



Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційні систем та технологій

ЛАБОРАТОРНА РОБОТА №3

з дисципліни «Основи програмування - 2»

Тема: «JSP»

Виконали:

студенти групи ІА-31
Клим'юк В.Л, Самелюк А.С,
Дук М.Д, Сакун Д.С

Перевірив:

асистент кафедри ІСТ
Степанов А. С.

Тема: JSP

Мета: Розробка розуміння технології JavaServer Pages (JSP) та її застосування для створення динамічних веб-сторінок у середовищі Java EE.

Хід роботи

1. Пригадати як використовувати JSP для створення динамічних веб-сторінок.
2. Для свого варіанту(№1) з л/р №2 першого семестру створити JSP-сторінку з HTML-формою для введення значень параметрів для обчислення формули (початкове значення, кінцеве значення, крок). При першому відкритті сторінки в полях параметрів мають стояти значення за замовчуванням. При другому та наступних запусках – значення параметрів, що вводились минулого разу. При натисканні кнопки «обчислити» на сторінці має з'явитись таблиця, яка містить значення параметрів та результат обчислення функції. В процесі виконання завдання потрібно продемонструвати використання таких JSP-елементів: scriptlets, expressions, declarations.

```
<%@ page import="com.example.lab_03.FormData" %>
<%@ page import="com.example.lab_03.Result" %>
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%
    FormData formData = (FormData)request.getAttribute("formData");
%>
<%! String title = "Form 1"; %>
<!DOCTYPE html>
<html>
<head>
    <title><%=title%></title>
    <link rel="stylesheet" href="css/form.css">
</head>
<body>
<div class="main-container">
    <div class="image-container">
        
    </div>
    <form class="calculator-container" method="post">
        <div class="calculator-field">
            <div class="calculator-field-name">a</div>
            <div class="calculator-field-from">from:</div>
            <input class="calculator-field-input-from" type="number"
value="<%=formData.getFromA()%>" step="0.01" name="fromA"/>
            <div class="calculator-field-to">to:</div>
            <input class="calculator-field-input-to" type="number"
value="<%=formData.getToA()%>" step="0.01" name="toA"/>
        </div>
    </form>
</div>
</body>
</html>
```

```

        <div class="calculator-field-step">step:</div>
        <input class="calculator-field-input-step" type="number"
value="<%=formData.getStepA() %>" step="0.01" name="stepA"/>
        </div>
        <div class="calculator-field">
            <div class="calculator-field-name">b</div>
            <div class="calculator-field-from">from:</div>
            <input class="calculator-field-input-from" type="number"
value="<%=formData.getFromB() %>" step="0.01" name="fromB"/>
            <div class="calculator-field-to">to:</div>
            <input class="calculator-field-input-to" type="number"
value="<%=formData.getToB() %>" step="0.01" name="toB"/>
            <div class="calculator-field-step">step:</div>
            <input class="calculator-field-input-step" type="number"
value="<%=formData.getStepB() %>" step="0.01" name="stepB"/>
            </div>
            <div class="calculator-field">
                <div class="calculator-field-name">c</div>
                <div class="calculator-field-from">from:</div>
                <input class="calculator-field-input-from" type="number"
value="<%=formData.getFromC() %>" step="0.01" name="fromC"/>
                <div class="calculator-field-to">to:</div>
                <input class="calculator-field-input-to" type="number"
value="<%=formData.getToC() %>" step="0.01" name="toC"/>
                <div class="calculator-field-step">step:</div>
                <input class="calculator-field-input-step" type="number"
value="<%=formData.getStepC() %>" step="0.01" name="stepC"/>
                </div>
                <div class="calculator-field">
                    <div class="calculator-field-name">d</div>
                    <div class="calculator-field-from">from:</div>
                    <input class="calculator-field-input-from" type="number"
value="<%=formData.getFromD() %>" step="0.01" name="fromD"/>
                    <div class="calculator-field-to">to:</div>
                    <input class="calculator-field-input-to" type="number"
value="<%=formData.getToD() %>" step="0.01" name="toD"/>
                    <div class="calculator-field-step">step:</div>
                    <input class="calculator-field-input-step" type="number"
value="<%=formData.getStepD() %>" step="0.01" name="stepD"/>
                    </div>
                    <div class="calculator-field-submit">
                        <button class="calculator-submit-button" type="submit">
                            Calculate
                        </button>
                    </div>
                </div>
            </form>
            <div class="results-container">
                <table>
                    <thead>
                        <tr>
                            <th>a</th>
                            <th>b</th>
                            <th>c</th>
                            <th>d</th>
                            <th>res</th>
                        </tr>
                    </thead>
                    <tbody>
                        <%
for(Result res : formData.getResults()){
%>
                        <tr>
                            <td><%= String.format("%.2f", res.getA()) %></td>
                            <td><%= String.format("%.2f", res.getB()) %></td>
                            <td><%= String.format("%.2f", res.getC()) %></td>

```

```

                <td><%= String.format("%.2f", res.getD()) %></td>
                <td><%= res.getRes() %></td>
            </tr>
        <%
        }
        %>
    </tbody>
</table>
</div>
</div>
</body>
</html>

```

Код 1.1 – form1-page.jsp

```

<%@ page import="com.example.lab_03.FormData" %>
<%@ page import="com.example.lab_03.Result" %>
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%
    FormData formData = (FormData)request.getAttribute("formData");
%>
<%! String title = "Form 2"; %>
<!DOCTYPE html>
<html>
<head>
    <title><%=title%></title>
    <link rel="stylesheet" href="css/form.css">
</head>
<body>
<div class="main-container">
    <div class="image-container">
        
    </div>
    <form class="calculator-container" method="post">
        <div class="calculator-field">
            <div class="calculator-field-name">a</div>
            <div class="calculator-field-from">from:</div>
            <input class="calculator-field-input-from" type="number"
value="<%=formData.getFromA()%>" step="0.01" name="fromA"/>
            <div class="calculator-field-to">to:</div>
            <input class="calculator-field-input-to" type="number"
value="<%=formData.getToA()%>" step="0.01" name="toA"/>
            <div class="calculator-field-step">step:</div>
            <input class="calculator-field-input-step" type="number"
value="<%=formData.getStepA()%>" step="0.01" name="stepA"/>
        </div>
        <div class="calculator-field">
            <div class="calculator-field-name">b</div>
            <div class="calculator-field-from">from:</div>
            <input class="calculator-field-input-from" type="number"
value="<%=formData.getFromB()%>" step="0.01" name="fromB"/>
            <div class="calculator-field-to">to:</div>
            <input class="calculator-field-input-to" type="number"
value="<%=formData.getToB()%>" step="0.01" name="toB"/>
            <div class="calculator-field-step">step:</div>
            <input class="calculator-field-input-step" type="number"
value="<%=formData.getStepB()%>" step="0.01" name="stepB"/>
        </div>
        <div class="calculator-field">
            <div class="calculator-field-name">c</div>
            <div class="calculator-field-from">from:</div>
            <input class="calculator-field-input-from" type="number"

```

```

value="<%=formData.getFromC()%>" step="0.01" name="fromC"/>
    <div class="calculator-field-to">to:</div>
    <input class="calculator-field-input-to" type="number"
value="<%=formData.getToC()%>" step="0.01" name="toC"/>
    <div class="calculator-field-step">step:</div>
    <input class="calculator-field-input-step" type="number"
value="<%=formData.getStepC()%>" step="0.01" name="stepC"/>
    </div>
    <div class="calculator-field">
        <div class="calculator-field-name">d</div>
        <div class="calculator-field-from">from:</div>
        <input class="calculator-field-input-from" type="number"
value="<%=formData.getFromD()%>" step="0.01" name="fromD"/>
        <div class="calculator-field-to">to:</div>
        <input class="calculator-field-input-to" type="number"
value="<%=formData.getToD()%>" step="0.01" name="toD"/>
        <div class="calculator-field-step">step:</div>
        <input class="calculator-field-input-step" type="number"
value="<%=formData.getStepD()%>" step="0.01" name="stepD"/>
    </div>
    <div class="calculator-field-submit">
        <button class="calculator-submit-button" type="submit">
            Calculate
        </button>
    </div>
</form>
<div class="results-container">
    <table>
        <thead>
            <tr>
                <th>a</th>
                <th>b</th>
                <th>c</th>
                <th>d</th>
                <th>res</th>
            </tr>
        </thead>
        <tbody>
            <%
                for(Result res : formData.getResults()){
            <%
            <tr>
                <td><%= res.getA() %></td>
                <td><%= res.getB() %></td>
                <td><%= res.getC() %></td>
                <td><%= res.getD() %></td>
                <td><%= res.getRes() %></td>
            </tr>
            <%
                }
            <%
            </tbody>
        </table>
    </div>
</div>
</body>
</html>

```

Код 1.2 – form2-page.jsp

```

<%@ page import="com.example.lab_03.FormData" %>
<%@ page import="com.example.lab_03.Result" %>
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>

```

```

<%
    FormData formData = (FormData)request.getAttribute("formData");
%>
<%! String title = "Form 3"; %>
<!DOCTYPE html>
<html>
<head>
    <title><%=title%></title>
    <link rel="stylesheet" href="css/form.css">
</head>
<body>
<div class="main-container">
    <div class="image-container">
        
    </div>
    <form class="calculator-container" method="post">
        <div class="calculator-field">
            <div class="calculator-field-name">a</div>
            <div class="calculator-field-from">from:</div>
            <input class="calculator-field-input-from" type="number"
value="<%=formData.getFromA()%>" step="0.01" name="fromA"/>
            <div class="calculator-field-to">to:</div>
            <input class="calculator-field-input-to" type="number"
value="<%=formData.getToA()%>" step="0.01" name="toA"/>
            <div class="calculator-field-step">step:</div>
            <input class="calculator-field-input-step" type="number"
value="<%=formData.getStepA()%>" step="0.01" name="stepA"/>
            </div>
            <div class="calculator-field">
                <div class="calculator-field-name">b</div>
                <div class="calculator-field-from">from:</div>
                <input class="calculator-field-input-from" type="number"
value="<%=formData.getFromB()%>" step="0.01" name="fromB"/>
                <div class="calculator-field-to">to:</div>
                <input class="calculator-field-input-to" type="number"
value="<%=formData.getToB()%>" step="0.01" name="toB"/>
                <div class="calculator-field-step">step:</div>
                <input class="calculator-field-input-step" type="number"
value="<%=formData.getStepB()%>" step="0.01" name="stepB"/>
            </div>
            <div class="calculator-field">
                <div class="calculator-field-name">c</div>
                <div class="calculator-field-from">from:</div>
                <input class="calculator-field-input-from" type="number"
value="<%=formData.getFromC()%>" step="0.01" name="fromC"/>
                <div class="calculator-field-to">to:</div>
                <input class="calculator-field-input-to" type="number"
value="<%=formData.getToC()%>" step="0.01" name="toC"/>
                <div class="calculator-field-step">step:</div>
                <input class="calculator-field-input-step" type="number"
value="<%=formData.getStepC()%>" step="0.01" name="stepC"/>
            </div>
            <div class="calculator-field">
                <div class="calculator-field-name">d</div>
                <div class="calculator-field-from">from:</div>
                <input class="calculator-field-input-from" type="number"
value="<%=formData.getFromD()%>" step="0.01" name="fromD"/>
                <div class="calculator-field-to">to:</div>
                <input class="calculator-field-input-to" type="number"
value="<%=formData.getToD()%>" step="0.01" name="toD"/>
                <div class="calculator-field-step">step:</div>
                <input class="calculator-field-input-step" type="number"
value="<%=formData.getStepD()%>" step="0.01" name="stepD"/>
            </div>
        </form>
    </div>
</div>

```

```

        <div class="calculator-field-submit">
            <button class="calculator-submit-button" type="submit">
                Calculate
            </button>
        </div>
    </form>
    <div class="results-container">
        <table>
            <thead>
                <tr>
                    <th>a</th>
                    <th>b</th>
                    <th>c</th>
                    <th>d</th>
                    <th>res</th>
                </tr>
            <tbody>
                <%
                    for(Result res : formData.getResults()){
                %>
                <tr>
                    <td><%= res.getA() %></td>
                    <td><%= res.getB() %></td>
                    <td><%= res.getC() %></td>
                    <td><%= res.getD() %></td>
                    <td><%= res.getRes() %></td>
                </tr>
                <%
                    }
                %>
            </tbody>
        </table>
    </div>
</div>
</body>
</html>

```

Код 1.3 – form3-page.jsp

```

package com.example.lab_03;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.Cookie;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;
import java.security.InvalidKeyException;
import java.util.ArrayList;
import java.util.Base64;

public abstract class AbstractVladServlet extends HttpServlet{
    protected String _path = "";
    protected String _formFile = "";

    public void init() {

    }

    //Calculate
    public abstract float calculate(float a, float b, float c, float d);

```

```

//Forward request to jsp file
public void forwardToJspFile(HttpServletRequest request, HttpServletResponse
response) throws IOException, ServletException {
    request.getRequestDispatcher(_formFile).forward(request, response);
}

//GET handler
public void doGet(
    HttpServletRequest request,
    HttpServletResponse response
) throws IOException, ServletException
{
    try{
        FormData data = createFormDataFromCookie(request);
        request.setAttribute("formData", data);
    }catch (InvalidKeyException exc){
        response.sendError(400);
        return;
    }
    forwardToJspFile(request, response);
}

//POST handler
public void doPost(
    HttpServletRequest request,
    HttpServletResponse response
) throws IOException, ServletException
{
    FormData data = createFormDataFromRequest(request);
    fillResults(data);
    request.setAttribute("formData", data);
    writeCookie(data, response);
    forwardToJspFile(request, response);
}

private void fillResults(FormData formData){
    float epsilon = 0.00001F;
    ArrayList<Result> results = formData.getResults();

    for(float a = formData.getFromA(); a - formData.getToA() < epsilon; a +=
formData.getStepA()){
        for(float b = formData.getFromB(); b - formData.getToB() < epsilon; b +=
formData.getStepB()){
            for(float c = formData.getFromC(); c - formData.getToC() < epsilon; c
+= formData.getStepC()){
                for(float d = formData.getFromD(); d - formData.getToD() <
epsilon; d += formData.getStepD()){
                    Result newResult = new Result();
                    newResult.setA(a);
                    newResult.setB(b);
                    newResult.setC(c);
                    newResult.setD(d);
                    newResult.setRes(this.calculate(a,b,c,d));
                    results.add(newResult);
                }
            }
        }
    }
}

private FormData createFormDataFromRequest(HttpServletRequest request){
    FormData res = new FormData();

```



```

        res.setFromA(Float.parseFloat(request.getParameter("fromA")));
        res.setToA(Float.parseFloat(request.getParameter("toA")));
        res.setStepA(Float.parseFloat(request.getParameter("stepA")));
        res.setFromB(Float.parseFloat(request.getParameter("fromB")));
        res.setToB(Float.parseFloat(request.getParameter("toB")));
        res.setStepB(Float.parseFloat(request.getParameter("stepB")));
        res.setFromC(Float.parseFloat(request.getParameter("fromC")));
        res.setToC(Float.parseFloat(request.getParameter("toC")));
        res.setStepC(Float.parseFloat(request.getParameter("stepC")));
        res.setFromD(Float.parseFloat(request.getParameter("fromD")));
        res.setToD(Float.parseFloat(request.getParameter("toD")));
        res.setStepD(Float.parseFloat(request.getParameter("stepD")));

        return res;
    }

    private FormData createFormDataFromCookie(HttpServletRequest request) throws
InvalidKeyException {
        FormData res = new FormData();

        for (var c : request.getCookies()) {
            if (c.getName().equals("formData")) {
                String params = new String(Base64.getDecoder().decode(c.getValue()));

                String[] splitParams = params.split(";");
                for (var p : splitParams) {
                    String[] keyValue = p.split("=");
                    switch (keyValue[0]) {
                        case "fromA":    res.setFromA(Float.parseFloat(keyValue[1]));
break;
                        case "toA":      res.setToA(Float.parseFloat(keyValue[1]));
break;
                        case "stepA":    res.setStepA(Float.parseFloat(keyValue[1]));
break;
                        case "fromB":    res.setFromB(Float.parseFloat(keyValue[1]));
break;
                        case "toB":      res.setToB(Float.parseFloat(keyValue[1]));
break;
                        case "stepB":    res.setStepB(Float.parseFloat(keyValue[1]));
break;
                        case "fromC":    res.setFromC(Float.parseFloat(keyValue[1]));
break;
                        case "toC":      res.setToC(Float.parseFloat(keyValue[1]));
break;
                        case "stepC":    res.setStepC(Float.parseFloat(keyValue[1]));
break;
                        case "fromD":    res.setFromD(Float.parseFloat(keyValue[1]));
break;
                        case "toD":      res.setToD(Float.parseFloat(keyValue[1]));
break;
                        case "stepD":    res.setStepD(Float.parseFloat(keyValue[1]));
break;
                        default: throw new InvalidKeyException("Invalid parameter in
cookie data");
                    }
                }
                return res;
            }
        }
        return res;
    }

    private void writeCookie(FormData data, HttpServletResponse response) {
        StringBuilder builder = new StringBuilder();

```

```

        builder.append("fromA=");    builder.append(data.getFromA());
builder.append(';');
        builder.append("toA=");    builder.append(data.getToA());
builder.append(';');
        builder.append("stepA=");  builder.append(data.getStepA());
builder.append(';');
        builder.append("fromB=");  builder.append(data.getFromB());
builder.append(';');
        builder.append("toB=");    builder.append(data.getToB());
builder.append(';');
        builder.append("stepB=");  builder.append(data.getStepB());
builder.append(';');
        builder.append("fromC=");  builder.append(data.getFromC());
builder.append(';');
        builder.append("toC=");    builder.append(data.getToC());
builder.append(';');
        builder.append("stepC=");  builder.append(data.getStepC());
builder.append(';');
        builder.append("fromD=");  builder.append(data.getFromD());
builder.append(';');
        builder.append("toD=");    builder.append(data.getToD());
builder.append(';');
        builder.append("stepD=");  builder.append(data.getStepD());

        Cookie cookie = new Cookie("formData",
Base64.getEncoder().encodeToString(builder.toString().getBytes()));
        cookie.setPath(this._path);
        cookie.setHttpOnly(true);
        cookie.setSecure(true);
        cookie.setMaxAge(60 * 60 * 24);

        response.addCookie(cookie);
    }

    public void destroy() {
    }
}

```

Код 1.4 – AbstractVladServlet

```

package com.example.lab_03;
import jakarta.servlet.annotation.WebServlet;

@WebServlet(name = "vladServlet1", value = "/form1")
public class VladServlet1 extends AbstractVladServlet {

    public VladServlet1() {
        _formFile = "form1-page.jsp";
        _path = "/lab2_3_war_exploded/form1";
    }

    @Override
    public float calculate(float a, float b, float c, float d) {
        return (float) Math.sqrt(Math.abs(Math.sin(a) - 4.0F * Math.log(b) /
Math.pow(c, d)));
    }
}

```

Код 1.5 – VladServlet1

```

package com.example.lab_03;
import jakarta.servlet.annotation.WebServlet;

@WebServlet(name = "vladServlet2", value = "/form2")
public class VladServlet2 extends AbstractVladServlet {
    public VladServlet2() {
        _formFile = "form2-page.jsp";
        _path = "/lab2_3_war_exploded/form2";
    }

    @Override
    public float calculate(float a, float b, float c, float d) {
        return (float) ((Math.exp(a) + 3 * Math.log10(c) * Math.abs(Math.atan(d))) /
(Math.pow(b, 1 / c)));
    }
}

```

Код 1.6 – VladServlet2

```

package com.example.lab_03;
import jakarta.servlet.annotation.WebServlet;

@WebServlet(name = "vladServlet3", value = "/form3")
public class VladServlet3 extends AbstractVladServlet {
    public VladServlet3() {
        _formFile = "form3-page.jsp";
        _path = "/lab2_3_war_exploded/form3";
    }

    @Override
    public float calculate(float a, float b, float c, float d) {
        return (float) Math.pow(2.0 * Math.sin(a) + Math.cos(Math.abs(b *
Math.sqrt(c))), d);
    }
}

```

Код 1.7 – VladServlet3

```

package com.example.lab_03;

import java.util.ArrayList;

public class FormData {
    public float getFromA() {
        return fromA;
    }

    public void setFromA(float fromA) {
        this.fromA = fromA;
    }

    public float getToA() {
        return toA;
    }

    public void setToA(float toA) {
        this.toA = toA;
    }

    public float getStepA() {
        return stepA;
    }
}

```

```

    }

    public void setStepA(float stepA) {
        this.stepA = stepA;
    }

    public float getFromB() {
        return fromB;
    }

    public void setFromB(float fromB) {
        this.fromB = fromB;
    }

    public float getToB() {
        return toB;
    }

    public void setToB(float toB) {
        this.toB = toB;
    }

    public float getStepB() {
        return stepB;
    }

    public void setStepB(float stepB) {
        this.stepB = stepB;
    }

    public float getFromC() {
        return fromC;
    }

    public void setFromC(float fromC) {
        this.fromC = fromC;
    }

    public float getToC() {
        return toC;
    }

    public void setToC(float toC) {
        this.toC = toC;
    }

    public float getStepC() {
        return stepC;
    }

    public void setStepC(float stepC) {
        this.stepC = stepC;
    }

    public float getFromD() {
        return fromD;
    }

    public void setFromD(float fromD) {
        this.fromD = fromD;
    }

    public float getToD() {
        return toD;
    }

```

```

    }

    public void setToD(float toD) {
        this.toD = toD;
    }

    public float getStepD() {
        return stepD;
    }

    public void setStepD(float stepD) {
        this.stepD = stepD;
    }

    public ArrayList<Result> getResults() {
        return results;
    }

    private float fromA;
    private float toA;
    private float stepA;
    private float fromB;
    private float toB;
    private float stepB;
    private float fromC;
    private float toC;
    private float stepC;
    private float fromD;
    private float toD;
    private float stepD;
    private final ArrayList<Result> results = new ArrayList<>();
}

```

Код 1.8 – FormData

```

package com.example.lab_03;

public class Result {
    public float getA() {
        return a;
    }

    public void setA(float a) {
        this.a = a;
    }

    public float getB() {
        return b;
    }

    public void setB(float b) {
        this.b = b;
    }

    public float getC() {
        return c;
    }

    public void setC(float c) {
        this.c = c;
    }
}

```

```

public float getD() {
    return d;
}

public void setD(float d) {
    this.d = d;
}

public float getRes() {
    return res;
}

public void setRes(float res) {
    this.res = res;
}

private float a;
private float b;
private float c;
private float d;
private float res;
}

```

Код 1.9 – Result

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Lab 3</title>
</head>
<body>
<p><a href="./form1">Formula 1</a></p>
<p><a href="./form2">Formula 2</a></p>
<p><a href="./form3">Formula 3</a></p>
</body>
</html>

```

Код 1.10 – Реалізація головної сторінки

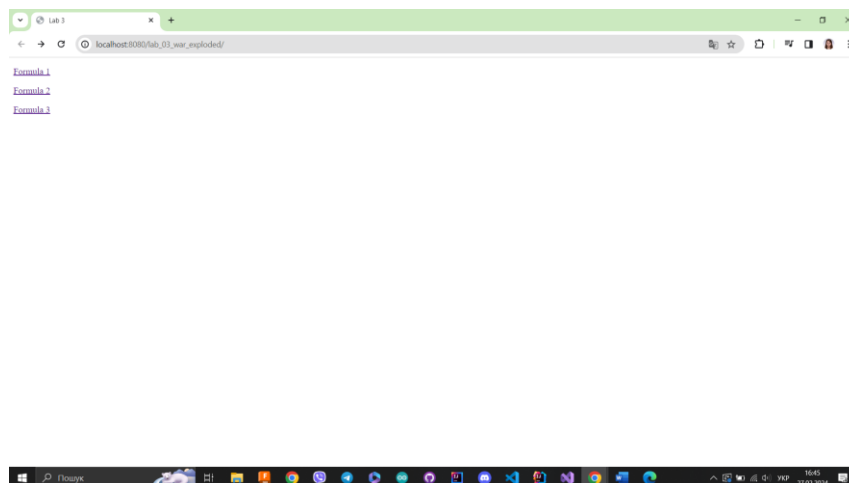


Рисунок 1.1 – Головна сторінка сайту

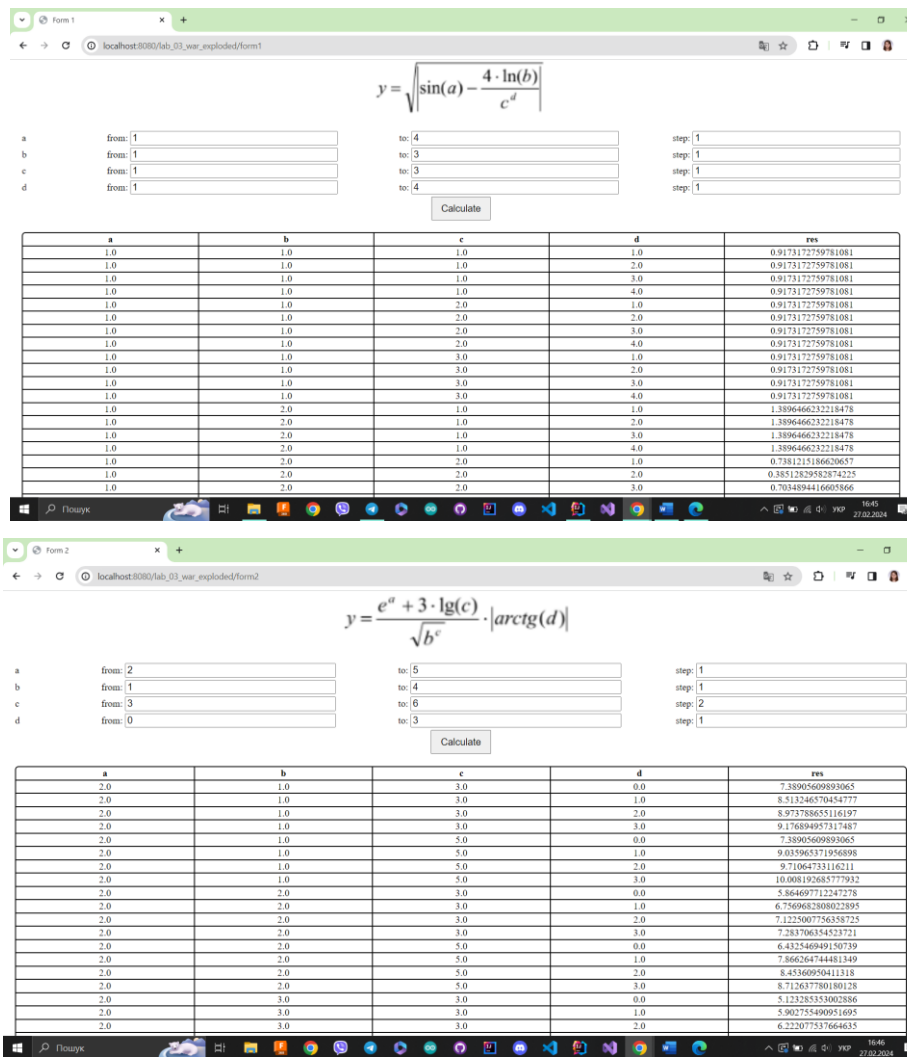


Рисунок 1.2(3) – Приклади інших сторінок сайту

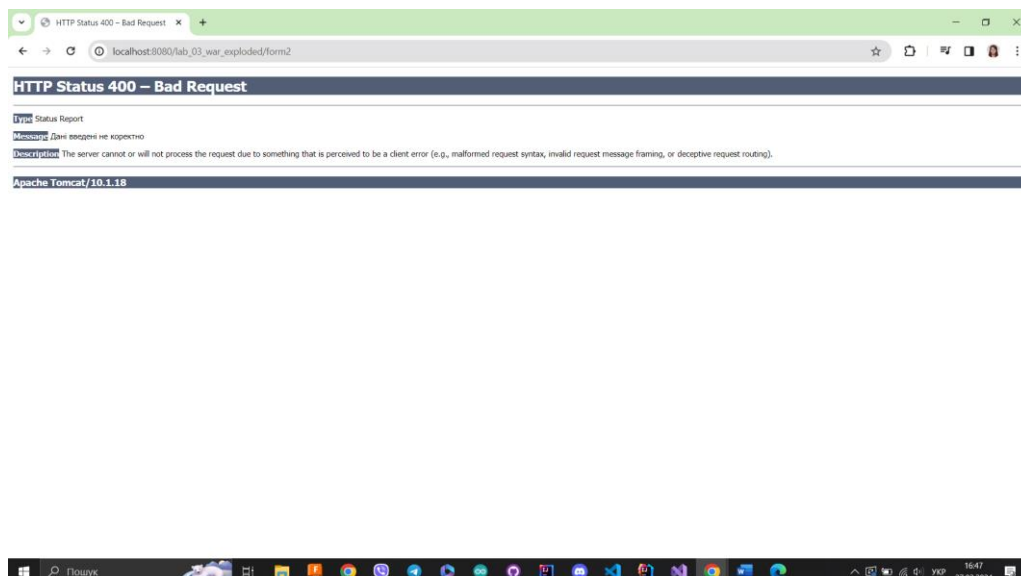


Рисунок 1.4 – Повернення помилки на некоректно введені дані

3. Відповісти на контрольні питання.

Висновки: У процесі виконання лабораторної роботи ми успішно ознайомились з технологією JavaServer Pages (JSP) та її можливостями для створення динамічних веб-сторінок. Розуміння принципів роботи JSP дозволить нам більш ефективно розробляти веб-застосунки у майбутньому.