МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Програмна інженерія та   
інТЕЛЕКТУАЛЬні технології управління»

ЗВІТ

з лабораторної роботи № 1

з навчальної дисципліни

«Web додаткИ на основі Java»

ВИКОНАВ

Студент групи КН-221б

Тимур ОВЧАРЕНКО

ПЕРЕВІРИВ

Доцент кафедри ПІІТУ

кандидат технічних наук доцент

Дмитро ДВУХГЛАВОВ

Харків 2023

ДОСЛІДЖЕННЯ СТЕКУ ТЕХНОЛОГІЙ ДЛЯ СТВОРЕННЯ   
ВЕБ-ДОДАТКУ НА JAVA

Мета

1. Закріпити знання про розгортання програмного забезпечення для створення та запуску веб-додатків Java ЕЕ.
2. Набути навички створення розробки веб-додатків.
3. Набути навички розробки сервлетів.

Завдання роботи

1. Встановити програмне забезпечення для розробки веб-застосунків.

* Integrated Development Environment;
* Web Server або Application Server.

1. Розробити веб-застосування HelloWebWorld … .
2. Розробити веб-застосування AboutAuthor … .

Хід роботи

1 Встановлення програмного забезпечення для розробки веб-застосунків

1.1 Установка IDE

Для встановлення IDE Intellij IDEA потрібно виконати кроки, описані далі.

1. Перейти по посиланню <https://www.jetbrains.com/store/#commercial> для завантаження інсталяційного файлу.
2. Завантажити його у каталог downloads. Після цього запустити інсталяційний файл.

Після запуску інсталяційного файлу оберіть папку для встановлення додатку(див. рис. 1)

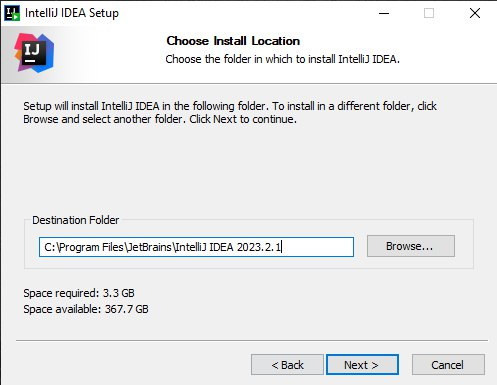


Рисунок 1 – Сторінка вибору папки для встановлення додатку

Потім оберіть налаштування для програми(див. рис. 2)

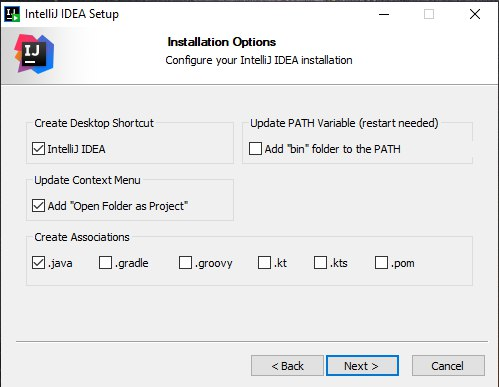


Рисунок 2 – Сторінка вибору налаштувань для програми

Запускаємо процес інсталяції(див. рис. 3)

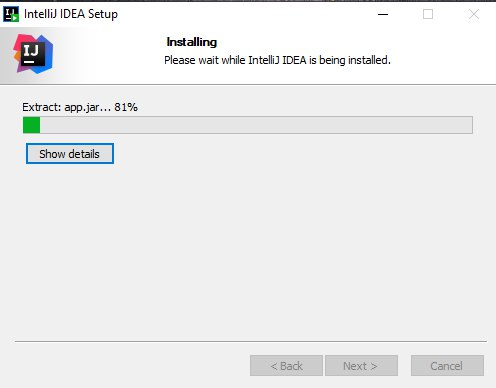


Рисунок 3 – Процес інсталяції програми

1.2 Установка серверу

Для встановлення web або application server tomcat 9.0.80необхідно:

1. Перейти по посиланню <https://tomcat.apache.org/download-90.cgi> для завантаження інсталяційного файлу.
2. Завантажити його у каталог downloads. Після цього запустити інсталяційний файл.

Пілся запуску інсталяційного файлу обираємо налаштування для tomcat(див. рис. 4)

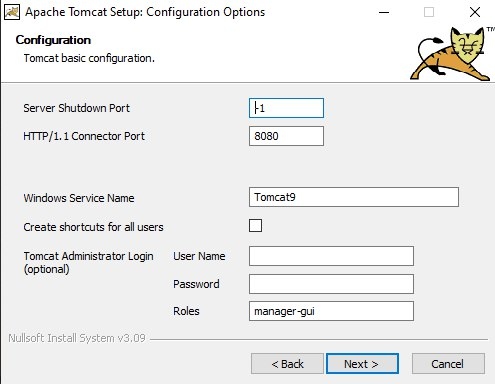


Рисунок 4 – Налаштування tomcat

Потім потрібно вибрати місце, куди було встановленно jdk(див. рис. 5)

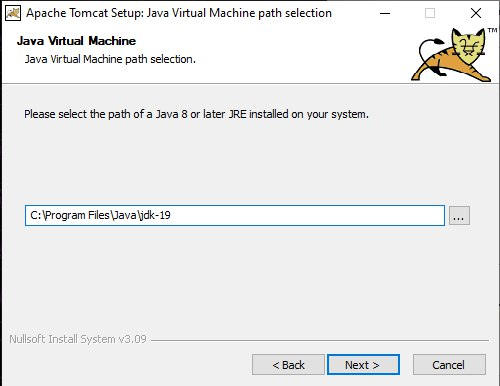


Рисунок 5 – Вибір місця куди було встановленно jdk

Вибір місця куди буде інстальовано програму(див. рис. 6)

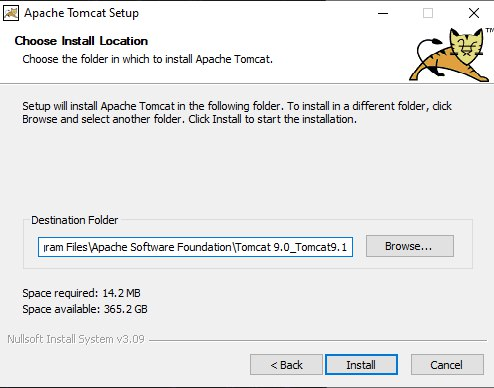


Рисунок 6 – Вибір місця куди буде встановленно програму

2 Дослідження створення веб-застосунків

**Завдання 1.** Встановити програмне забезпечення для розробки веб-застосунків на Java та розробити веб-застосунок, який відображає на екрані текстовий напис "Hello, WebWorld!!!", а також прізвище та ініціали студента, номер групи, в якій він навчається, та номер у списку групи.

Для створення проекту необхідно запустити Intellih IDEA та створити новий проект використовуючи Jakarta ee(див. рис. 7)

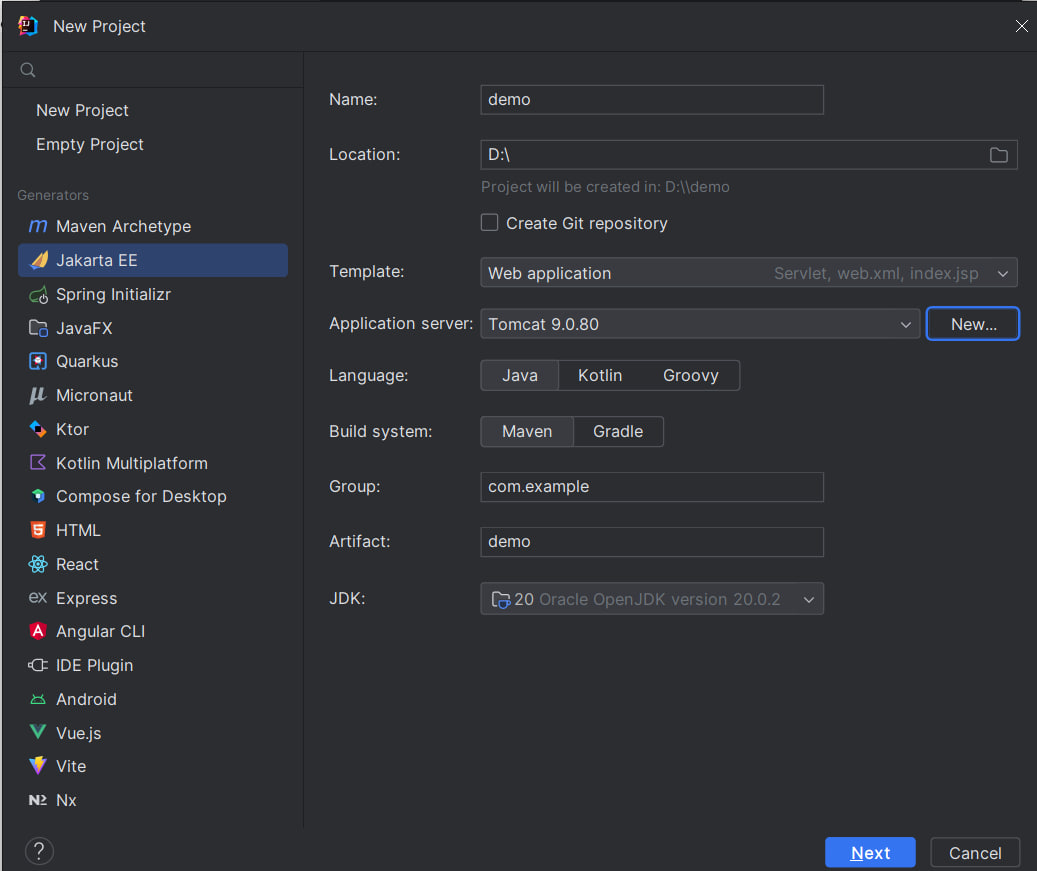


Рисунок 7 – Створення проекту за допомогою jakarta ee

Потім вибираємо зазделегідь встановленний веб сервер apache tomcat(див. рис. 8)

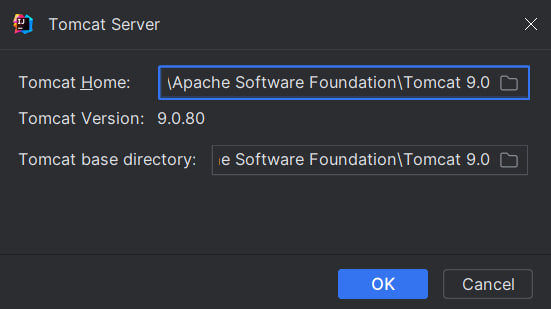


Рисунок 8 – Вибір веб серверу

Натискаємо далі та обираємо Java EE 8(див. рис. 9)

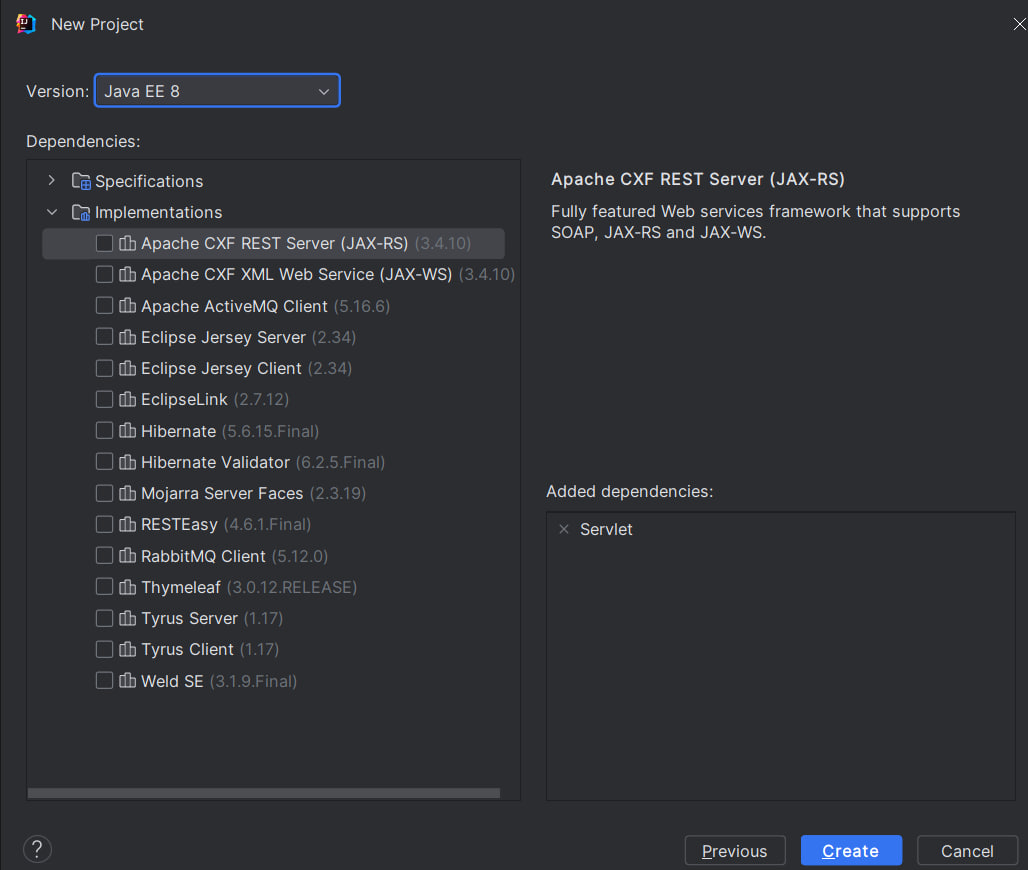


Рисунок 9 – Вибір версії Jakarta

Для запуску натискаємо на зелений трикутник(див. рис. 10)



Рисунок 10 – Кнопка для запуску проекту

Результат роботи після запуску проекту представленно на рисунку(див. рис. 11)

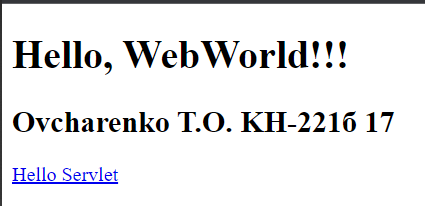


Рисунок 11 – Результат роботи

**Завдання 2.** Розробити веб-застосунок, який дозволяє відобразити на веб-сторінці перелік співробітників деякої ІТ-компанії.

**Завдання 3.** Модифікувати проєкт, створений після виконання завдання 2, шляхом додавання функціоналу переходу на готову сторінку із сервлету, використовуючи команди forward() та sendRedirect().

Результат роботи після запуску представлено на рисунку(див. Рис. 12)



Рисунок 12 – Результат роботи

Після натискання на посилання table переходимо до таблиці робітників(див. Рис. 13)

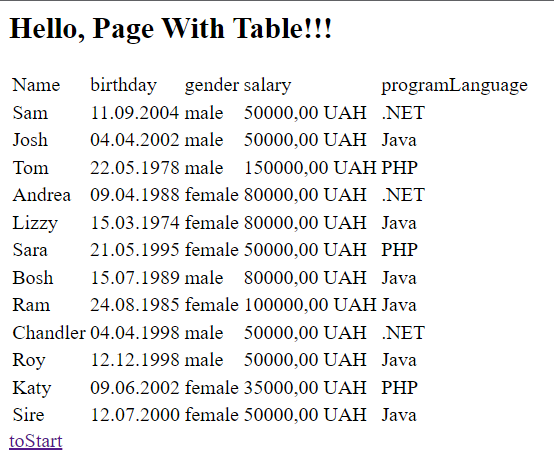


Рисунок 13 – Таблиця робітників

Після натискання на посилання table(redirect) переходимо до таблиці робітників за допомогою редіректу(див. Рис. 14)

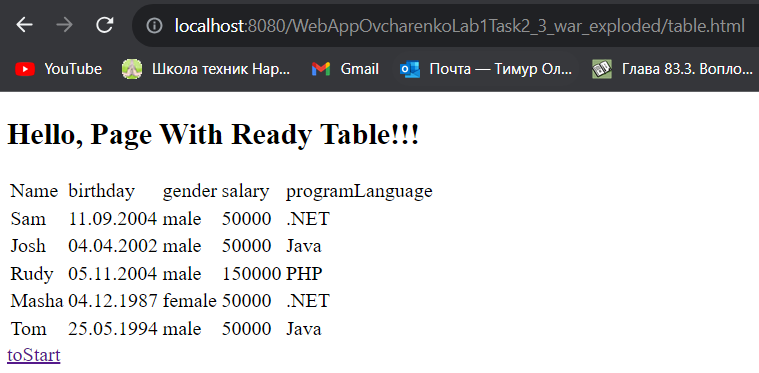


Рисунок 14 – Сторінка з таблицею до якої перейшли за допомогою редіректу

Після натискання на посилання table(forward) переходимо до таблиці робітників за допомогою редіректу(див. Рис. 15)

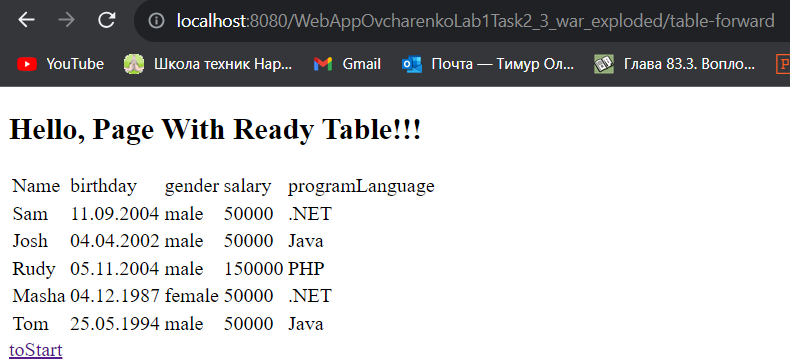


Рисунок 15 - Сторінка з таблицею до якої перейшли за допомогою форварду

**Завдання 4.** Розробити веб-застосунок, який містить наступний функціонал:

* показує стартову сторінку із полями для вводу логіну та паролю, після вводу правильної пари відкриває головку сторінку застосунку;
* показує на головній сторінці меню, що дозволяє перейти до сторінок, створених відповідно до завдань 4.1 та 4.2.

**Завдання 4.1.** Розробити функціонал, що формує JSP із списком співробітників у вигляді таблиці без застосування можливостей Expression Language (EL) та Java Standard Tag Library (JSTL).

**Завдання 4.2.** Розробити функціонал, що формує JSP із списком співробітників у вигляді таблиці із застосуванням EL та JSTL.

У ході виконання завдання 4 було використано такі jstl теги:

1. forEach – тег, який створює цикл в проміжку заданих значень або пропускає через цикл всі елементи массиву чи колекції
2. if – тег, що показує його зміст при виконані умови

У додатку А представлено код меню для навігації. У додатку Б представлено код, який відображає підпис студента, який виконав завдання. У додатку В показано код головнї сторінки веб-застосунку. У додатку Г представлено код, який відображає таблицю робітників. У додатку Е представлено код, який відображає таблицю робітників з використанням jstl. У додатку Є представлено код для виведення форми входу. У додатку Ж представлено код сервлету, який оброблює форму для входу до веб-застосунку. В ньмоу було використано метод doPost так, як ми відправляли форму для оброблення, тому в сервлеті булоа перевірка правильності введених даних. У додатку З представлено код сервлету який повертає сторінку з таблицею робітників. У додатку И представлено код сервлету який повертає сторінку з таблицею робітників створену з допомогою jstl.

На рисунку представлено форму для входу до веб-застосунку(див. Рис. 16)

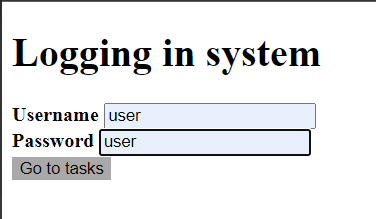


Рисунок 16 – Форма для входу

Після правильного вводу даних, відкривається головна сторінка(див. Рис. 17)

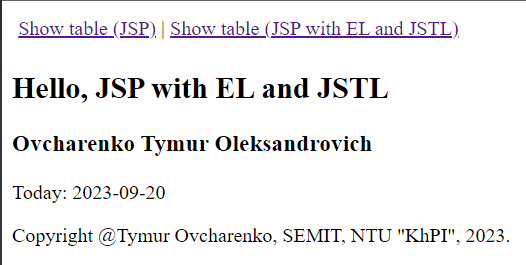


Рисунок 17 – Головна сторінка застосунку

На головній сторінці присутнє меню з посиланями на сторінки з таблицею робітників з(див. рис. 18) та без(див. рис. 19) використання jstl.

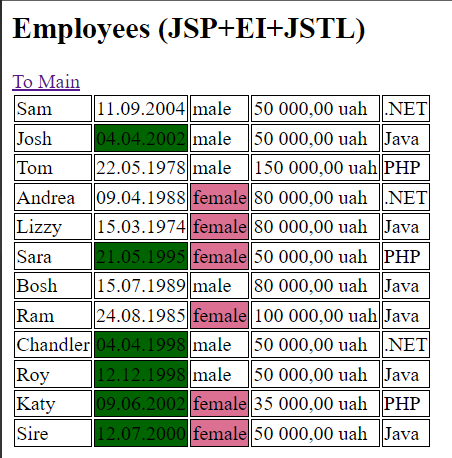


Рисунок 18 – Таблиця робітників з jstl

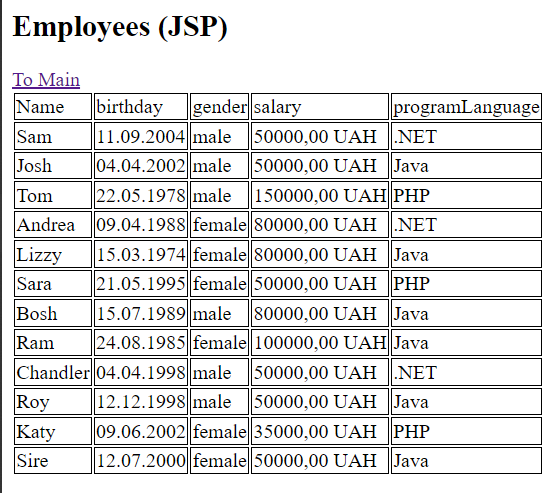


Рисунок 19 – Таблиця робітників

Висновки

При ході виконання лабараторної роботи було створено веб-застосунок. Спочатку було досить складно зрозуміти як працюють сервлети, так як ознайомившись з Spring Framework були отримані навички створення веб-застосунків і структура їх створення дещо відрізнялась від данної, тому було дивним створення нового класу для кожної кінцевої точки у веб-застосунку.

Виконання першого, другого та третього завдання не було складним так, як є велика кількість інформації в інтернеті, але при виконані четвертого виникла проблема з підключеням jstl тегів.

Вивчення даної технології є досить важливим, так як досить часто використовують саме Java як мову для backend розробки

# Додаток А Код сторінки main.jsp

<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Tymur KH-221b</title>  
</head>  
<body>  
 <div style="padding: 5px;">  
 <a href="show-jsp">Show table (JSP)</a>  
 <p style="display: inline">|</p>  
 <a href="show-jsp-jstl">Show table (JSP with EL and JSTL)</a>  
 </div>  
</body>  
</html>

# Додаток Б Код сторінки footer.jsp

<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Tymur KH-221b</title>  
</head>  
<body>  
 <p>Copyright @Tymur Ovcharenko, SEMIT, NTU "KhPI", 2023.</p>  
</body>  
</html>

# Додаток В Код сторінки JavaWEB\_L1\_OTO\_main.jsp

<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Tymur KH-221b</title>  
</head>  
<body>  
 <header>  
 <%@ **include** file="main.jsp"%>  
 </header>  
 <h2>Hello, JSP with EL and JSTL</h2>  
 <h3>Ovcharenko Tymur Oleksandrovich</h3>  
 <p>Today: <%=java.time.LocalDate.*now*()%></p>  
 <footer>  
 <%@ **include** file="footer.jsp"%>  
 </footer>  
</body>  
</html>

# Додаток Г Код сторінки JavaWEB\_L1\_OTO\_jsp.jsp

<%@ **page** import="com.example.webappovcharenkolab1task4.entity.Employee" %>  
<%@ **page** import="com.example.webappovcharenkolab1task4.entity.EmployeeList" %><%--  
 Created by IntelliJ IDEA.  
 User: 09662  
 Date: 12.09.2023  
 Time: 22:16  
 To change this template use File | Settings | File Templates.  
--%>  
<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Tymur KH-221b</title>  
</head>  
<body>  
 <h2>Employees (JSP)</h2>  
 <a href="./login">To Main</a>  
 <table>  
 <tr>  
 <td style="border: 1px solid;">Name</td>  
 <td style="border: 1px solid;">birthday</td>  
 <td style="border: 1px solid;">gender</td>  
 <td style="border: 1px solid;">salary</td>  
 <td style="border: 1px solid;">programLanguage</td>  
 </tr>  
 <%  
 for (Employee e : (EmployeeList) request.getAttribute("list")) {  
 %>  
 <tr>  
 <td style="border: 1px solid;"><%=e.getName()%></td>  
 <td style="border: 1px solid;"><%=e.getBirthday()%></td>  
 <td style="border: 1px solid;"><%=e.getGender()%></td>  
 <td style="border: 1px solid;"><%=e.getSalaryStr()%></td>  
 <td style="border: 1px solid;"><%=e.getProgramLanguage()%></td>  
 </tr>  
 <%  
 }  
 %>  
 </table>  
</body>  
</html>

# Додаток Е Код сторінки JavaWEB\_L1\_OTO\_jsp\_jstl.jsp

<%@ **taglib** prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>  
<%@ **taglib** prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>  
<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Tymur KH-221b</title>  
</head>  
<body>  
 <h2>Employees (JSP+EI+JSTL)</h2>  
 <a href="./login">To Main</a>  
 <table>  
 <**c:forEach** var="employee" items="${list}">  
 <tr>  
 <td style="border: 1px solid;">${employee.name}</td>  
 <**c:if** test="${employee.getYears() < 20 || employee.getYears() > 30}">  
 <td style="border: 1px solid;">${employee.birthday}</td>  
 </**c:if**>  
 <**c:if** test="${employee.getYears() >= 20 && employee.getYears() <= 30}">  
 <td style="background-color: darkgreen; border: 1px solid;">${employee.birthday}</td>  
 </**c:if**>  
 <**c:if** test="${employee.gender}">  
 <td style="border: 1px solid;">${employee.getGender()}</td>  
 </**c:if**>  
 <**c:if** test="${!employee.gender}">  
 <td style="background-color: palevioletred; border: 1px solid;">${employee.getGender()}</td>  
 </**c:if**>  
 <td style="border: 1px solid;"><**fmt:formatNumber** type="currency" currencySymbol="uah">${employee.getSalary()}</**fmt:formatNumber**></td>  
 <td style="border: 1px solid;">${employee.programLanguage}</td>  
 </tr>  
 </**c:forEach**>  
 </table>  
</body>  
</html>

# Додаток є Код сторінки JavaWEB\_L1\_OTO.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Tymur KH-221b</title>  
</head>  
<body>  
 <h1 style="font-weight: bold;">Logging in system</h1>  
 <form action="./login" method="post">  
 <label for="login" style="font-weight: bold;">Username</label>  
 <input id="login" name="login" type="text" placeholder="Login"><br>  
 <label for="password" style="font-weight: bold;">Password</label>  
 <input id="password" name="password" type="text" placeholder="Password"><br>  
 <button type="submit" style="background-color: darkgray; border: 1px;">Go to tasks</button>  
 </form>  
</body>  
</html>

# Додаток ж Код класу LoginServlet

package com.example.webappovcharenkolab1task4.servlets;  
  
import java.io.\*;  
import javax.servlet.ServletException;  
import javax.servlet.http.\*;  
import javax.servlet.annotation.\*;  
  
@WebServlet(name = "helloServlet", value = "/login")  
public class LoginServlet extends HttpServlet {  
  
 public void init() {  
 }  
  
 @Override  
 protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 req.getRequestDispatcher("/JavaWEB\_L1\_OTO\_main.jsp").include(req, resp);  
 }  
  
 @Override  
 protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 String login = req.getParameter("login");  
 String password = req.getParameter("password");  
 if (login.equals("user") && password.equals("user")) {  
 doGet(req, resp);  
 }  
 }  
  
 public void destroy() {  
 }  
}

# Додаток З Код класу JspServlet

package com.example.webappovcharenkolab1task4.servlets;  
  
import com.example.webappovcharenkolab1task4.entity.Employee;  
import com.example.webappovcharenkolab1task4.entity.EmployeeList;  
  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
import java.util.List;  
  
@WebServlet("/show-jsp")  
public class JspServlet extends HttpServlet {  
  
 @Override  
 protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 List<Employee> list = EmployeeList.*getInstance*();  
 req.setAttribute("list", list);  
 req.getRequestDispatcher("/JavaWEB\_L1\_OTO\_jsp.jsp").include(req, resp);  
 }  
}

# Додаток И Код класу JspJstlServlet

package com.example.webappovcharenkolab1task4.servlets;  
  
import com.example.webappovcharenkolab1task4.entity.Employee;  
import com.example.webappovcharenkolab1task4.entity.EmployeeList;  
  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
import java.util.List;  
  
@WebServlet("/show-jsp-jstl")  
public class JspJstlServlet extends HttpServlet {  
  
 @Override  
 protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 List<Employee> list = EmployeeList.*getInstance*();  
 req.setAttribute("list", list);  
 req.getRequestDispatcher("/JavaWEB\_L1\_OTO\_jsp\_jstl.jsp").include(req, resp);  
 }  
}