Java Technologies for Web Applications

Lab Guides

Document Code	**25e-BM/HR/HDCV/FSOFT**
Version	**1.1**
Effective Date	**20/11/2012**

Hanoi, 04/2019

RECORD OF CHANGES

No	Effective Date	Change Description	Reason	Reviewer	Approver
1	25/Jun/2018	Create a new Lab	Create new	DieuNT1	VinhNV
2	01/May/2019	Update Fsoft Template	Update	DieuNT1	VinhNV

Contents

CODE:	NWEB.M.L201
TYPE:	Medium
LOC:	
DURATION:	180 MINUTES

Unit 2 - MVC Model and Session Tracking

Objectives:

- ✓ Understand the basic concepts of web development technologies with java (JSP / Servlet)
- ✓ Able to write servlets using the Java programming language (Java servlets)
- ✓ Create dynamic HTML content with Servlets and JavaServer Pages, using the Expression Language, and the JSP Standard Tag Library (JSTL)
- ✓ Create robust web applications using MVC architecture, session management, filters, and database integration (JDBC)
- ✓ Make Servlets and JSP work together cleanly
- ✓ Create secure web applications using the features of the Java EE web container

Descriptions and Guidelines:

Link Bootstrap 4:

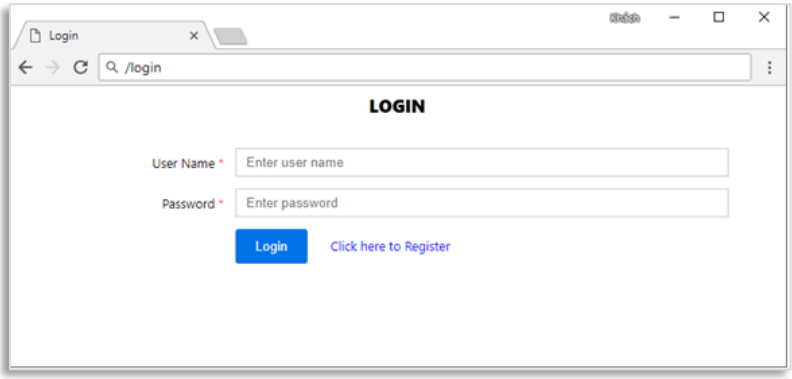
```
<link rel="stylesheet"
      href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css">
```

Link Font Awesome:

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
```

Step1: Create login page the following as:

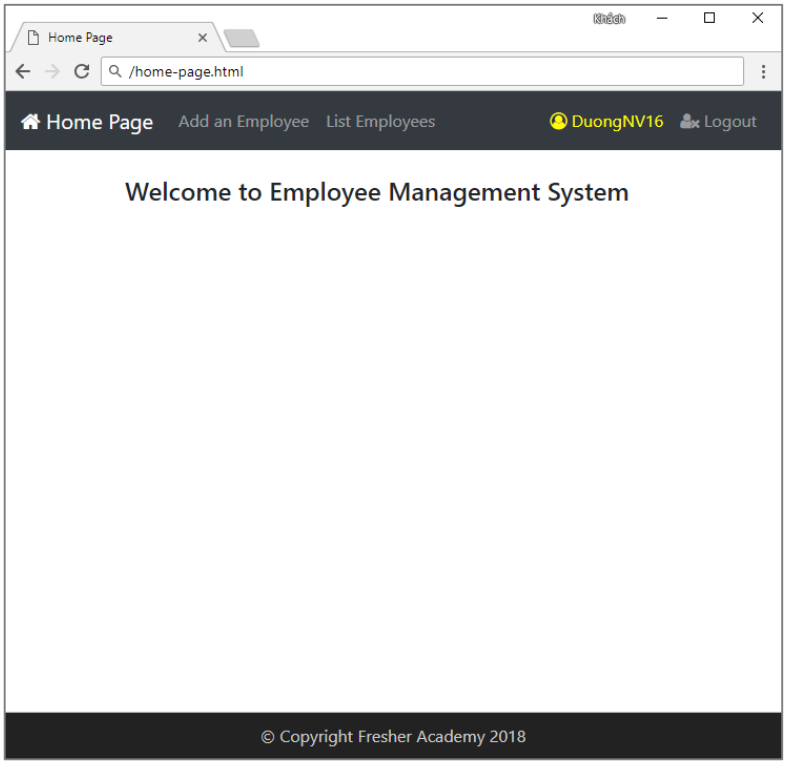
✓ *login.jsp screen:*



Screen 01_Layout 01

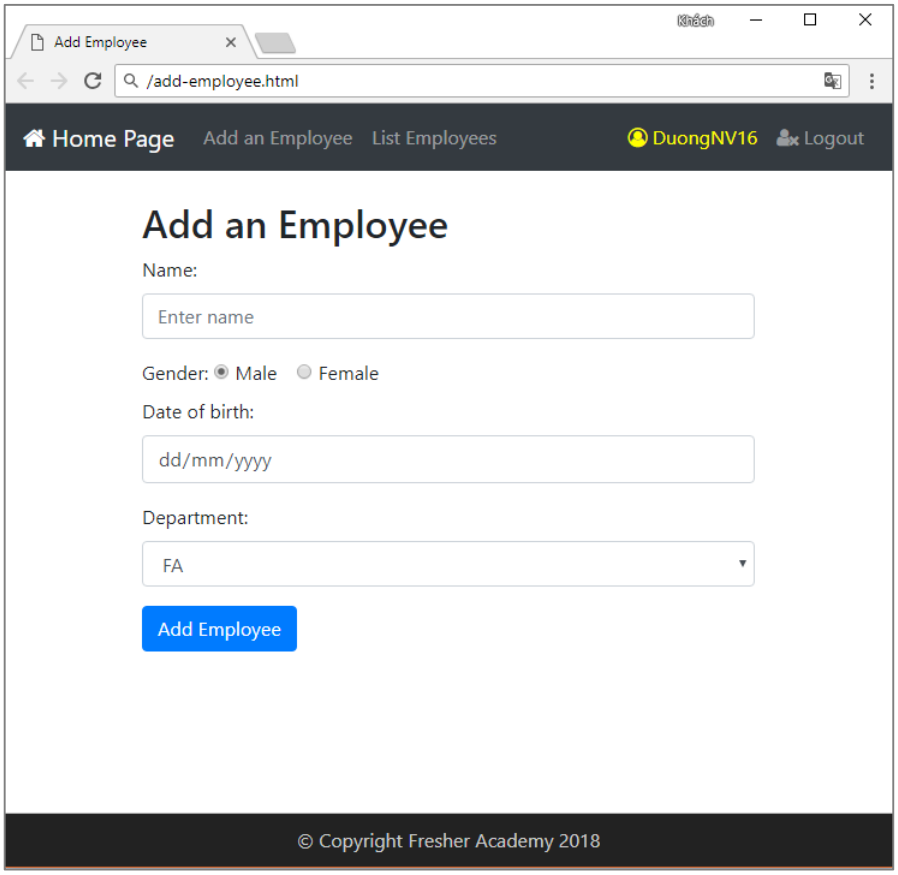
Source code download here: login.jsp , login.css , login.js

✓ *home-page.jsp screen:*

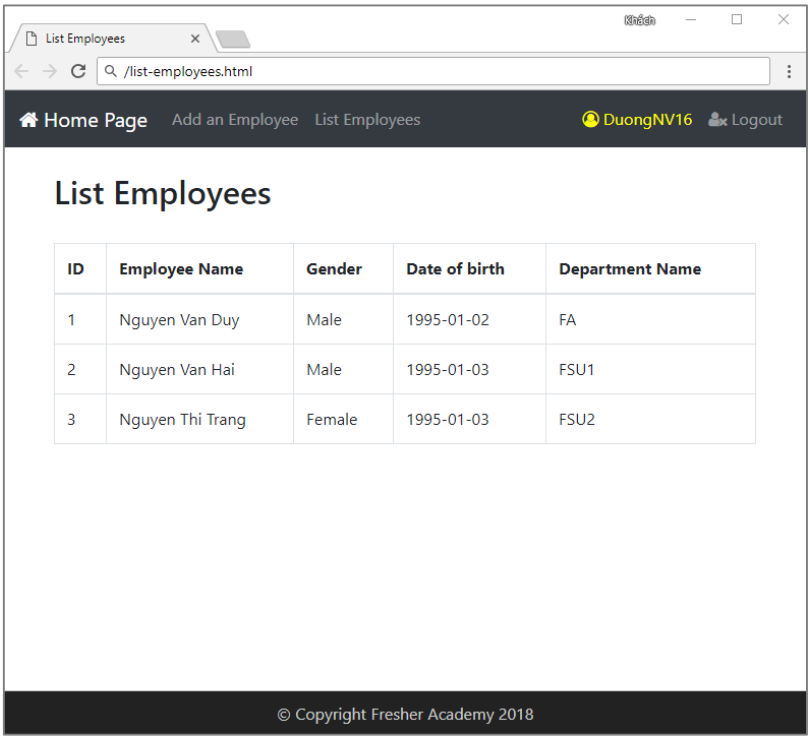


Source code download here: home-page.jsp

✓ *add-employee.jsp screen:*

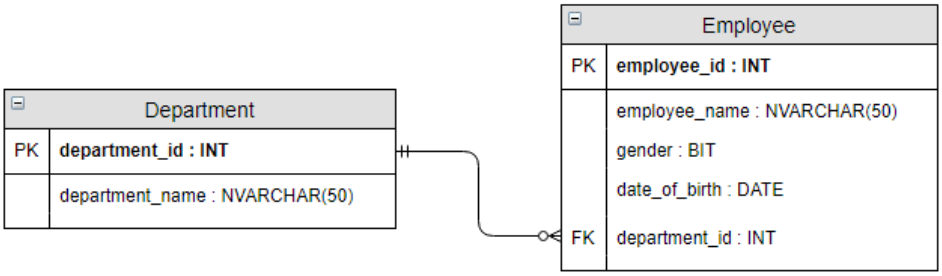


✓ *list-employees.jsp screen:*



Step2: Create Database

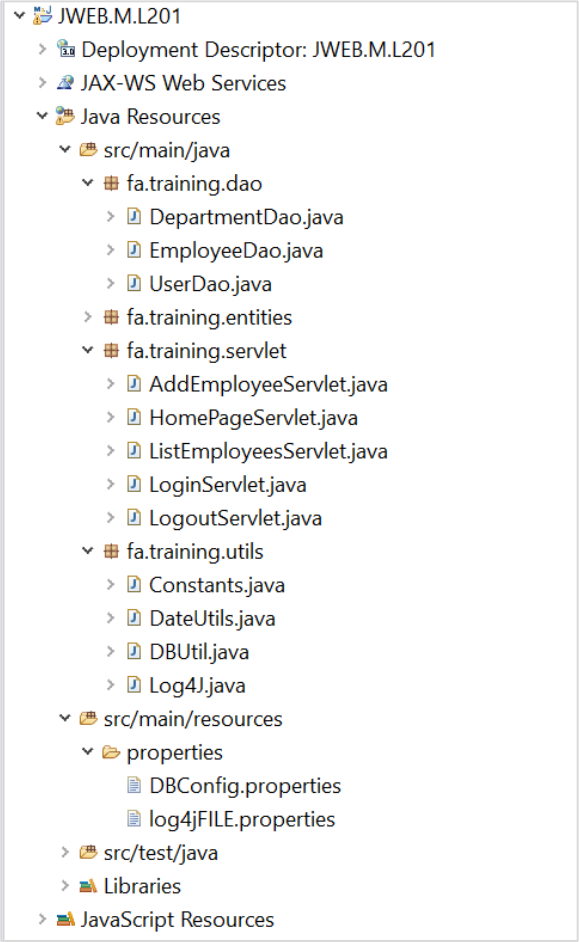
Create a database named "**JNWEBML201_SMS**" có các bảng và quan hệ như sau:

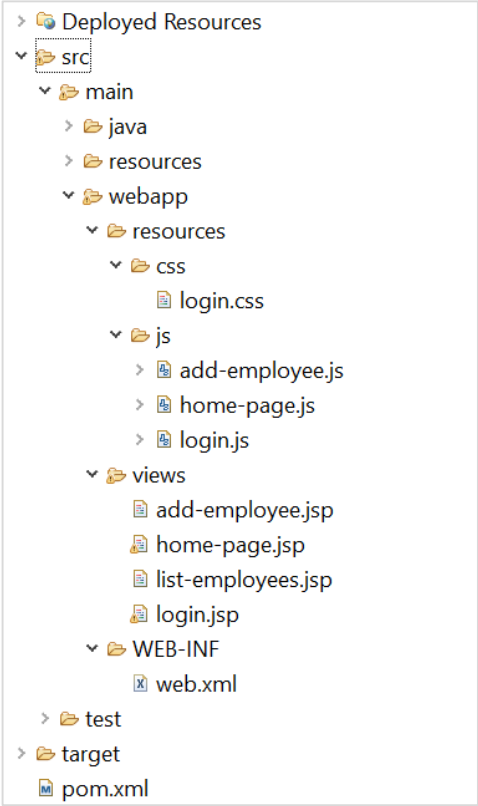


Tạo stored procedure "**usp_registerUser**" như sau:

```
CREATE PROC [dbo].[usp_registerUser]
@firstName VARCHAR(50),
@lastName VARCHAR(50),
@email VARCHAR(100),
@userName VARCHAR(50),
@password VARCHAR(50)
AS
BEGIN
    INSERT INTO Users VALUES (@firstName, @lastName, @email, @userName, @password)
END
```

Step3: Create a maven project named "JWEB.M.L201":





File **pom.xml**

```
1.  <dependencies>
2.              <dependency>
3.                      <groupId>javax.servlet</groupId>
4.                      <artifactId>javax.servlet-api</artifactId>
5.                      <version>3.1.0</version>
6.              </dependency>
7.
8.              <dependency>
9.                      <groupId>com.microsoft.sqlserver</groupId>
10.                     <artifactId>mssql-jdbc</artifactId>
11.                     <version>7.0.0.jre8</version>
12.             </dependency>
13.
14.             <dependency>
15.                     <groupId>log4j</groupId>
16.                     <artifactId>log4j</artifactId>
17.                     <version>1.2.17</version>
18.             </dependency>
19.
20.             <dependency>
21.                     <groupId>javax.servlet</groupId>
22.                     <artifactId>jstl</artifactId>
23.                     <version>1.2</version>
24.             </dependency>
25.     </dependencies>
26.     <build>
27.             <finalName>JavaWeb_P_L002</finalName>
28.             <plugins>
29.                     <plugin>
```

```
30.                         <groupId>org.apache.maven.plugins</groupId>
31.                         <artifactId>maven-compiler-plugin</artifactId>
32.                         <version>3.7.0</version>
33.                         <configuration>
34.                                 <source>1.8</source>
35.                                 <target>1.8</target>
36.                         </configuration>
37.                 </plugin>
38.
39.
40.                 <plugin>
41.                         <groupId>org.apache.maven.plugins</groupId>
42.                         <artifactId>maven-war-plugin</artifactId>
43.                         <version>3.2.2</version>
44.                         <configuration>
45.                                 <warSourceDirectory>src/main/webapp
46.                                 </warSourceDirectory>
47.                                 <failOnMissingWebXml>
48.                                         false
49.                                 </failOnMissingWebXml>
50.                         </configuration>
51.                 </plugin>
52.         </plugins>
53.     </build>
```

Step4: Process login

✓ Create **User** class in package **fa.training.entities**:

```
User.java
 9  public class User {
10      private String firstName;
11      private String lastName;
12      private String email;
13      private String userName;
14      private String password;
15
16⊕     public User(String firstName, String lastName, String email, String userName,
25
26⊕     public User(String userName, String password) {
31
32⊕     public User(String firstName, String lastName, String email,
40
41⊕     public User() {
44
45⊕     public String getFirstName() {
48
49⊕     public void setFirstName(String firstName) {
52
53⊕     public String getLastName() {
56
57⊕     public void setLastName(String lastName) {
60
61⊕     public String getEmail() {
64
65⊕     public void setEmail(String email) {
68
69⊕     public String getUserName() {
72
73⊕     public void setUserName(String userName) {
76
77⊕     public String getPassword() {
80
81⊕     public void setPassword(String password) {
```

✓ Create package **fa.training.servlet** to contains Servlet class.

✓ Create **LoginServlet** class in package **fa.training.servlet** to handle login the following:

```
1.  @WebServlet(urlPatterns = "/login")
2.  public class LoginServlet extends HttpServlet {
3.
4.     private static final long serialVersionUID = 1L;
5.
6.      private static UserDao userDao = new UserDao();
7.
8.      @Override
9.      protected void doPost(HttpServletRequest request, HttpServletResponse response)
10.             throws ServletException, IOException {
11.
12.         Log4J.getLogger().info("Running on doPost method of LoginServlet");
13.
14.         String userName = request.getParameter("userName");
15.         String password = request.getParameter("password");
16.         User user = new User(userName, password);
17.
18.         try {
19.             if (userDao.login(user)) {
20.                 HttpSession session = request.getSession();
21.                 // if login successfully, save session user, who have just logined
22.                 session.setAttribute("userLogin", user);
23.                 response.sendRedirect(request.getContextPath() + "/home");
24.             } else {
25.                 request.setAttribute("userRegister", user);
26.                 request.setAttribute("loginFail", "User name or password is incorrect");
27.                 request.getRequestDispatcher("/views/login.jsp").forward(request, response);
28.             }
29.         } catch (ClassNotFoundException e) {
30.             Log4J.getLogger().
31.                           error("Class not found exception in method doPost of LoginServlet");
32.         } catch (SQLException e) {
33.             Log4J.getLogger().error("SQL exception in method doPost of LoginServlet");
34.         }
35.     }
36.
37.     @Override
38.     protected void doGet(HttpServletRequest request, HttpServletResponse response)
39.             throws ServletException, IOException {
40.         Log4J.getLogger().info("Running on doGet method of LoginServlet");
41.         request.getRequestDispatcher("views/login.jsp").forward(request, response);
42.     }
43.
44. }
```

If the login is successful, it will save the logged-in session user, and redirect to request **/home** to display the home-page.

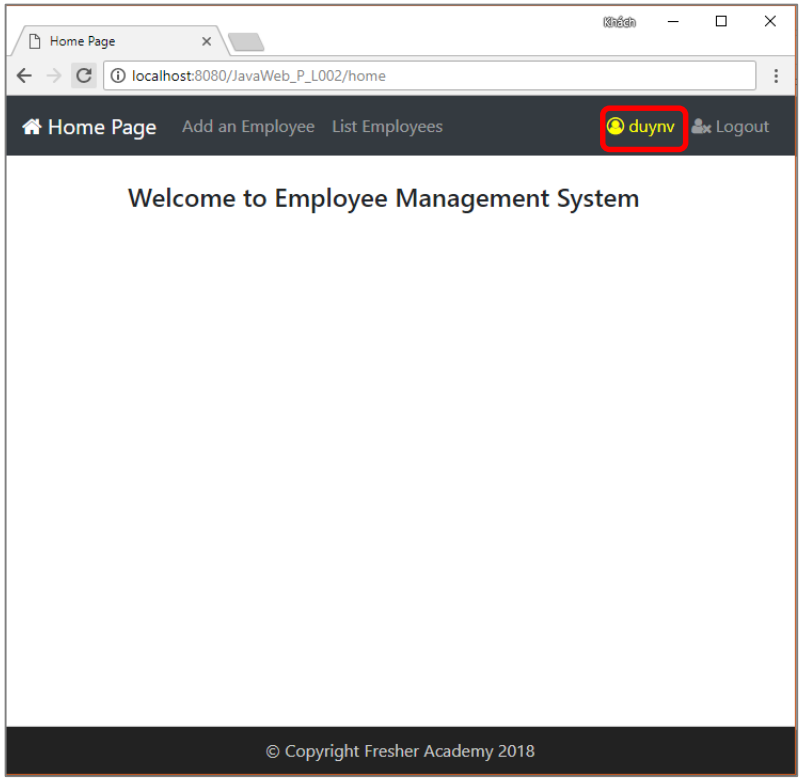
✓ Create a **HomePageServlet** class to handle this requestion:

```
1.  @WebServlet("/home")
2.  public class HomePageServlet extends HttpServlet {
3.
4.      private static final long serialVersionUID = 1L;
5.
6.      @Override
7.      protected void doGet(HttpServletRequest request, HttpServletResponse response)
8.                     throws ServletException, IOException {
9.
10.             request.getRequestDispatcher("/views/home-page.jsp").forward(request, response);
11.     }
12. }
```

✓ In **home-page.jsp**, show login **userName** here:

```
1.      <ul class="navbar-nav ml-auto">
2.      <li class="nav-item">
3.              <a class="nav-link" style="color: yellow" href="#">
4.                      <i class="fa fa-user-circle-o"></i>
5.                              ${userLogin.userName}
6.              </a>
7.              </li>
8.      <li class="nav-item">
9.              <a class="nav-link" href="<%=request.getContextPath()%>/logout">
10.                     <i class="fa fa-user-times"></i> Logout
11.             </a>
12.     </li>
13.     </ul>
```

Screen result:



✓ Continue, create a **LogoutServlet** class the following as:

```
1.  @WebServlet("/logout")
2.  public class LogoutServlet extends HttpServlet {
3.
4.      private static final long serialVersionUID = 1L;
5.
6.      @Override
7.      protected void doGet(HttpServletRequest request, HttpServletResponse response)
8.                      throws ServletException, IOException {
9.              Log4J.getLogger().info("Logging out");
10.             // remove session userLogin
11.             request.getSession().removeAttribute("userLogin");
12.             // redirect to /login
13.             response.sendRedirect(request.getContextPath() + "/login");
14.     }
15. }
```

Step5. Create entity classes

✓ Create package fa**.training.entities** and the entity classes the following as:

Department.java class:

```
1.  public class Department {
2.      private int departmentId;
3.      private String departmentName;
4.
5.      // Constructors and getters/setters
6.  }
```

Employee.java class:

```
1.  public class Employee {
2.      private int employeeId;
3.      private String employeeName;
4.      private byte gender;
5.      private Date dateOfBirth;
6.      private int departmentId;
7.
8.      // Constructors and getters/setters
9.  }
```

Step6. Use jQuery/AJAX to send the requests to server

✓ Create **home-page.js** the following as:

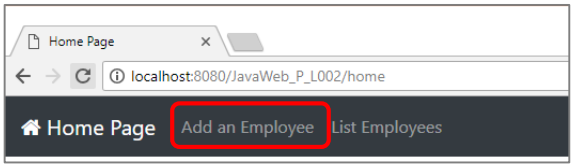
```
1.  $(document).ready(function() {
2.      $("#addEmpLink").click(function() {
3.              $.get({
4.                      url : "/JavaWeb_P_L002/add-emp",
5.                      success : function(response) {
6.                              $(".container").html(response);
7.                      }
8.              });
9.      });
10.
11.     $("#listEmpsLink").click(function() {
12.             $.get({
13.                     url : "/JavaWeb_P_L002/list-employees",
14.                     success : function(response) {
15.                             $(".container").html(response);
16.                     }
17.             });
18.     });
19.
20. });
```

✓ Link to **home-page.js** in **home-page.jsp**:

```
1.      <script type="text/javascript"
2.              src="<%=request.getContextPath()%>/resources/js/home-page.js"></script>
```

✓ Create **AddEmployeeServlet** class in **fa.training.servlet** the following as:

When the user selects "**Add an Employee**" link, the request will be processed by ***doGet***() method in **AddEmployeeServlet** class:



```
1.  @WebServlet("/add-emp")
2.  public class AddEmployeeServlet extends HttpServlet {
3.      private static final long serialVersionUID = 1L;
4.
5.      private DepartmentDao departmentDao = new DepartmentDao();
6.      private EmployeeDao employeeDao = new EmployeeDao();
7.
8.      @Override
9.      protected void doGet(HttpServletRequest request, HttpServletResponse response)
10.             throws ServletException, IOException {
11.         try {
12.             // Get all of departments from DB and display on selected-box in
13.             // add-employee.jsp page
14.             List<Department> listOfDepartment = departmentDao.findAllDepartment();
15.             request.setAttribute("listOfDepartment", listOfDepartment);
16.
17.             // This method doGet(): trả về response là trang add-employee.jsp cho ajax để
18.             // hiển thi trên trang home-page
19.
20.             request.getRequestDispatcher("/views/add-employee.jsp").
21.                                                     forward(request, response);
22.
23.         } catch (ClassNotFoundException | SQLException e) {
24.             Log4J.getLogger().error(e.getMessage());
25.         }
26.
27.     }
28.
29.     // Method doPost(): xử lý khi click button “Add Employee”
30.
31.     @Override
32.     protected void doPost(HttpServletRequest request, HttpServletResponse response)
33.             throws ServletException, IOException {
34.
35.      int deptId = Integer.parseInt(request.getParameter("deptId"));
36.      String employeeName = request.getParameter("employeeName");
37.         byte gender = Byte.parseByte(request.getParameter("gender"));
38.         Date dateOfBirth = null;
39.
40.     try {
41.             dateOfBirth = DateUtils.convertStringToDate(request.
42.                                                 getParameter("dateOfBirth"));
43.         } catch (ParseException e) {
44.             Log4J.getLogger().error("Parse Exception when convert string to date");
45.         }
46.
47.
48.         Employee employee = new Employee(employeeName, gender, dateOfBirth, deptId);
49.
50.      try {
51.             employeeDao.addEmployee(employee);
52.             List<Department> listOfDepartment = departmentDao.findAllDepartment();
53.             request.setAttribute("listOfDepartment", listOfDepartment);
54.             request.setAttribute("employee", employee);
55.             request.setAttribute("message", "Add new employee successfully");
56.
57.             request.getRequestDispatcher("/views/add-employee.jsp").
58.                                                     forward(request, response);
59.
60.         } catch (ClassNotFoundException | SQLException e) {
61.             Log4J.getLogger().error("An exception occurs");
62.         }
63.     }
64.
65. }
```

✓ **add-employee.jsp** page:

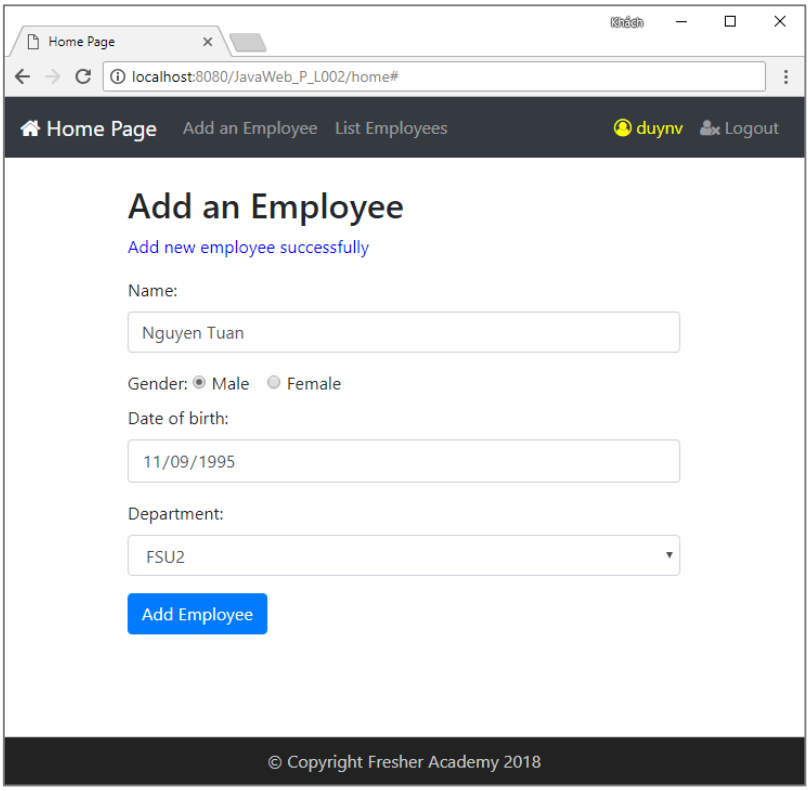
```
1.  <%@ page language="java" contentType="text/html; charset=UTF-8"
2.      pageEncoding="UTF-8"%>
3.  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
4.  <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
5.
6.  <body>
7.  <form action="#" method="post" name="frm-addEmp">
8.      <div class="row">
9.      <div class="col-md-6 offset-md-3">
10.             <h2>Add a Employee</h2>
11.             <p style="color: blue">${message}</p>
12.             <div class="form-group">
13.                     <label for="employeeName">Name:</label>
14.                     <input type="text" class="form-control" id="employeeName"
15.                             placeholder="Enter name" name="employeeName"
16.                             value="${employee.employeeName}">
17.             </div>
18.
19.             <label for="gender">Gender:</label>
20.             <div class="form-check-inline">
21.             <label class="form-check-label">
22.                     <input type="radio" class="form-check-input" name="gender"
23.                                                     value="1" checked>Male
24.             </label>
25.             </div>
26.
27.             <div class="form-check-inline">
28.             <label class="form-check-label">
29.                     <input type="radio" class="form-check-input" name="gender"
30.                             value="0" ${employee.gender==0 ? 'checked' : '' }>Female
31.             </label>
32.             </div>
33.
34.             <div class="form-group">
35.             <label for="dateOfBirth">Date of birth:</label>
36.                     <input type="date" class="form-control" id="dateOfBirth"
37.                             placeholder="Enter date of birth" name="dateOfBirth"
38.                             value="<fmt:formatDate
39.                             value='${employee.dateOfBirth}' pattern='yyyy-MM-dd' />">
40.             </div>
41.
42.             <div class="form-group">
43.             <label for="dept">Department:</label>
44.                     <select class="form-control" id="dept">
45.                             <c:forEach items="${listOfDepartment}" var="department">
46.                                     <option value="${department.departmentId}"
47.                                     ${department.departmentId==employee.departmentId
48.                                     ? 'selected' : '' }>
49.                                             ${department.departmentName}
50.                                     </option>
51.                             </c:forEach>
52.
53.                     </select>
54.             </div>
55.     </div>
56.
57.     </div>
58.
59.     <div class="row">
60.             <div class="col-md-6 offset-md-3">
61.             <button type="button" id="btn-addEmp" class="btn btn-primary">
62.                     Add Employee
63.             </button>
64.             </div>
65.     </div>
66. </form>
```

```
67.
68. <script type="text/javascript" src="<%=request.getContextPath()%>/resource">
69. </script>
70.
71. </body>
```

✓ **add-employee.js** file:

```
1.  $(document).ready(function() {
2.      $("#btn-addEmp").click(function() {
3.              var employeeName = $("#employeeName").val();
4.              var gender = $("input[name=gender]:checked").val();
5.              var dateOfBirth = $("#dateOfBirth").val();
6.              var deptId = $("#dept").val();
7.
8.              $.post({
9.                      url : "/JavaWeb_P_L002/add-emp",
10.                     data : {
11.                             employeeName : employeeName,
12.                             gender : gender,
13.                             dateOfBirth : dateOfBirth,
14.                             deptId : deptId
15.                     },
16.                     success : function(response) {
17.                             $(".container").html(response);
18.                     }
19.             });
20.     });
21. });
```

Result page:



✓ Create **ListEmployeeServlet** class in fa.training.servlet the following as:

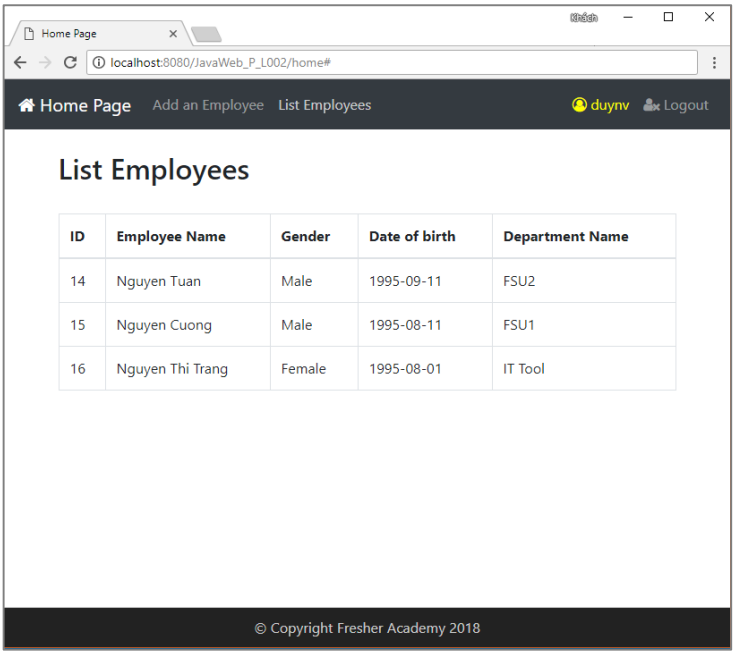
```
1.  @WebServlet("/list-employees")
2.  public class ListEmployeesServlet extends HttpServlet {
3.
4.      private static final long serialVersionUID = 1L;
5.      private EmployeeDao employeeDao = new EmployeeDao();
```

```
6.      private DepartmentDao departmentDao = new DepartmentDao();
7.
8.      @Override
9.      protected void doGet(HttpServletRequest request, HttpServletResponse response)
10.             throws ServletException, IOException {
11.         try {
12.             List<Employee> listOfEmployee = employeeDao.findAllEmployee();
13.             List<Department> listOfDepartment = departmentDao.findAllDepartment();
14.             request.setAttribute("listOfEmployee", listOfEmployee);
15.             request.setAttribute("listOfDepartment", listOfDepartment);
16.
17.             request.getRequestDispatcher("/views/list-employees.jsp").
18.                                                         forward(request, response);
19.
20.         } catch (ClassNotFoundException | SQLException e) {
21.                 Log4J.getLogger().error(e.getMessage());
22.         }
23.
24.     }
25. }
```

✓ **list-employee.jsp** page:

```
1.  <%@ page language="java" contentType="text/html; charset=UTF-8"
2.      pageEncoding="UTF-8"%>
3.  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
4.  <body>
5.      <br />
6.      <h2>List of Employees</h2>
7.      <br />
8.      <table class="table table-bordered">
9.              <thead>
10.                     <tr>
11.                             <th>ID</th>
12.                             <th>Employee Name</th>
13.                             <th>Gender</th>
14.                             <th>Date of birth</th>
15.                             <th>Department Name</th>
16.                     </tr>
17.             </thead>
18.             <tbody>
19.                     <c:forEach items="${listOfEmployee}" var="employee">
20.                             <tr>
21.                                     <td>${employee.employeeId}</td>
22.                                     <td>${employee.employeeName}</td>
23.                                     <td>${employee.gender == 1 ? 'Male' : 'Female'}</td>
24.                                     <td>${employee.dateOfBirth}</td>
25.                                     <c:forEach items="${listOfDepartment}" var="department">
26.                                     <c:if test="${employee.departmentId ==
27.                                                         department.departmentId}">
28.                                     <td>${department.departmentName}</td>
29.                                     </c:if>
30.                                     </c:forEach>
31.
32.                             </tr>
33.                     </c:forEach>
34.
35.             </tbody>
36.     </table>
37. </body>
```

Result page:



Step7: Some utility classes

Constants class

```
1.  package fa.training.utils;
2.
3.  public class Constants {
4.      public static final String REGISTER_SUCCESSFULLY_MESSAGE = "Register user successfully";
5.      public static final String REGISTER_FAIL_MESSAGE = "Register user fail";
6.  }
```

DateUtils class

```
1.  package fa.training.utils;
2.
3.  import java.text.ParseException;
4.  import java.text.SimpleDateFormat;
5.  import java.util.Date;
6.
7.  public class DateUtils {
8.    public static Date convertStringToDate(String dateString)
9.        throws ParseException {
10.     SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
11.     Date date = formatter.parse(dateString);
12.     return date;
13.   }
14.
15.   public static java.sql.Date convertJavaDateToSqlDate(Date javaDate) {
16.     java.sql.Date sqlDate = new java.sql.Date(javaDate.getTime());
17.     return sqlDate;
18.   }
19. }
```

Log4J class

```
1.  package fa.training.utils;
2.
3.  import org.apache.log4j.Logger;
4.  import org.apache.log4j.PropertyConfigurator;
```

```
5.
6.  /**
7.   * Class Log4J utility
8.   * @author FA
9.   *
10.  */
11. public class Log4J {
12.   private static final Logger logger = Logger.getLogger(Log4J.class);
13.
14.   /**
15.    * method configure Log4J.
16.    *
17.    * @return Logger logger
18.    */
19.   public static Logger getLogger() {
20.     PropertyConfigurator.configure(
21.         Log4J.class.getResourceAsStream("/properties/log4jFILE.properties"));
22.     return logger;
23.   }
24. }
```

-- THE END --