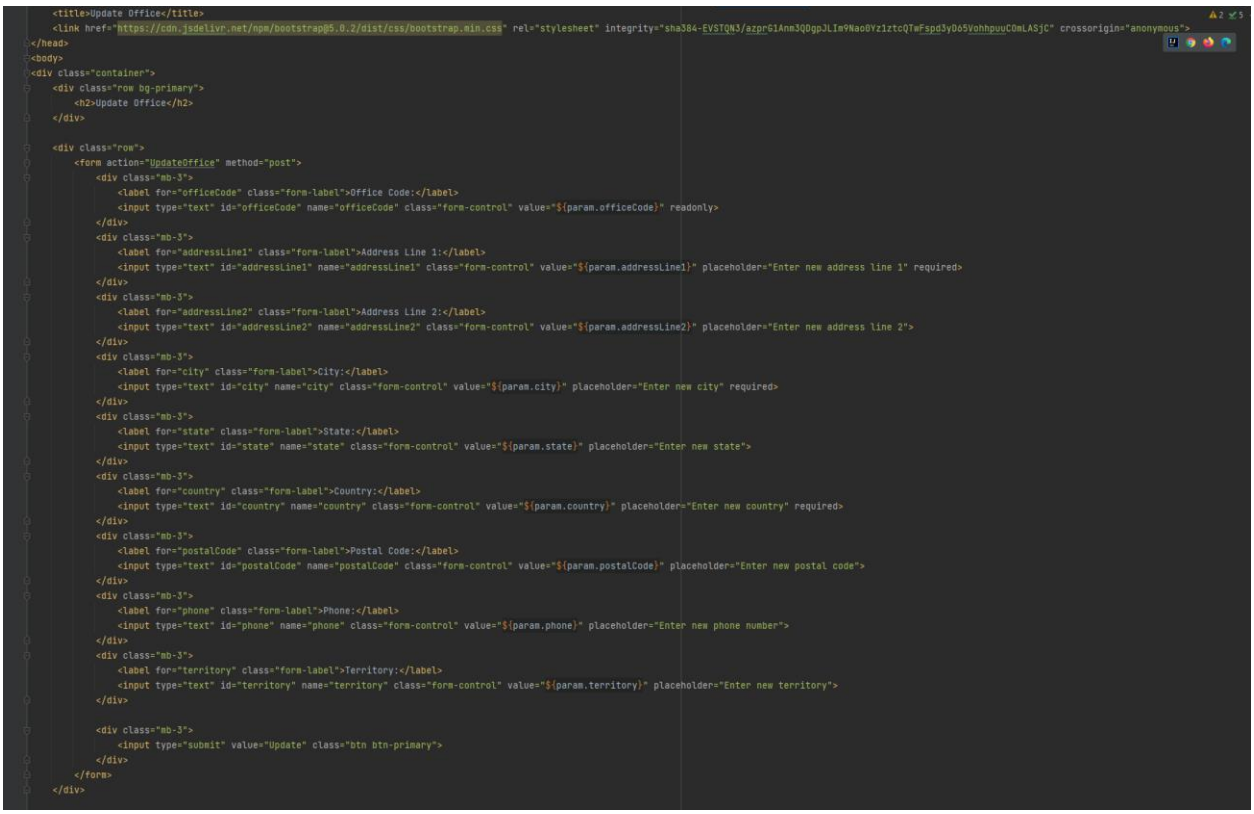


```java
@WebServlet("/deleteOffice")
public class DeleteOfficeServlet extends HttpServlet {
    1 usage
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        String officeCode = request.getParameter( name: "officeCode");


        OfficeRepository officeRepository = new OfficeRepository();
        Office officeToDelete = officeRepository.find(officeCode);

        if (officeToDelete != null && officeToDelete.getEmployeeList() != null && !officeToDelete.getEmployeeList().isEmpty()) {

            response.getWriter().println("Cannot delete office with code " + officeCode + ". It has associated employees.");


            response.sendRedirect( location: request.getContextPath() + "/office_list");
        } else {

            boolean success = officeRepository.delete(officeCode);

            if (success) {

                response.sendRedirect( location: request.getContextPath() + "/office_list");
            } else {


                response.getWriter().println("Failed to delete office with code: " + officeCode);
            }
        }
    }
}
```

- OfficeDeleteServlet (Servlet สำหรับลบข้อมูล office)
    - Servlet นี้ถูกเรียก: เมื่อผู้ใช้กดปุ่ม "Delete" บนหน้ารายละเอียดของ office (/office_delete) เพื่อลบข้อมูล office ที่มี id ที่ระบุ.
    - ขั้นตอนการทำงาน:
    - รับพารามิเตอร์: รับ id ของ office ที่ต้องการลบผ่าน URL Parameter.
    - ลบข้อมูล: เรียกใช้ OfficeRepository เพื่อทำการลบข้อมูล office
    - Redirect: หลังจากที่ทำการลบเรียบร้อย, ทำการ redirect กลับไปที่หน้ารายการ office
- OfficeRepository (คลาสที่ทำการจัดการกับข้อมูล office ในฐานข้อมูล)
    - deleteOffice: เมธอดที่ใช้ในการลบข้อมูล office
    - รับ id ของ office ที่ต้องการลบ.
    - ใช้ EntityManager ใน JPA เพื่อทำการลบข้อมูล.
- หน้า JSP (office_detail.jsp)
    - ปุ่ม Delete: แสดงปุ่ม "Delete" ที่เมื่อผู้ใช้กด, จะทำการส่ง request ไปยัง OfficeDeleteServlet เพื่อทำการลบข้อมูล.

- Flow การทำงานทั้งหมด:
    - ผู้ใช้เข้าสู่หน้ารายละเอียดของ office
    - แสดงข้อมูล office และปุ่ม "Delete".
    - ผู้ใช้คลิกที่ปุ่ม "Delete".
    - Servlet (OfficeDeleteServlet) รับ id ของ office ที่ต้องการลบ.
    - เรียกใช้ OfficeRepository เพื่อทำการลบข้อมูล office
    - ลบเรียบร้อย:
    - Redirect กลับไปที่หน้ารายการ office หลังจากที่ทำการลบข้อมูล.
    - ผู้ใช้สามารถลบข้อมูล office ได้ตามกระบวนการข้างต้น.

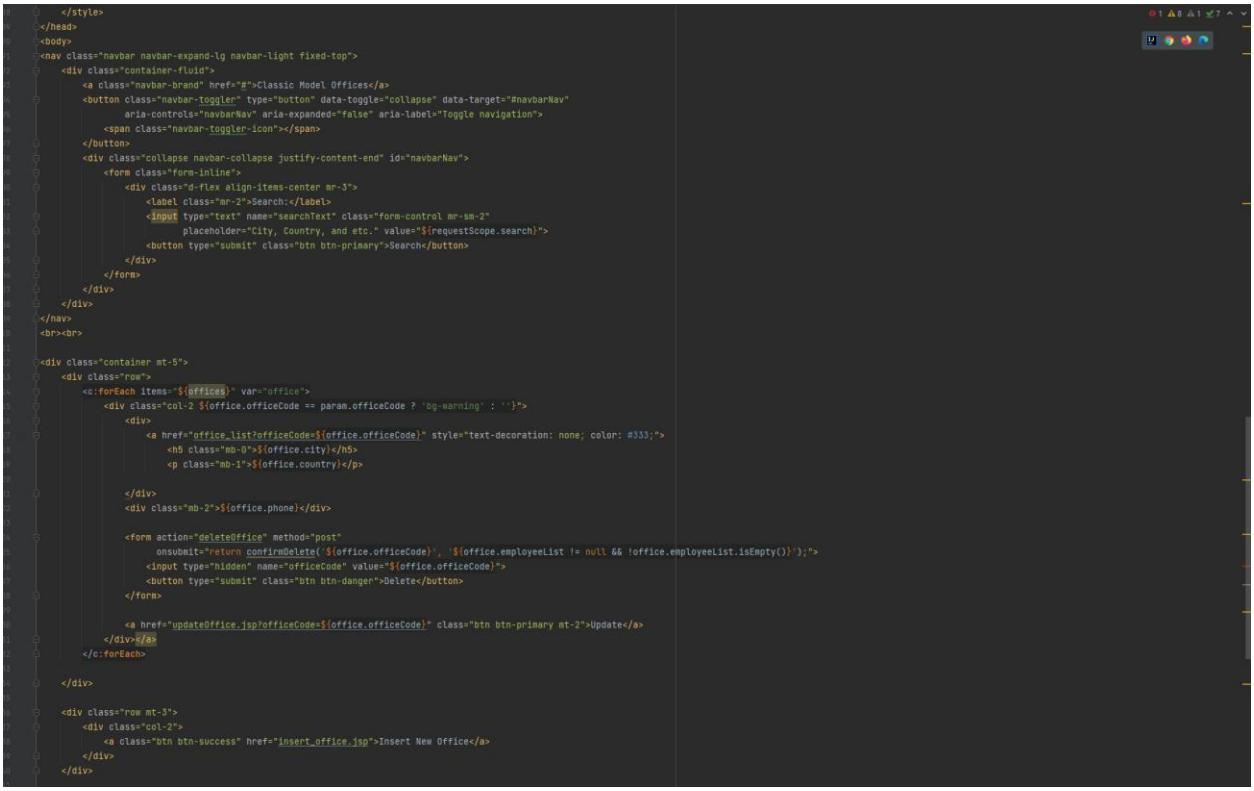
Update.jsp



```
    <title>Update Office</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">
</head>
<body>
<div class="container">
    <div class="row bg-primary">
        <h2>Update Office</h2>
    </div>

    <div class="row">
        <form action="UpdateOffice" method="post">
            <div class="mb-3">
                <label for="officeCode" class="form-label">Office Code:</label>
                <input type="text" id="officeCode" name="officeCode" class="form-control" value="${param.officeCode}" readonly>
            </div>
            <div class="mb-3">
                <label for="addressLine1" class="form-label">Address Line 1:</label>
                <input type="text" id="addressLine1" name="addressLine1" class="form-control" value="${param.addressLine1}" placeholder="Enter new address line 1" required>
            </div>
            <div class="mb-3">
                <label for="addressLine2" class="form-label">Address Line 2:</label>
                <input type="text" id="addressLine2" name="addressLine2" class="form-control" value="${param.addressLine2}" placeholder="Enter new address line 2">
            </div>
            <div class="mb-3">
                <label for="city" class="form-label">City:</label>
                <input type="text" id="city" name="city" class="form-control" value="${param.city}" placeholder="Enter new city" required>
            </div>
            <div class="mb-3">
                <label for="state" class="form-label">State:</label>
                <input type="text" id="state" name="state" class="form-control" value="${param.state}" placeholder="Enter new state">
            </div>
            <div class="mb-3">
                <label for="country" class="form-label">Country:</label>
                <input type="text" id="country" name="country" class="form-control" value="${param.country}" placeholder="Enter new country" required>
            </div>
            <div class="mb-3">
                <label for="postalCode" class="form-label">Postal Code:</label>
                <input type="text" id="postalCode" name="postalCode" class="form-control" value="${param.postalCode}" placeholder="Enter new postal code">
            </div>
            <div class="mb-3">
                <label for="phone" class="form-label">Phone:</label>
                <input type="text" id="phone" name="phone" class="form-control" value="${param.phone}" placeholder="Enter new phone number">
            </div>
            <div class="mb-3">
                <label for="territory" class="form-label">Territory:</label>
                <input type="text" id="territory" name="territory" class="form-control" value="${param.territory}" placeholder="Enter new territory">
            </div>

            <div class="mb-3">
                <input type="submit" value="Update" class="btn btn-primary">
            </div>
        </form>
    </div>
```

Insert.jsp

```jsp
    <div class="row">
        <form action="office_list" method="get">
            <div class="mb-3">
                <label for="officeCode" class="form-label">Office Code:</label>
                <input type="text" id="officeCode" name="officeCode" class="form-control" required>
            </div>
            <div class="mb-3">
                <label for="addressLine1" class="form-label">Address Line 1:</label>
                <input type="text" id="addressLine1" name="addressLine1" class="form-control" required>
            </div>
            <div class="mb-3">
                <label for="addressLine2" class="form-label">Address Line 2:</label>
                <input type="text" id="addressLine2" name="addressLine2" class="form-control">
            </div>
            <div class="mb-3">
                <label for="city" class="form-label">City:</label>
                <input type="text" id="city" name="city" class="form-control" required>
            </div>
            <div class="mb-3">
                <label for="state" class="form-label">State:</label>
                <input type="text" id="state" name="state" class="form-control">
            </div>
            <div class="mb-3">
                <label for="country" class="form-label">Country:</label>
                <input type="text" id="country" name="country" class="form-control" required>
            </div>
            <div class="mb-3">
                <label for="postalCode" class="form-label">Postal Code:</label>
                <input type="text" id="postalCode" name="postalCode" class="form-control">
            </div>
            <div class="mb-3">
                <label for="phone" class="form-label">Phone:</label>
                <input type="text" id="phone" name="phone" class="form-control">
            </div>
            <div class="mb-3">
                <label for="territory" class="form-label">Territory:</label>
                <input type="text" id="territory" name="territory" class="form-control">
            </div>

            <div class="mb-3">
                <input type="submit" value="Insert" class="btn btn-success">
            </div>
        </form>
    </div>


    <div class="row">
        <div class="col-2">
            <a class="btn btn-secondary" href="office_list">Back to Office List</a>
        </div>
    </div>
</div>
</body>
</html>
```



```
    </style>
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-light fixed-top">
    <div class="container-fluid">
        <a class="navbar-brand" href="#">Classic Model Offices</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav"
                aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse justify-content-end" id="navbarNav">
            <form class="form-inline">
                <div class="d-flex align-items-center mr-3">
                    <label class="mr-2">Search:</label>
                    <input type="text" name="searchText" class="form-control mr-sm-2"
                           placeholder="City, Country, and etc." value="${requestScope.search}">
                    <button type="submit" class="btn btn-primary">Search</button>
                </div>
            </form>
        </div>
    </div>
</nav>
<br><br>

<div class="container mt-5">
    <div class="row">
        <c:forEach items="${offices}" var="office">
            <div class="col-2 ${office.officeCode == param.officeCode ? 'bg-warning' : ''}">
                <div>
                    <a href="office_list?officeCode=${office.officeCode}" style="text-decoration: none; color: #333;">
                        <h5 class="mb-0">${office.city}</h5>
                        <p class="mb-1">${office.country}</p>

                    </div>
                    <div class="mb-2">${office.phone}</div>

                    <form action="deleteOffice" method="post"
                          onsubmit="return confirmDelete('${office.officeCode}', '${office.employeeList != null && !office.employeeList.isEmpty()}');">
                        <input type="hidden" name="officeCode" value="${office.officeCode}">
                        <button type="submit" class="btn btn-danger">Delete</button>
                    </form>

                    <a href="updateOffice.jsp?officeCode=${office.officeCode}" class="btn btn-primary mt-2">Update</a>
                </div></a>
            </c:forEach>

    </div>

    <div class="row mt-3">
        <div class="col-2">
            <a class="btn btn-success" href="insert_office.jsp">Insert New Office</a>
        </div>
    </div>
```

```
    <div class="row mt-5 bg-light">
        <b>Employees ::</b>
    </div>

    <div class="row mt-3">
        <c:forEach items="${offices}" var="office">
            <c:if test="${office.officeCode == param.officeCode}">
                <c:set var="selectedOffice" value="${office}" />
            </c:if>
        </c:forEach>

        <c:forEach items="${selectedOffice.employeeList}" var="employee">
            <div class="col-2 pl-2 mt-2 border rounded-pill">
                <div>${employee.firstName} ${employee.lastName} - ${employee.jobTitle}</div>
            </div>
        </c:forEach>
    </div>
</div>


</body>
</html>
```