

Documents Autorisés J2EE

Contact App :

index.jsp :

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>

<!DOCTYPE html>

<html>

<head>

    <title>JSP Page</title>

    <script>

        function confirmDelete() {

            return confirm("Are you sure you want to delete this item?");

        }

    </script>

</head>

<body>

    <!-- Formulaire d'ajout -->

    <form action="ContactServlet" method="POST">

        <input name="nom" placeholder="entrez votre nom" />

        <br>

        <input type="email" name="email" placeholder="entrez votre email" />

        <br>

        <input type="tel" name="tel" placeholder="entrez votre numéro de téléphone" />

        <br>

        <input type="submit" value="ajouter" />

    </form>

    <!-- Boutons de tri -->

    <form action="ContactServlet" method="POST">
```

```

<input type="hidden" name="action" value="A-Z">
<input type="submit" value="A-Z"/>
</form>

<form action="ContactServlet" method="POST">
  <input type="hidden" name="action" value="Z-A">
  <input type="submit" value="Z-A"/>
</form>

<!-- Message d'erreur -->
<c:if test="${not empty sessionScope.error}">
  <script>alert('${sessionScope.error}');

```

```

        <input type="hidden" name="itemId" value="${status.index}" />
        <input type="hidden" name="action" value="delete" />
        <input type="submit" value="delete" />
    </form>
</td>
</tr>
</c:forEach>
</table>

</body>
</html>

<%--
    Document : index
    Created on : 31 Mar 2025, 21:52:04
    Author : farou
--%>

<%-- <%@page import="java.util.List"%>
<%@page import="models.Contact"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
    <script>
        function confirmDelete() {
            return confirm("Are you sure you want to delete this item?");
        }
    </script>

```

```

</head>
<body>
  <form action="ContactServlet" method="POST">
    <input name="nom" placeholder="entrez votre nom" />
    <br>
    <input type="email" name="email" placeholder="entrez votre email" />
    <br>
    <input type="tel" name="tel" placeholder="entrez votre numero de téléphone" />
    <br>
    <input type="submit" value="ajouter" />
  </form>

  <%
    out.print("<form action='ContactServlet' method='POST'>"
      + "<input type='hidden' name='action' value='A-Z'>"
      + "<input type='submit' value='A-Z' />"
      + "</form>");
    out.print("<form action='ContactServlet' method='POST'>"
      + "<input type='hidden' name='action' value='Z-A'>"
      + "<input type='submit' value='Z-A' />"
      + "</form>");
  %>

  <table>
    <tr>
      <th>Nom</th>
      <th>Email</th>
      <th>Numéro de téléphone</th>
      <th>Actions</th>
    </tr>
  <%

```

```

HttpSession s = request.getSession(false);

if(s!=null){

    List<Contact> contacts = (List<Contact>)s.getAttribute("contacts");

    if(contacts!=null){

        if(s.getAttribute("error")!=null){

            String message = (String)s.getAttribute("error");

            out.print("<script>alert('"+message+"')</script>");

            s.removeAttribute("error");

        }

        for(int i = 0;i<contacts.size();i++){

            out.print("<tr>");

            out.print("<td>"+contacts.get(i).getNom()+"</td>");

            out.print("<td>"+contacts.get(i).getEmail()+"</td>");

            out.print("<td>"+contacts.get(i).getTel()+"</td>");

            out.println("<td>"

+ "<a href='modifier.jsp?email="+contacts.get(i).getEmail()+">Modifier</a>"

+ "<form action='ContactServlet' onsubmit='return confirmDelete();' method='POST'>"

+ "<input type='hidden' name='itemId' value='"+i+"' />"

+ "<input type='hidden' name='action' value='delete' />"

+ "<input type='submit' value='delete' />"

+ "</form>"

+ "</td>");

            out.print("</tr>");

        }

    }

}

%>

</table>

</body>

</html> --%>

```

Modifier.jsp :

```
<%--
```

```
Document : modifier
```

```
Created on : 31 Mar 2025, 23:45:36
```

```
Author : farou
```

```
--%>
```

```
<%@page import="java.util.List"%>
```

```
<%@page import="models.Contact"%>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
<title>JSP Page</title>
```

```
</head>
```

```
<body>
```

```
<%
```

```
HttpSession s = request.getSession(false);
```

```
String email = request.getParameter("email");
```

```
Contact c = null;
```

```
if(s!=null){
```

```
List<Contact> contacts = (List<Contact>)s.getAttribute("contacts");
```

```
if(contacts!=null && !email.isEmpty()){
```

```
for(Contact cont : contacts){
```

```
if(cont.getEmail().equals(email)){
```

```
c = cont;
```

```
break;
```

```
}
```

```
}
```

```

    }

}

%>

<form action="ContactServlet" method="POST">

    <input type="hidden" name="uniqueKey" <% out.print("value="+c.getEmail()); %> />

    <input type="hidden" name="action" value="modify"/>

    <input name="nom" placeholder="entrez votre nom" <%= "value="+c.getNom()%>/>

    <br>

    <input type="email" name="email" placeholder="entrez votre email" <% out.print("value="+c.getEmail()); %> />

    <br>

    <input type="tel" name="tel" placeholder="entrez votre numero de téléphone" <%
out.print("value="+c.getTel()); %> />

    <br>

    <input type="submit" value="modifier" />

</form>

</body>

</html>

<%-- <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<!DOCTYPE html>

<html>

<head>

    <title>JSP Page</title>

</head>

<body>

<c:set var="email" value="${param.email}" />

<c:set var="contactToEdit" value="${null}" />

```

```

<c:if test="${not empty sessionScope.contacts and not empty email}">

    <c:forEach var="contact" items="${sessionScope.contacts}">

        <c:if test="${contact.email == email}">

            <c:set var="contactToEdit" value="${contact}" />

        </c:if>

    </c:forEach>

</c:if>

<c:if test="${not empty contactToEdit}">

    <form action="ContactServlet" method="POST">

        <input type="hidden" name="uniqueKey" value="${contactToEdit.email}" />

        <input type="hidden" name="action" value="modify" />

        <input name="nom" placeholder="entrez votre nom" value="${contactToEdit.nom}" />

        <br>

        <input type="email" name="email" placeholder="entrez votre email" value="${contactToEdit.email}" />

        <br>

        <input type="tel" name="tel" placeholder="entrez votre numero de téléphone" value="${contactToEdit.tel}" />

        <br>

        <input type="submit" value="modifier" />

    </form>

</c:if>

<c:if test="${empty contactToEdit}">

    <p>Contact introuvable.</p>

</c:if>

</body>

</html>

--%>

```

Servlet.java :


```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to edit this template
 */
package controllers;

import java.io.IOException;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;
import java.util.ArrayList;
import java.util.Comparator;
import java.util.List;
import models.Contact;

/**
 *
 * @author farou
 */
@WebServlet("/ContactServlet")
public class ContactServlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        HttpSession s = request.getSession(false);
        if(s==null){

```

```

s = request.getSession();
List<Contact> contacts = new ArrayList<>();
s.setAttribute("contacts", contacts);
}
request.getRequestDispatcher("index.jsp").forward(request, response);
}

```

@Override

```

protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

```

```

    HttpSession s = request.getSession();
    List<Contact> contacts = (List<Contact>)s.getAttribute("contacts");

```

```

    if(contacts==null){
        contacts = new ArrayList<>();
        s.setAttribute("contacts", contacts);
    }

```

```

    String action = request.getParameter("action");

```

```

    if("A-Z".equals(action)){

```

```

        contacts.sort(Comparator.comparing(Contact::getNom));
        s.setAttribute("contacts", contacts);
        response.sendRedirect("index.jsp");
        return;

```

```

    }else if("Z-A".equals(action)){

```

```

contacts.sort(Comparator.comparing(Contact::getNom).reversed());
s.setAttribute("contacts", contacts);
response.sendRedirect("index.jsp");
return;
}

```

```

if ("delete".equals(action)) {
    String indexToDelete = request.getParameter("itemId");
    if (indexToDelete != null) {
        int index = Integer.parseInt(indexToDelete);
        if (index >= 0 && index < contacts.size()) {
            contacts.remove(index);
            s.setAttribute("contacts", contacts);
        }
    }
    response.sendRedirect("index.jsp");
    return;
}

```

```

String nom = request.getParameter("nom");
String email = request.getParameter("email");
String tel = request.getParameter("tel");

if ("modify".equals(action)) {
    if (nom == null || nom.isEmpty() || email == null || email.isEmpty()) {
        s.setAttribute("error", "You have to fill all fields");
    } else {
        String emailUnique = request.getParameter("uniqueKey");

```

```

        for (int i = 0; i < contacts.size(); i++) {
            if (contacts.get(i).getEmail().equals(emailUnique)) {
                Contact updatedContact = new Contact(nom, email, tel);
                contacts.set(i, updatedContact);
                s.setAttribute("contacts", contacts);
                break;
            }
        }
    }
} else {
    // Default action (add new contact)
    if(nom == null || nom.isEmpty() || email == null || email.isEmpty()){
        s.setAttribute("error", "Name and email are required");
    } else {
        Contact c = tel == null || tel.isEmpty()
            ? new Contact(nom, email)
            : new Contact(nom, email, tel);
        contacts.add(c);
        s.setAttribute("contacts", contacts);
    }
}

response.sendRedirect("index.jsp");
}
}

```

Controle de mi-semestre :

Accueil.jsp 👍

<%--

Document : Accueil

Created on : 30 Mar 2025, 01:38:06

Author : farou

--%>

```
<%@page import="java.time.LocalDate"%>
```

```
<%@page import="java.util.List"%>
```

```
<%@page import="model.Task"%>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
<title>JSP Page</title>
```

```
</head>
```

```
<body>
```

```
<fieldset style="width: 350px;">
```

```
<legend>Ajout d'une tache</legend>
```

```
<form action="TaskServlet" method="POST">
```

```
Nom de la tache : <input name="name"/>
```

```
<br>
```

```
Description :
```

```
<textarea name="description">
```

```
</textarea>
```

```
<br>
```

```
Délai de fin : <input type="date" name="delai" />
```

```
<br>
```

```
<input type="submit" value="ajouter" />
```

```
</form>
```

```
</fieldset>
```

```
<h2>La liste des taches</h2>
```

```
<%
```

```

HttpSession s = request.getSession(false);
if(s!=null){
    if(s.getAttribute("error")!=null){
        String m =(String) s.getAttribute("error");
        out.print("<div style='color:red;font-weight:bold;'>" + m + "</div>");
        s.removeAttribute("error");
    }
    List<Task> tasks = (List<Task>)s.getAttribute("tasks");
    //Boolean isFinishedObj = (Boolean) s.getAttribute("isFinished");
    //boolean isFinished = (isFinishedObj != null) ? isFinishedObj : false;

    if (tasks != null) {
        for (int i = 0; i < tasks.size(); i++) {
            out.print("<fieldset style='color:" + (tasks.get(i).isFinished ? "green" : "red") + ";>"
                + "<legend> Tache N" + (i + 1) + " - " + tasks.get(i).getName() + "</legend>"
                + "<div>" + tasks.get(i).getDescription() + " (" + tasks.get(i).getDelai() + ")</div>"
            );

            if (tasks.get(i).isFinished) {
                out.print("<span>terminée</span>");
            } else {
                out.print("<form action='TaskServlet' method='POST'>"
                    + "<input type='hidden' name='taskIndex' value='" + i + "'>"
                    + "<input type='hidden' name='action' value='finish'>"
                    + "<button type='submit'>Terminer</button>"
                    + "</form>");
            }

            out.print("</fieldset>");
        }
    }
}

```

```

    }
    %>
</body>
</html>

```

```

<%-- <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <title>JSP Page</title>
</head>
<body>

```

```

<fieldset style="width: 350px;">
    <legend>Ajout d'une tache</legend>
    <form action="TaskServlet" method="POST">
        Nom de la tache : <input name="name" />
        <br>
        Description :
        <textarea name="description"></textarea>
        <br>
        Délai de fin : <input type="date" name="delai" />
        <br>
        <input type="submit" value="ajouter" />
    </form>
</fieldset>

```

```

<h2>La liste des taches</h2>

```

```
<!-- Message d'erreur -->
```

```
<c:if test="${not empty sessionScope.error}">
```

```
  <div style="color:red;font-weight:bold;">${sessionScope.error}</div>
```

```
  <c:remove var="error" scope="session" />
```

```
</c:if>
```

```
<!-- Affichage des tâches -->
```

```
<c:if test="${not empty sessionScope.tasks}">
```

```
  <c:forEach var="task" items="${sessionScope.tasks}" varStatus="status">
```

```
    <fieldset style="color: <c:out value='${task.finished ? "green" : "red"}' />";>
```

```
      <legend>Tache N${status.index + 1} - ${task.name}</legend>
```

```
      <div>${task.description} (${task.delai})</div>
```

```
    <c:choose>
```

```
      <c:when test="${task.finished}">
```

```
        <span>terminée</span>
```

```
      </c:when>
```

```
    <c:otherwise>
```

```
      <form action="TaskServlet" method="POST">
```

```
        <input type="hidden" name="taskIndex" value="${status.index}" />
```

```
        <input type="hidden" name="action" value="finish" />
```

```
        <button type="submit">Terminer</button>
```

```
      </form>
```

```
    </c:otherwise>
```

```
  </c:choose>
```

```
</fieldset>
```

```
</c:forEach>
```

```
</c:if>
```

```
</body>
```



```
</html>
```

```
--%>
```

```
Servlet.java:
```

```
/*
```

```
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
```

```
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to edit this template
```

```
*/
```

```
package control;
```

```
import java.io.IOException;
```

```
import java.io.PrintWriter;
```

```
import jakarta.servlet.ServletException;
```

```
import jakarta.servlet.annotation.WebServlet;
```

```
import jakarta.servlet.http.HttpServlet;
```

```
import jakarta.servlet.http.HttpServletRequest;
```

```
import jakarta.servlet.http.HttpServletResponse;
```

```
import jakarta.servlet.http.HttpSession;
```

```
import java.time.LocalDate;
```

```
import java.util.ArrayList;
```

```
import java.util.Comparator;
```

```
import java.util.List;
```

```
import model.Task;
```

```
/**
```

```
 *
```

```
 * @author farou
```

```
 */
```

```
@WebServlet("/TaskServlet")
```

```
public class TaskServlet extends HttpServlet {
```

```
    @Override
```

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    HttpSession s = request.getSession(false);
    if(s==null){
        s = request.getSession();
        List<Task> tasks = new ArrayList<>();
        s.setAttribute("tasks", tasks);
    }
    request.getRequestDispatcher("Acceuil.jsp").forward(request, response);
}
```

@Override

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    HttpSession ses = request.getSession();

    boolean finished = false;

    String name = request.getParameter("name");
    String description = request.getParameter("description");
    String delai = request.getParameter("delai");

    String action = request.getParameter("action"); // Can be null
    String taskIndex = request.getParameter("taskIndex"); // Can be null

    String errorMessage = "You must fill all fields";

    // Check if the action is null or blank
    if (action == null || action.isBlank()) {
```

```

// Handle task addition logic
if (name != null && description != null && delai != null &&
    !name.isEmpty() && !description.isEmpty() && !delai.isEmpty()) {

    LocalDate d = LocalDate.parse(delai);

    if (d.isEqual(LocalDate.now()) || d.isBefore(LocalDate.now())) {
        finished = true;
    }
    ses.setAttribute("isFinished", finished);

    Task task = new Task(name, description, d, finished);

    List<Task> tasks = (List<Task>) ses.getAttribute("tasks");
    if (tasks == null) {
        tasks = new ArrayList<>();
        ses.setAttribute("tasks", tasks);
    }

    tasks.add(task);
    tasks.sort(Comparator.comparing(Task::getDelai));

    ses.setAttribute("tasks", tasks);
} else {
    ses.setAttribute("error", errorMessage);
}
} else if (action.equals("finish") && taskIndex != null) {
    // Handle marking a task as "finished"
    List<Task> tasks = (List<Task>) ses.getAttribute("tasks");
    if (tasks != null) {
        int index = Integer.parseInt(taskIndex);

```

```

        if (index >= 0 && index < tasks.size()) {
            tasks.get(index).setIsFinished(true);
        }
    }
}

request.getRequestDispatcher("Acceuil.jsp").forward(request, response);
}
}

```

Model.java

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package model;

import java.time.LocalDate;

/**
 *
 * @author farou
 */
public class Task {

    public String name;
    public String description;
    public LocalDate delai;
    public boolean isFinished;

    public Task(String name, String description, LocalDate delai, boolean isFinished) {

```

```
this.name = name;  
this.description = description;  
this.delai = delai;  
this.isFinished = isFinished;  
}
```

```
public boolean isIsFinished() {  
    return isFinished;  
}
```

```
public void setIsFinished(boolean isFinished) {  
    this.isFinished = isFinished;  
}
```

```
public String getName() {  
    return name;  
}
```

```
public void setName(String name) {  
    this.name = name;  
}
```

```
public String getDescription() {  
    return description;  
}
```

```
public void setDescription(String description) {  
    this.description = description;  
}
```

```

public LocalDate getDelai() {
    return delai;
}

public void setDelai(LocalDate delai) {
    this.delai = delai;
}
}

```

PanierApp :

Panier.jsp :

```

<%@page import="java.util.List"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Panier</title>
</head>
<body>
    <div style="border: 2px solid blue;padding: 5px;margin-top: 20px">
        <form action="AddArticlesServlet" method="POST" style="display:flex;justify-content:
space-around">
            <div>
                <label style="color: red">Nom de l'article</label>
                <br>
                <input name="name" style="background-color: whitesmoke;border-color: black"/>
            </div>

```

```

<div>
  <label style="color: red">Quantite</label>
  <br>
  <input type="number" name="quantity" style="background-color:
whitesmoke;border-color: black"/>
</div>

<div>
  <label style="color: red">Prix Unitaire</label>
  <br>
  <input type="number" step="0.01" name="unit_price" style="background-color:
whitesmoke;border-color: black"/>
</div>

<div style="align-self:center">
  <input style="color: red;background-color: whitesmoke;border-color: black;border-radius:
5px;" type="submit" value="ajouter"/>
</div>
</form>

<div style="display:flex;flex-direction: column;">
  <label style="color: red">
    Détails de la facture :
  </label>
  <table border="1">
    <thead>
      <tr>
        <th>Article</th>
        <th>Quantité</th>
        <th>Prix Unitaire</th>

```

```

</tr>
</thead>
<tbody>
<%
    HttpSession s = request.getSession(false);
    float newTotal = 0;
    if (s != null) {
        List<String> articles = (List<String>) s.getAttribute("articles");
        if (articles != null) {
            for (int i = 0; i < articles.size(); i += 3) {
                out.print("<tr>"
                    + "<td>" + articles.get(i) + "</td>"
                    + "<td>" + articles.get(i + 1) + "</td>"
                    + "<td>" + articles.get(i + 2) + "</td>"
                    + "</tr>");
            }
        }
    }
    if (request.getAttribute("totalValue") != null) {
        newTotal = (float) request.getAttribute("totalValue");
    } else {
        // If no request attribute, fall back to the cookie value
        Cookie[] cookies = request.getCookies();
        if (cookies != null) {
            for (Cookie c : cookies) {
                if ("total".equals(c.getName())) {
                    newTotal = Float.parseFloat(c.getValue());
                }
            }
        }
    }
%>

```



```

        }
    }
}

%>

</tbody>
</table>

    <label>Total à payer : <%= newTotal %></label>

</div>
</div>
</body>
</html>

<%--
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <title>Panier</title>
</head>
<body>
<div style="border: 2px solid blue;padding: 5px;margin-top: 20px">
    <!-- Formulaire d'ajout -->

    <form action="AddArticlesServlet" method="POST" style="display:flex;justify-content:
space-around">

```

```

<div>
  <label style="color: red">Nom de l'article</label>
  <br>
  <input name="name" style="background-color: whitesmoke;border-color: black"/>
</div>
<div>
  <label style="color: red">Quantite</label>
  <br>
  <input type="number" name="quantity" style="background-color:
whitesmoke;border-color: black"/>
</div>
<div>
  <label style="color: red">Prix Unitaire</label>
  <br>
  <input type="number" step="0.01" name="unit_price" style="background-color:
whitesmoke;border-color: black"/>
</div>
<div style="align-self:center">
  <input style="color: red;background-color: whitesmoke;border-color: black;border-radius:
5px;" type="submit" value="ajouter"/>
</div>
</form>

<!-- Affichage des articles -->
<div style="display:flex;flex-direction: column;">
  <label style="color: red">Détails de la facture :</label>
  <table border="1">
    <thead>

```

```

<tr>
  <th>Article</th>
  <th>Quantité</th>
  <th>Prix Unitaire</th>
</tr>
</thead>
<tbody>
  <c:if test="${not empty sessionScope.articles}">
    <c:set var="articles" value="${sessionScope.articles}" />
    <c:forEach var="i" begin="0" end="${fn:length(articles) - 1}" step="3">
      <tr>
        <td>${articles[i]}</td>
        <td>${articles[i + 1]}</td>
        <td>${articles[i + 2]}</td>
      </tr>
    </c:forEach>
  </c:if>
</tbody>
</table>

<!-- Affichage du total -->
<c:set var="totalValue" value="${requestScope.totalValue}" />
<c:choose>
  <c:when test="${not empty totalValue}">
    <label>Total à payer : ${totalValue}</label>
  </c:when>
  <c:otherwise>

```

```

    <c:set var="newTotal" value="0" />
    <c:forEach var="cookie" items="${pageContext.request.cookies}">
        <c:if test="${cookie.name == 'total'}">
            <c:set var="newTotal" value="${cookie.value}" />
        </c:if>
    </c:forEach>
    <label>Total à payer : ${newTotal}</label>
</c:otherwise>
</c:choose>
</div>
</div>
</body>
</html>
--%>

```

Servlet.java :

```

package controller;

import java.io.IOException;
import java.util.*;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.Cookie;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;

```

```
@WebServlet("/AddArticlesServlet")
```

```
public class AddArticlesServlet extends HttpServlet {
```

```
    @Override
```

```
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
```

```
        throws ServletException, IOException {
```

```
        HttpSession s = request.getSession(false);
```

```
        if (s == null) {
```

```
            s = request.getSession();
```

```
            List<String> articles = new ArrayList<>();
```

```
            s.setAttribute("articles", articles);
```

```
        }
```

```
        request.getRequestDispatcher("Panier.jsp").forward(request, response);
```

```
    }
```

```
    @Override
```

```
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
        throws ServletException, IOException {
```

```
        String nom_article = request.getParameter("name");
```

```
        String quantity_string = request.getParameter("quantity");
```

```
        String unit_price = request.getParameter("unit_price");
```

```
        HttpSession s = request.getSession();
```

```
        List<String> articles = (List<String>) s.getAttribute("articles");
```

```

if (articles == null) {
    articles = new ArrayList<>();
    s.setAttribute("articles", articles);
}

```

```

float totalPrice = 0;
if (quantity_string != null && unit_price != null) {
    int quantity = Integer.parseInt(quantity_string);
    float UP = Float.parseFloat(unit_price);
    totalPrice = quantity * UP;
}

```

```

articles.add(nom_article);
articles.add(String.valueOf(quantity));
articles.add(String.valueOf(UP));
}

```

```

// Retrieve existing total from cookies
float existingTotal = 0;
Cookie[] cookies = request.getCookies();
if (cookies != null) {
    for (Cookie c : cookies) {
        if ("total".equals(c.getName())) {
            existingTotal = Float.parseFloat(c.getValue());
        }
    }
}
}

```

```

// Update total value
float newTotal = existingTotal + totalPrice;

// Store the updated total in a cookie
Cookie total_cookie = new Cookie("total", String.valueOf(newTotal));
total_cookie.setMaxAge(60 * 60 * 24);
response.addCookie(total_cookie);

if (cookies != null) {
    for (Cookie c : cookies) {
        if ("total".equals(c.getName())) {
            Cookie new_cookie = new Cookie("total", "2");
        }
    }
}

// ✅ Store the total in request scope so JSP can immediately use it
request.setAttribute("totalValue", newTotal);

request.getRequestDispatcher("Panier.jsp").forward(request, response);
}

}

```

StudentApp

Home.jsp :

```
<%--
```

```
    Document : home.jsp
```

```
    Created on : 29 Mar 2025, 23:50:47
```

```
    Author : farou
```

```
--%>
```

```
<%@page import="java.util.List"%>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
    <head>
```

```
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
        <title>JSP Page</title>
```

```
    </head>
```

```
    <body>
```

```
        <div style="border: 2px solid blue;padding: 5px;margin-top: 20px">
```

```
            <form action="Servlet" method="POST" style="display:flex;justify-content:
space-around;">
```

```
                <div>
```

```
                    <label style="color: red">Nom et prenom</label>
```

```
                    <br>
```

```
                    <input name="fullname" style="background-color: whitesmoke;border-color:
black"/>
```

```
                </div>
```

```
                <div>
```



```

    <label style="color: red">Date de naissance</label>

    <br>

    <input type="date" name="date" style="background-color:
whitesmoke;border-color: black"/>

</div>

<div>

    <label style="color: red">Sexe</label>

    <br>

    <label for="male">M</label>

    <input type="radio" name="sexe" value="male" style="background-color:
whitesmoke;border-color: black"/>

    <label for="female">F</label>

    <input type="radio" name="sexe" value="female" style="background-color:
whitesmoke;border-color: black"/>

</div>

<div>

    <label style="color: red">Filiere</label>

    <br>

    <input name="filiere" style="background-color: whitesmoke;border-color:
black"/>

</div>

<div>

    <label style="color: red">Compétences</label>

    <br>

    <input type="checkbox" name="comptences" value="java"
style="background-color: whitesmoke;border-color: black"/>

    <label for="java">java</label>

    <input type="checkbox" name="comptences" value="c" style="background-color:
whitesmoke;border-color: black"/>

```

```

    <label for="c">C</label>

    <input type="checkbox" name="comptences" value="c++"
style="background-color: whitesmoke;border-color: black"/>

    <label for="c++">C++</label>

    <input type="checkbox" name="comptences" value="php"
style="background-color: whitesmoke;border-color: black"/>

    <label for="php">PHP</label>

</div>

<div style="align-self:center">

    <input style="color: red;background-color: whitesmoke;border-color:
black;border-radius: 5px;" type="submit" value="ajouter"/>

</div>

</form>

<div style="display:flex;flex-direction: column;">

    <label style="color: red">

        Détails de la facture :

    </label>

    <table border="1">

        <thead>

            <tr>

                <th>Nom Complet</th>

                <th>Date de naissance</th>

                <th>Sexe</th>

                <th>Filiere</th>

                <th>compétences</th>

            </tr>

```

```

</thead>
<tbody>
  <%
    HttpSession s = request.getSession(false);
    if (s != null) {
      List<String> infos = (List<String>) s.getAttribute("infos");
      if (infos != null) {
        for (int i = 0; i < infos.size(); i += 5) {
          out.print("<tr>"
            + "<td>" + infos.get(i) + "</td>"
            + "<td>" + infos.get(i + 1) + "</td>"
            + "<td>" + infos.get(i + 2) + "</td>"
            + "<td>" + infos.get(i + 3) + "</td>"
            + "<td>" + infos.get(i + 4) + "</td>"
            + "</tr>");
        }
      }
    }

  %>
</tbody>
</table>
</div>
</div>

</body>

```

```
</html>
```

```
<%--
```

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
```

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>JSP Page</title>
```

```
</head>
```

```
<body>
```

```
<div style="border: 2px solid blue;padding: 5px;margin-top: 20px">
```

```
  <!-- Formulaire -->
```

```
  <form action="Servlet" method="POST" style="display:flex;justify-content:
space-around;">
```

```
    <div>
```

```
      <label style="color: red">Nom et prénom</label>
```

```
      <br>
```

```
      <input name="fullname" style="background-color: whitesmoke;border-color:
black"/>
```

```
    </div>
```

```
    <div>
```

```
      <label style="color: red">Date de naissance</label>
```

```
      <br>
```

```

    <input type="date" name="date" style="background-color:
whitesmoke;border-color: black"/>

```

```

</div>

```

```

<div>

```

```

    <label style="color: red">Sexe</label>

```

```

    <br>

```

```

    <label for="male">M</label>

```

```

    <input type="radio" name="sexe" value="male" style="background-color:
whitesmoke;border-color: black"/>

```

```

    <label for="female">F</label>

```

```

    <input type="radio" name="sexe" value="female" style="background-color:
whitesmoke;border-color: black"/>

```

```

</div>

```

```

<div>

```

```

    <label style="color: red">Filière</label>

```

```

    <br>

```

```

    <input name="filiere" style="background-color: whitesmoke;border-color: black"/>

```

```

</div>

```

```

<div>

```

```

    <label style="color: red">Compétences</label>

```

```

    <br>

```

```

    <input type="checkbox" name="comptences" value="java" />

```

```

    <label for="java">Java</label>

```

```

    <input type="checkbox" name="comptences" value="c" />

```

```

    <label for="c">C</label>

```

```

    <input type="checkbox" name="comptences" value="c++" />

```

```

    <label for="c++">C++</label>

```

```

        <input type="checkbox" name="competences" value="php" />
        <label for="php">PHP</label>
    </div>

    <div style="align-self:center">
        <input type="submit" value="ajouter" style="color: red;background-color:
whitesmoke;border-color: black;border-radius: 5px;" />
    </div>
</form>

<!-- Tableau d'affichage -->
<div style="display:flex;flex-direction: column;">
    <label style="color: red">Détails de la facture :</label>
    <table border="1">
        <thead>
            <tr>
                <th>Nom Complet</th>
                <th>Date de naissance</th>
                <th>Sexe</th>
                <th>Filière</th>
                <th>Compétences</th>
            </tr>
        </thead>
        <tbody>
            <c:if test="${not empty sessionScope.infos}">
                <c:set var="infos" value="${sessionScope.infos}" />
                <c:forEach var="i" begin="0" end="${fn:length(infos) - 1}" step="5">
                    <tr>

```

```

        <td>${infos[i]}</td>
        <td>${infos[i + 1]}</td>
        <td>${infos[i + 2]}</td>
        <td>${infos[i + 3]}</td>
        <td>${infos[i + 4]}</td>
    </tr>
</c:forEach>
</c:if>
</tbody>
</table>
</div>

```

```

</div>

```

```

</body>

```

```

</html>

```

```

--%>

```

Servlet.java :

```

/*

```

```

 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
this license

```

```

 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to edit this
template

```

```

*/

```

```

package control;

```

```

import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author farou
 */
public class Servlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        HttpSession s = request.getSession(false);
        if(s==null){
            s = request.getSession();
            List<String> infos = new ArrayList<>();
            s.setAttribute("infos", infos);

```



```
}
```

```
request.getRequestDispatcher("home.jsp").forward(request, response);
```

```
}
```

```
@Override
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
```

```
String fullname = request.getParameter("fullname");
```

```
String birthdate = request.getParameter("date");
```

```
String sexe = request.getParameter("sexe");
```

```
String filiere = request.getParameter("filiere");
```

```
String[] selectedArticles = request.getParameterValues("competences");
```

```
HttpSession s = request.getSession();
```

```
List<String> infos = (List<String>)s.getAttribute("infos");
```

```
if(infos==null){
```

```
    infos = new ArrayList<>();
```

```
s.setAttribute("infos", infos);
}
```

```
infos.add(fullname);
infos.add(birthdate);
infos.add(sexe);
infos.add(filiere);
if (selectedArticles != null) {
    infos.add(String.join(", ", selectedArticles));
} else {
    infos.add("No articles selected");
}
```

```
s.setAttribute("infos", infos);
```

```
request.getRequestDispatcher("home.jsp").forward(request, response);
}

}
```

Chase Tresor :

Voici un nouvel exercice dans le même style que le précédent, utilisant le paradigme **MVC** avec **JSP** et **Servlet**. Cette fois, le jeu s'appelle **"Chasse au Trésor"**. L'utilisateur doit entrer un nom d'utilisateur (stocké dans

la session), choisir entre deviner une **couleur** ou un **lieu**, et jouer selon des règles spécifiques. Comme dans l'exercice précédent, des indices seront fournis à certaines étapes, un tableau affichera les mauvaises réponses, et les tentatives/scores seront stockés dans des cookies.

Description de l'exercice : "Chasse au Trésor"

- **Objectif** : L'utilisateur doit deviner soit une couleur, soit un lieu où est caché un trésor.
- **Username** : Stocké dans la session pour identifier le joueur.
- **Choix** :
 - **Couleur** : L'utilisateur devine une couleur parmi une liste de 10 (ex. Rouge, Bleu, etc.). Il a 5 tentatives. À la 3e tentative, un indice sur la première lettre est donné. À la 4e, un indice sur la longueur du mot.
 - **Lieu** : L'utilisateur devine un lieu parmi une liste de 20 (ex. Forêt, Montagne, etc.). Il a 5 tentatives. À la 2e tentative, un indice sur le type de lieu (naturel/artificiel). À la 4e, un indice sur une lettre spécifique.
- **Tableau** : Affiche les mauvaises réponses pour les deux types de jeu.
- **Cookies** : Stockent les tentatives utilisées et le score (100 points au 1er essai, 80 au 2e, etc.).

Étape 1 : Le Modèle (`Tresor.java`)

Classe pour gérer l'état du jeu.

```
```java
```

```
package com.tresor.model;
```

```
import java.util.ArrayList;
```

```
import java.util.Arrays;
```

```
import java.util.List;
```

```
import java.util.Random;

public class Tresor {

 private String type; // "couleur" ou "lieu"

 private String cible; // Couleur ou lieu cible

 private int tentativesRestantes;

 private List<String> optionsPossibles;

 private List<String> mauvaisesReponses; // Stocke les mauvaises réponses

 private Random random;

 public Tresor() {

 this.random = new Random();

 this.tentativesRestantes = 5;

 this.mauvaisesReponses = new ArrayList<>();

 }

 public void initCouleur() {

 this.type = "couleur";

 this.optionsPossibles = Arrays.asList("Rouge", "Bleu", "Vert", "Jaune", "Noir", "Blanc", "Rose", "Violet", "Orange",
"Gris");

 this.cible = optionsPossibles.get(random.nextInt(optionsPossibles.size()));

 }

 public void initLieu() {

 this.type = "lieu";

 this.optionsPossibles = Arrays.asList(

 "Forêt", "Montagne", "Plage", "Désert", "Ville", "Village", "Château", "Grotte", "Île", "Lac",
```

```
 "Rivière", "Plaine", "Colline", "Tour", "Pont", "Caverne", "Ferme", "Port", "Temple", "Jardin"

);

 this.cible = optionsPossibles.get(random.nextInt(optionsPossibles.size()));
}

public String getType() {

 return type;

}

public String getCible() {

 return cible;

}

public int getTentativesRestantes() {

 return tentativesRestantes;

}

public void decremenerTentatives() {

 this.tentativesRestantes--;

}

public List<String> getOptionsPossibles() {

 return optionsPossibles;

}

public List<String> getMauvaisesReponses() {
```

```

 return mauvaisesReponses;
}

public void ajouterMauvaiseReponse(String reponse) {
 mauvaisesReponses.add(reponse);
}

public int calculerScore() {
 if (tentativesRestantes == 5) return 100; // Réussi au 1er essai
 return tentativesRestantes * 20; // 80, 60, 40, 20, 0
}

public String getIndice() {
 if ("couleur".equals(type)) {
 if (tentativesRestantes == 2) { // 3e tentative (après 2 restantes)
 return "Indice : La première lettre est " + cible.charAt(0) + " ";
 } else if (tentativesRestantes == 1) { // 4e tentative (après 1 restante)
 return "Indice : Le mot a " + cible.length() + " lettres.";
 }
 } else if ("lieu".equals(type)) {
 if (tentativesRestantes == 3) { // 2e tentative (après 3 restantes)
 String typeLieu = (cible.equals("Ville") || cible.equals("Village") || cible.equals("Château") ||
 cible.equals("Tour") || cible.equals("Pont") || cible.equals("Ferme") ||
 cible.equals("Port") || cible.equals("Temple")) ? "artificiel" : "naturel";
 return "Indice : C'est un lieu " + typeLieu + " ";
 } else if (tentativesRestantes == 1) { // 4e tentative (après 1 restante)

```

```

 return "Indice : Une lettre est " + cible.charAt(cible.length() / 2) + " (milieu).";
 }

}

return "";

}

}

...

```

### ### Étape 2 : La Vue (`tresor.jsp`)

Page JSP pour l'interface utilisateur.

```

```.jsp

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%@ page import="com.tresor.model.Tresor" %>

<%@ page import="java.util.List" %>

<!DOCTYPE html>

<html>

<head>

    <title>Chasse au Trésor</title>

    <style>

        .message { color: blue; }

        .error { color: red; }

        table { border-collapse: collapse; width: 50%; }

        th, td { border: 1px solid black; padding: 8px; text-align: left; }
```

```

</style>

</head>

<body>

  <h2>Chasse au Trésor</h2>

  <%

    String username = (String) session.getAttribute("username");

    Tresor tresor = (Tresor) session.getAttribute("tresor");

    String message = (String) request.getAttribute("message");

    String error = (String) request.getAttribute("error");

  %>

  <!-- Formulaire pour entrer le username -->

  <% if (username == null) { %>

    <form action="TresorServlet" method="post">

      <input type="hidden" name="action" value="setUsername">

      <label>Entrez votre nom d'utilisateur :</label>

      <input type="text" name="username" required>

      <input type="submit" value="Commencer">

    </form>

  <% } else { %>

    <p>Bienvenue, <%= username %> !</p>

  <!-- Choix du type de jeu -->

  <% if (tresor == null) { %>

    <form action="TresorServlet" method="post">

```



```

<input type="hidden" name="action" value="choisirType">

<label>Choisissez une option :</label><br>

<input type="radio" name="type" value="couleur" required> Deviner une couleur<br>

<input type="radio" name="type" value="lieu"> Deviner un lieu<br>

<input type="submit" value="Jouer">

</form>

<% } else { %>

<h3>Jeu : <%= tresor.getType().equals("couleur") ? "Deviner une couleur" : "Deviner un lieu" %></h3>

<p>Tentatives restantes : <%= tresor.getTentativesRestantes() %></p>

<!-- Jeu de couleur ou lieu -->

<p>Devinez <%= tresor.getType().equals("couleur") ? "la couleur du trésor" : "le lieu du trésor" %> :</p>

<% if (!tresor.getIndice().isEmpty()) { %>

<p class="message"><%= tresor.getIndice() %></p>

<% } %>

<form action="TresorServlet" method="post">

  <input type="hidden" name="action" value="deviner">

  <select name="guess" required>

    <% for (String option : tresor.getOptionsPossibles()) { %>

      <option value="<%= option %>"><%= option %></option>

    <% } %>

  </select>

  <input type="submit" value="Deviner">

</form>

<!-- Tableau des mauvaises réponses -->

```

```

<% if (!tresor.getMauvaisesReponses().isEmpty()) { %>

<h3>Mauvaises réponses :</h3>

<table>

  <tr>

    <th>Tentative</th>

    <th>Réponse</th>

  </tr>

  <%

    List<String> mauvaisesReponses = tresor.getMauvaisesReponses();

    for (int i = 0; i < mauvaisesReponses.size(); i++) {

  %>

  <tr>

    <td><%= i + 1 %></td>

    <td><%= mauvaisesReponses.get(i) %></td>

  </tr>

  <% } %>

</table>

<% } %>

<!-- Messages -->

<% if (message != null) { %>

<p class="message"><%= message %></p>

<% } %>

<% if (error != null) { %>

<p class="error"><%= error %></p>

<% } %>

```

```
<%}%>
```

```
<%}%>
```

```
</body>
```

```
</html>
```

```
...
```

```
---
```

Étape 3 : Le Contrôleur (`TresorServlet.java`)

Servlet pour gérer les actions du jeu.

```
```java
```

```
package com.tresor.controller;
```

```
import com.tresor.model.Tresor;
```

```
import java.io.IOException;
```

```
import javax.servlet.ServletException;
```

```
import javax.servlet.annotation.WebServlet;
```

```
import javax.servlet.http.Cookie;
```

```
import javax.servlet.http.HttpServlet;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;
```

```
import javax.servlet.http.HttpSession;
```

```
@WebServlet("/TresorServlet")
```

```
public class TresorServlet extends HttpServlet {
```

@Override

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
 throws ServletException, IOException {
 request.getRequestDispatcher("/WEB-INF/views/tresor.jsp").forward(request, response);
}
```

@Override

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
 throws ServletException, IOException {
```

```
 HttpSession session = request.getSession();
```

```
 String action = request.getParameter("action");
```

```
 if ("setUsername".equals(action)) {
```

```
 String username = request.getParameter("username");
```

```
 session.setAttribute("username", username);
```

```
 response.addCookie(new Cookie("lastUsername", username));
```

```
 } else if ("choisirType".equals(action)) {
```

```
 Tresor tresor = new Tresor();
```

```
 String type = request.getParameter("type");
```

```
 if ("couleur".equals(type)) {
```

```
 tresor.initCouleur();
```

```
 } else if ("lieu".equals(type)) {
```

```
 tresor.initLieu();
```

```
 }
```

```
 session.setAttribute("tresor", tresor);
```

```

} else if ("deviner".equals(action)) {

 Tresor tresor = (Tresor) session.getAttribute("tresor");

 String guess = request.getParameter("guess");

 tresor.decrementerTentatives();

 if (guess.equals(tresor.getCible())) {

 int score = tresor.calculerScore();

 request.setAttribute("message", "Félicitations ! Le trésor était " +

 (tresor.getType().equals("couleur") ? "de couleur " : "caché dans ") +

 tresor.getCible() + ". Score : " + score);

 updateCookies(request, response, tresor.getTentativesRestantes(), score);

 session.removeAttribute("tresor");

 } else {

 tresor.ajouterMauvaiseReponse(guess);

 if (tresor.getTentativesRestantes() == 0) {

 request.setAttribute("error", "Perdu ! Le trésor était " +

 (tresor.getType().equals("couleur") ? "de couleur " : "caché dans ") +

 tresor.getCible());

 updateCookies(request, response, tresor.getTentativesRestantes(), 0);

 session.removeAttribute("tresor");

 } else {

 request.setAttribute("message", "Ce n'est pas correct ! Essayez encore.");

 }

 }

}

request.getRequestDispatcher("/WEB-INF/views/tresor.jsp").forward(request, response);

```

```

 }

 private void updateCookies(HttpServletRequest request, HttpServletResponse response, int tentatives, int score) {

 String username = (String) request.getSession().getAttribute("username");

 Cookie attemptsCookie = new Cookie(username + "_attempts", String.valueOf(5 - tentatives));

 Cookie scoreCookie = new Cookie(username + "_score", String.valueOf(score));

 attemptsCookie.setMaxAge(60 * 60 * 24); // 1 jour

 scoreCookie.setMaxAge(60 * 60 * 24);

 response.addCookie(attemptsCookie);

 response.addCookie(scoreCookie);

 }

}

...

```

### Étape 4 : Structure du projet

...

projet-tresor/

├── src/

| ├── com.tresor.model/

| | └── Tresor.java

| └── com.tresor.controller/

| └── TresorServlet.java

├── WEB-INF/

| ├── views/

```

| | └─ tresor.jsp
| └─ web.xml (optionnel si annotation utilisée)
...

```

### ### Fonctionnement

#### 1. **\*\*Entrée du username\*\*** :

- L'utilisateur entre un nom (ex. "Bob"), stocké dans la session et dans un cookie `"lastUsername"`.

#### 2. **\*\*Choix du jeu\*\*** :

- Choix entre "couleur" ou "lieu", initialisant l'objet `Tresor``.

#### 3. **\*\*Jeu de couleur\*\*** :

- Liste de 10 couleurs dans un menu déroulant.
- Si incorrect, la réponse est ajoutée au tableau des mauvaises réponses.
- **\*\*Indices\*\*** :
  - 3e tentative (2 restantes) : "Indice : La première lettre est 'R'" (pour "Rouge").
  - 4e tentative (1 restante) : "Indice : Le mot a 5 lettres."
- 5 tentatives max, score dans les cookies.

#### 4. **\*\*Jeu de lieu\*\*** :

- Liste de 20 lieux dans un menu déroulant.
- Si incorrect, la réponse est ajoutée au tableau des mauvaises réponses.
- **\*\*Indices\*\*** :
  - 2e tentative (3 restantes) : "Indice : C'est un lieu naturel." (pour "Forêt").

- 4e tentative (1 restante) : "Indice : Une lettre est 'r' (milieu)." (pour "Forêt").

- 5 tentatives max, score dans les cookies.

5. **\*\*Tableau des mauvaises réponses\*\*** :

- Affiche chaque tentative incorrecte (ex. "1 - Bleu", "2 - Vert").

---

### Exemple de déroulement

- **\*\*Username\*\*** : "Bob".

- **\*\*Choix\*\*** : "couleur".

- Système choisit : "Rouge".

- Tentatives :

1. "Bleu" → "Incorrect !" (Tableau : "1 - Bleu").

2. "Vert" → "Incorrect !" (Tableau : "1 - Bleu", "2 - Vert").

3. "Jaune" → "Incorrect ! Indice : La première lettre est 'R'" (Tableau : "1 - Bleu", "2 - Vert", "3 - Jaune").

4. "Noir" → "Incorrect ! Indice : Le mot a 5 lettres." (Tableau : "1 - Bleu", "2 - Vert", "3 - Jaune", "4 - Noir").

5. "Gris" → "Perdu ! Le trésor était de couleur Rouge." (Tableau : "1 - Bleu", "2 - Vert", "3 - Jaune", "4 - Noir", "5 - Gris").

- Cookies : `"Bob_attempts"=5`,"Bob_score"=0``.

- **\*\*Choix\*\*** : "lieu".

- Système choisit : "Forêt".

- Tentatives :

1. "Ville" → "Incorrect !" (Tableau : "1 - Ville").

2. "Plage" → "Incorrect ! Indice : C'est un lieu naturel." (Tableau : "1 - Ville", "2 - Plage").

3. "Forêt" → "Gagné ! Score : 60" (Tableau : "1 - Ville", "2 - Plage").



- Cookies : ``"Bob\_attempts"=2`,`"Bob\_score"=60`.

---

### ### Déploiement

Déployez sur Tomcat à `http://localhost:8080/projet-tresor/TresorServlet`. Les sessions isolent les jeux par utilisateur, les cookies stockent les résultats, et le tableau affiche les erreurs.

Si vous voulez des modifications ou un autre exemple, faites-moi signe !

## Autre Exemples Panier :

Voici une version améliorée de l'application de facture avec les fonctionnalités demandées : **suppression** et **modification** d'un article, un **filtre** par nom ou date de stockage, ainsi que la gestion des **sessions** pour associer les articles à chaque utilisateur et l'utilisation de **cookies** pour mémoriser des informations (par exemple, le dernier utilisateur connecté).

Je vais mettre à jour le code précédent en suivant le paradigme MVC avec JSP et Servlet.

---

### ### Modifications principales

1. **Ajout de la date** : Chaque article aura une date de stockage (ajoutée automatiquement).
2. **Suppression et modification** : Boutons pour supprimer ou modifier un article dans le tableau.
3. **Filtre** : Formulaire pour filtrer par nom ou date.
4. **Sessions** : Chaque utilisateur aura sa propre facture stockée dans sa session.
5. **Cookies** : Stockage du dernier utilisateur connecté pour personnalisation.

---

### Étape 1 : Mise à jour du Modèle (`Article.java`)

Ajout d'un champ `dateAjout` et d'un identifiant unique pour chaque article.

```
```java
```

```
package com.facture.model;
```

```
import java.time.LocalDateTime;
```

```
import java.util.UUID;
```

```
public class Article {
```

```
    private String id; // Identifiant unique
```

```
    private String nom;
```

```
    private int quantite;
```

```
    private double prixUnitaire;
```

```
    private LocalDateTime dateAjout;
```

```
    public Article(String nom, int quantite, double prixUnitaire) {
```

```
        this.id = UUID.randomUUID().toString(); // Génération d'un ID unique
```

```
        this.nom = nom;
```

```
        this.quantite = quantite;
```

```
        this.prixUnitaire = prixUnitaire;
```

```
        this.dateAjout = LocalDateTime.now(); // Date actuelle
```

```
    }
```

```
public String getId() {  
    return id;  
}
```

```
public String getNom() {  
    return nom;  
}
```

```
public void setNom(String nom) {  
    this.nom = nom;  
}
```

```
public int getQuantite() {  
    return quantite;  
}
```

```
public void setQuantite(int quantite) {  
    this.quantite = quantite;  
}
```

```
public double getPrixUnitaire() {  
    return prixUnitaire;  
}
```

```
public void setPrixUnitaire(double prixUnitaire) {  
    this.prixUnitaire = prixUnitaire;  
}
```

```

    }

    public LocalDateTime getDateAjout() {

        return dateAjout;

    }

    public double getPrixTotal() {

        return quantite * prixUnitaire;

    }

}

...

---
```

Étape 2 : Mise à jour de la Vue (`facture.jsp`)

Ajout des fonctionnalités de suppression, modification, filtrage, et affichage des informations utilisateur via session/cookies.

```

```.jsp

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%@ page import="com.facture.model.Article" %>

<%@ page import="java.util.List" %>

<%@ page import="java.time.format.DateTimeFormatter" %>

<!DOCTYPE html>

<html>

<head>

 <title>Application Facture</title>
```

```

<style>

 table { border-collapse: collapse; width: 70%; }

 th, td { border: 1px solid black; padding: 8px; text-align: left; }

 .filter-form { margin: 20px 0; }

</style>

</head>

<body>

 <h2>Facture pour <%= session.getAttribute("username") != null ? session.getAttribute("username") : "Utilisateur
inconnu" %></h2>

 <!-- Formulaire pour ajouter un article -->

 <form action="FactureServlet" method="post">

 <input type="hidden" name="action" value="ajouter">

 <label>Nom de l'article :</label>

 <input type="text" name="nom" required>

 <label>Quantité :</label>

 <input type="number" name="quantite" min="1" required>

 <label>Prix unitaire :</label>

 <input type="number" step="0.01" name="prixUnitaire" min="0" required>

 <input type="submit" value="Ajouter">

 </form>

 <!-- Formulaire de filtrage -->

 <div class="filter-form">

 <form action="FactureServlet" method="get">

 <label>Filtrer par nom :</label>

 <input type="text" name="filtreNom">


```

```

<label>Filtrer par date (YYYY-MM-DD) :</label>

<input type="date" name="filtreDate">

<input type="submit" value="Filtrer">

</form>

</div>

<!-- Tableau de la facture -->

<h3>Détails de la facture :</h3>

<table>

 <tr>

 <th>ID</th>

 <th>Article</th>

 <th>Quantité</th>

 <th>Prix Unitaire</th>

 <th>Prix Total</th>

 <th>Date Ajout</th>

 <th>Actions</th>

 </tr>

 <%

 List<Article> facture = (List<Article>) request.getAttribute("facture");

 double totalGeneral = 0;

 DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm");

 if (facture != null) {

 for (Article article : facture) {

 double prixTotal = article.getPrixTotal();

 totalGeneral += prixTotal;

```

```

%>

<tr>

 <td><%= article.getId() %></td>

 <td><%= article.getNom() %></td>

 <td><%= article.getQuantite() %></td>

 <td><%= article.getPrixUnitaire() %></td>

 <td><%= prixTotal %></td>

 <td><%= article.getDateAjout().format(formatter) %></td>

 <td>

 <form action="FactureServlet" method="post" style="display:inline;">

 <input type="hidden" name="action" value="supprimer">

 <input type="hidden" name="id" value="<%= article.getId() %>">

 <input type="submit" value="Supprimer">

 </form>

 <form action="FactureServlet" method="post" style="display:inline;">

 <input type="hidden" name="action" value="modifier">

 <input type="hidden" name="id" value="<%= article.getId() %>">

 <input type="text" name="nouveauNom" value="<%= article.getNom() %>" required>

 <input type="number" name="nouvelleQuantite" value="<%= article.getQuantite() %>" min="1" required>

 <input type="number" step="0.01" name="nouveauPrix" value="<%= article.getPrixUnitaire() %>" min="0"
required>

 <input type="submit" value="Modifier">

 </form>

 </td>

</tr>

<%

}

```

```

 }

 %>

 <tr>

 <td colspan="6">Total à payer :</td>

 <td><%= totalGeneral %> DH</td>

 </tr>

</table>

</body>

</html>

...

```

### Étape 3 : Mise à jour du Contrôleur (`FactureServlet.java`)

Ajout de la logique pour supprimer, modifier, filtrer, gérer les sessions par utilisateur, et utiliser les cookies.

```

```java

package com.facture.controller;

import com.facture.model.Article;

import java.io.IOException;

import java.time.LocalDate;

import java.time.LocalDateTime;

import java.time.format.DateTimeFormatter;

import java.util.ArrayList;

import java.util.List;
```



```

import java.util.stream.Collectors;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

@WebServlet("/FactureServlet")

public class FactureServlet extends HttpServlet {

    @Override

    protected void doGet(HttpServletRequest request, HttpServletResponse response)

        throws ServletException, IOException {

        HttpSession session = request.getSession();

        String username = getUsernameFromCookies(request);

        if (username == null) {

            username = "User" + session.getId().substring(0, 5); // ID unique basé sur la session

            Cookie userCookie = new Cookie("username", username);

            userCookie.setMaxAge(60 * 60 * 24); // Cookie valide 1 jour

            response.addCookie(userCookie);

        }

        session.setAttribute("username", username);

        List<Article> facture = (List<Article>) session.getAttribute("facture");

```

```

if (facture == null) {

    facture = new ArrayList<>();

    session.setAttribute("facture", facture);

}

// Filtrage

String filtreNom = request.getParameter("filtreNom");

String filtreDate = request.getParameter("filtreDate");

List<Article> factureFiltree = new ArrayList<>(facture);

if (filtreNom != null && !filtreNom.isEmpty()) {

    factureFiltree = factureFiltree.stream()

        .filter(a -> a.getNom().toLowerCase().contains(filtreNom.toLowerCase()))

        .collect(Collectors.toList());

}

if (filtreDate != null && !filtreDate.isEmpty()) {

    LocalDate date = LocalDate.parse(filtreDate);

    factureFiltree = factureFiltree.stream()

        .filter(a -> a.getDateAjout().toLocalDate().equals(date))

        .collect(Collectors.toList());

}

request.setAttribute("facture", factureFiltree);

request.getRequestDispatcher("/WEB-INF/views/facture.jsp").forward(request, response);

}

```

@Override

```

protected void doPost(HttpServletRequest request, HttpServletResponse response)

    throws ServletException, IOException {

    HttpSession session = request.getSession();

    List<Article> facture = (List<Article>) session.getAttribute("facture");

    if (facture == null) {

        facture = new ArrayList<>();

        session.setAttribute("facture", facture);

    }

    String action = request.getParameter("action");

    if ("ajouter".equals(action)) {

        String nom = request.getParameter("nom");

        int quantite = Integer.parseInt(request.getParameter("quantite"));

        double prixUnitaire = Double.parseDouble(request.getParameter("prixUnitaire"));

        Article article = new Article(nom, quantite, prixUnitaire);

        facture.add(article);

    } else if ("supprimer".equals(action)) {

        String id = request.getParameter("id");

        facture.removeIf(a -> a.getId().equals(id));

    } else if ("modifier".equals(action)) {

        String id = request.getParameter("id");

        String nouveauNom = request.getParameter("nouveauNom");

        int nouvelleQuantite = Integer.parseInt(request.getParameter("nouvelleQuantite"));

        double nouveauPrix = Double.parseDouble(request.getParameter("nouveauPrix"));

        for (Article a : facture) {

```

```

        if (a.getId().equals(id)) {

            a.setNom(nouveauNom);

            a.setQuantite(nouvelleQuantite);

            a.setPrixUnitaire(nouveauPrix);

            break;

        }

    }

}

session.setAttribute("facture", facture);

request.setAttribute("facture", facture);

request.getRequestDispatcher("/WEB-INF/views/facture.jsp").forward(request, response);

}

```

```

private String getUsernameFromCookies(HttpServletRequest request) {

    Cookie[] cookies = request.getCookies();

    if (cookies != null) {

        for (Cookie cookie : cookies) {

            if ("username".equals(cookie.getName())) {

                return cookie.getValue();

            }

        }

    }

    return null;

}

}

```

...

Étape 4 : Structure du projet (inchangée)

...

projet-facture/

|—src/

| |—com.facture.model/

| | |—Article.java

| |—com.facture.controller/

| |—FactureServlet.java

|—WEB-INF/

| |—views/

| | |—facture.jsp

| |—web.xml (optionnel)

...

Fonctionnement

1. **Sessions et Cookies** :

- Chaque utilisateur a une session unique avec sa propre liste d'articles (`facture`).
- Un cookie ``username`` est créé pour mémoriser l'identifiant de l'utilisateur (basé sur la session si aucun cookie n'existe).

2. **Ajout d'un article** :

- Formulaire POST avec `action=ajouter`.
- L'article est ajouté à la liste dans la session.

3. **Suppression** :

- Bouton "Supprimer" dans le tableau envoie une requête POST avec `action=supprimer` et l'ID de l'article.

4. **Modification** :

- Formulaire inline dans le tableau envoie une requête POST avec `action=modifier` et les nouvelles valeurs.

5. **Filtrage** :

- Formulaire GET avec `filtreNom` ou `filtreDate` pour filtrer les articles affichés.

6. **Affichage** :

- Le tableau montre les articles avec ID, nom, quantité, prix, date, et options d'action.
- Le total est recalculé dynamiquement.

Exemple de résultat

Pour un utilisateur "User12345" :

- Ajout de "Crayon" (20 × 2) et "Ramette" (15 × 30).

- Tableau :

...

ID	Article	Quantité	Prix Unitaire	Prix Total	Date Ajout	Actions
-----	-----	-----	-----	-----	-----	-----
UUID1	Crayon	20	2	40	2025-04-07 10:00	Suppr / Modifier

| UUID2 | Ramette | 15 | 30 | 450 | 2025-04-07 10:01 | Suppr / Modifier |

...

- Total : **490 DH**.

- Filtre par "Crayon" → affiche seulement la ligne "Crayon".

Jeu de devinette Words Edition 👍

Voici une version mise à jour de l'exercice avec les nouvelles fonctionnalités demandées :

- Pour le jeu de noms, à partir de la 2e tentative, le système affiche une lettre du nom cible avec sa position (1re, 2e, dernière, etc.), puis une autre lettre après la 4e tentative.

- Un tableau affiche les mauvaises réponses de l'utilisateur pour les deux types de jeu (nombre et nom) jusqu'à ce qu'il trouve la bonne réponse ou atteigne la limite des 5 tentatives.

Étape 1 : Mise à jour du Modèle (`Jeu.java`)

Ajout de la gestion des indices pour le jeu de noms et d'une liste pour stocker les mauvaises réponses.

```
```java
```

```
package com.jeu.model;
```

```
import java.util.ArrayList;
```

```
import java.util.Arrays;
```

```
import java.util.List;
```

```
import java.util.Random;
```

```
public class Jeu {
```

```
 private String type; // "nombre" ou "nom"
```

```

private int nombreCible; // Pour deviner un nombre

private String nomCible; // Pour deviner un nom

private int tentativesRestantes;

private List<String> nomsPossibles;

private List<String> mauvaisesReponses; // Stocke les mauvaises réponses

private Random random;

public Jeu() {

 this.random = new Random();

 this.tentativesRestantes = 5;

 this.mauvaisesReponses = new ArrayList<>();

 this.nomsPossibles = Arrays.asList(

 "Alice", "Bob", "Charlie", "David", "Emma", "Frank", "Grace", "Henry", "Isabelle", "James",

 "Kylie", "Liam", "Mia", "Noah", "Olivia", "Peter", "Quinn", "Rose", "Sam", "Tara",

 "Uma", "Victor", "Wendy", "Xavier", "Yara", "Zoe", "Ethan", "Sophia", "Mason", "Luna"

);

}

public void initNombre() {

 this.type = "nombre";

 this.nombreCible = random.nextInt(101); // 0 à 100

}

public void initNom() {

 this.type = "nom";

 this.nomCible = nomsPossibles.get(random.nextInt(nomsPossibles.size()));

```



```
}
```

```
public String getType() {
 return type;
}
```

```
public int getNombreCible() {
 return nombreCible;
}
```

```
public String getNomCible() {
 return nomCible;
}
```

```
public int getTentativesRestantes() {
 return tentativesRestantes;
}
```

```
public void decrementerTentatives() {
 this.tentativesRestantes--;
}
```

```
public List<String> getNomsPossibles() {
 return nomsPossibles;
}
```

```

public List<String> getMauvaisesReponses() {
 return mauvaisesReponses;
}

public void ajouterMauvaiseReponse(String reponse) {
 mauvaisesReponses.add(reponse);
}

public int calculerScore() {
 if (tentativesRestantes == 5) return 100; // Réussi au 1er essai
 return tentativesRestantes * 20; // 80, 60, 40, 20, 0
}

public String getIndiceNom() {
 if (tentativesRestantes <= 3 && tentativesRestantes > 1) { // 2e tentative (après 3 restantes)
 char lettre = nomCible.charAt(1); // 2e lettre
 return "Indice : La 2e lettre est " + lettre + ".";
 } else if (tentativesRestantes == 1) { // 4e tentative (après 1 restante)
 char lettre = nomCible.charAt(nomCible.length() - 1); // Dernière lettre
 return "Indice : La dernière lettre est " + lettre + ".";
 }
 return "";
}
}
...

```

---

### ### Étape 2 : Mise à jour de la Vue (`jeu.jsp`)

Ajout du tableau des mauvaises réponses et gestion des indices pour le jeu de noms.

```
```.jsp

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%@ page import="com.jeu.model.Jeu" %>

<%@ page import="java.util.List" %>

<!DOCTYPE html>

<html>

<head>

    <title>Jeu de Devinettes</title>

    <style>

        .message { color: blue; }

        .error { color: red; }

        table { border-collapse: collapse; width: 50%; }

        th, td { border: 1px solid black; padding: 8px; text-align: left; }

    </style>

</head>

<body>

    <h2>Jeu de Devinettes</h2>

    <%

        String username = (String) session.getAttribute("username");

        Jeu jeu = (Jeu) session.getAttribute("jeu");
```

```

String message = (String) request.getAttribute("message");

String error = (String) request.getAttribute("error");

%>

<!-- Formulaire pour entrer le username -->

<% if (username == null) { %>

<form action="JeuServlet" method="post">

    <input type="hidden" name="action" value="setUsername">

    <label>Entrez votre nom d'utilisateur :</label>

    <input type="text" name="username" required>

    <input type="submit" value="Commencer">

</form>

<% } else { %>

<p>Bienvenue, <%= username %> !</p>

<!-- Choix du type de jeu -->

<% if (jeu == null) { %>

<form action="JeuServlet" method="post">

    <input type="hidden" name="action" value="choisirType">

    <label>Choisissez une option :</label><br>

    <input type="radio" name="type" value="nombre" required> Deviner un nombre<br>

    <input type="radio" name="type" value="nom"> Deviner un nom<br>

    <input type="submit" value="Jouer">

</form>

<% } else { %>

<h3>Jeu : <%= jeu.getType().equals("nombre") ? "Deviner un nombre" : "Deviner un nom" %></h3>

```

```
<p>Tentatives restantes : <%= jeu.getTentativesRestantes() %></p>
```

```
<!-- Jeu de nombre -->
```

```
<% if ("nombre".equals(jeu.getType())) { %>
```

```
<p>Devinez un nombre entre 0 et 100.</p>
```

```
<form action="JeuServlet" method="post">
```

```
    <input type="hidden" name="action" value="devinerNombre">
```

```
    <input type="number" name="guess" min="0" max="100" required>
```

```
    <input type="submit" value="Deviner">
```

```
</form>
```

```
<% } %>
```

```
<!-- Jeu de nom -->
```

```
<% if ("nom".equals(jeu.getType())) { %>
```

```
<p>Choisissez le nom du kidnappeur parmi cette liste :</p>
```

```
<% if (!jeu.getIndiceNom().isEmpty()) { %>
```

```
<p class="message"><%= jeu.getIndiceNom() %></p>
```

```
<% } %>
```

```
<form action="JeuServlet" method="post">
```

```
    <input type="hidden" name="action" value="devinerNom">
```

```
    <select name="guess" required>
```

```
        <% for (String nom : jeu.getNomsPossibles()) { %>
```

```
        <option value="<%= nom %>"><%= nom %></option>
```

```
        <% } %>
```

```
</select>
```

```
    <input type="submit" value="Deviner">
```

```
</form>
```

```
<% } %>
```

```
<!-- Tableau des mauvaises réponses -->
```

```
<% if (!jeu.getMauvaisesReponses().isEmpty()) { %>
```

```
<h3>Mauvaises réponses :</h3>
```

```
<table>
```

```
<tr>
```

```
<th>Tentative</th>
```

```
<th>Réponse</th>
```

```
</tr>
```

```
<%
```

```
List<String> mauvaisesReponses = jeu.getMauvaisesReponses();
```

```
for (int i = 0; i < mauvaisesReponses.size(); i++) {
```

```
%>
```

```
<tr>
```

```
<td><%= i + 1 %></td>
```

```
<td><%= mauvaisesReponses.get(i) %></td>
```

```
</tr>
```

```
<% } %>
```

```
</table>
```

```
<% } %>
```

```
<!-- Messages -->
```

```
<% if (message != null) { %>
```

```
<p class="message"><%= message %></p>
```

```

<%}%>

<% if (error != null) { %>

<p class="error"><%= error %></p>

<%}%>

<%}%>

<%}%>

</body>

</html>

...

---
```

Étape 3 : Mise à jour du Contrôleur (`JeuServlet.java`)

Ajout de la logique pour les indices et le tableau des mauvaises réponses.

```

```java

package com.jeu.controller;

import com.jeu.model.Jeu;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;
```

```
import javax.servlet.http.HttpSession;
```

```
@WebServlet("/JeuServlet")
```

```
public class JeuServlet extends HttpServlet {
```

```
 @Override
```

```
 protected void doGet(HttpServletRequest request, HttpServletResponse response)
```

```
 throws ServletException, IOException {
```

```
 request.getRequestDispatcher("/WEB-INF/views/jeu.jsp").forward(request, response);
```

```
 }
```

```
 @Override
```

```
 protected void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
 throws ServletException, IOException {
```

```
 HttpSession session = request.getSession();
```

```
 String action = request.getParameter("action");
```

```
 if ("setUsername".equals(action)) {
```

```
 String username = request.getParameter("username");
```

```
 session.setAttribute("username", username);
```

```
 response.addCookie(new Cookie("lastUsername", username));
```

```
 } else if ("choisirType".equals(action)) {
```

```
 Jeu jeu = new Jeu();
```

```
 String type = request.getParameter("type");
```

```
 if ("nombre".equals(type)) {
```

```
 jeu.initNombre();
```



```

 } else if ("nom".equals(type)) {

 jeu.initNom();

 }

 session.setAttribute("jeu", jeu);

} else if ("devinerNombre".equals(action)) {

 Jeu jeu = (Jeu) session.getAttribute("jeu");

 int guess = Integer.parseInt(request.getParameter("guess"));

 jeu.decrementerTentatives();

 if (guess == jeu.getNombreCible()) {

 int score = jeu.calculerScore();

 request.setAttribute("message", "Félicitations ! Vous avez deviné le nombre : " + jeu.getNombreCible() + ".
Score : " + score);

 updateCookies(request, response, jeu.getTentativesRestantes(), score);

 session.removeAttribute("jeu");

 } else {

 jeu.ajouterMauvaiseReponse(String.valueOf(guess));

 if (jeu.getTentativesRestantes() == 0) {

 request.setAttribute("error", "Perdu ! Le nombre était : " + jeu.getNombreCible());

 updateCookies(request, response, jeu.getTentativesRestantes(), 0);

 session.removeAttribute("jeu");

 } else {

 String hint = guess > jeu.getNombreCible() ? "Trop grand !" : "Trop petit !";

 request.setAttribute("message", hint);

 }

 }

}

} else if ("devinerNom".equals(action)) {

 Jeu jeu = (Jeu) session.getAttribute("jeu");

```

```

String guess = request.getParameter("guess");

jeu.decrementerTentatives();

if (guess.equals(jeu.getNomCible())) {

 int score = jeu.calculerScore();

 request.setAttribute("message", "Félicitations ! Le kidnappeur était : " + jeu.getNomCible() + ". Score : " +
score);

 updateCookies(request, response, jeu.getTentativesRestantes(), score);

 session.removeAttribute("jeu");

} else {

 jeu.ajouterMauvaiseReponse(guess);

 if (jeu.getTentativesRestantes() == 0) {

 request.setAttribute("error", "Perdu ! Le kidnappeur était : " + jeu.getNomCible());

 updateCookies(request, response, jeu.getTentativesRestantes(), 0);

 session.removeAttribute("jeu");

 } else {

 request.setAttribute("message", "Ce n'est pas le bon nom ! Essayez encore.");

 }

}

}

request.getRequestDispatcher("/WEB-INF/views/jeu.jsp").forward(request, response);

}

private void updateCookies(HttpServletRequest request, HttpServletResponse response, int tentatives, int score) {

 String username = (String) request.getSession().getAttribute("username");

 Cookie attemptsCookie = new Cookie(username + "_attempts", String.valueOf(5 - tentatives));

 Cookie scoreCookie = new Cookie(username + "_score", String.valueOf(score));

```

```

 attemptsCookie.setMaxAge(60 * 60 * 24); // 1 jour

 scoreCookie.setMaxAge(60 * 60 * 24);

 response.addCookie(attemptsCookie);

 response.addCookie(scoreCookie);

}

}

...

```

---

### Étape 4 : Structure du projet (inchangée)

...

projet-jeu/

```

├── src/
| ├── com.jeu.model/
| | └── Jeu.java
| └── com.jeu.controller/
| └── JeuServlet.java
├── WEB-INF/
| ├── views/
| | └── jeu.jsp
| └── web.xml (optionnel si annotation utilisée)

```

...

---

### ### Fonctionnement

#### 1. **\*\*Entrée du username\*\*** :

- L'utilisateur entre un nom (ex. "Alice"), stocké dans la session et dans un cookie `"lastUsername"`.

#### 2. **\*\*Choix du jeu\*\*** :

- Choix entre "nombre" ou "nom", initialisant l'objet `Jeu``.

#### 3. **\*\*Jeu de nombre\*\*** :

- L'utilisateur entre un nombre (0-100).
- Si incorrect, la réponse est ajoutée au tableau des mauvaises réponses.
- Indications : "Trop grand" ou "Trop petit".
- 5 tentatives max, score stocké dans les cookies.

#### 4. **\*\*Jeu de nom\*\*** :

- Liste de 30 noms dans un menu déroulant.
- Si incorrect, la réponse est ajoutée au tableau des mauvaises réponses.
- **\*\*Indices\*\*** :
  - Après 2e tentative (3 restantes) : Affiche la 2e lettre (ex. "Indice : La 2e lettre est 'm.'" pour "Emma").
  - Après 4e tentative (1 restante) : Affiche la dernière lettre (ex. "Indice : La dernière lettre est 'a'").
- 5 tentatives max, score stocké dans les cookies.

#### 5. **\*\*Tableau des mauvaises réponses\*\*** :

- Affiche chaque tentative incorrecte avec un numéro (ex. "1 - Bob", "2 - Liam").

---

### ### Exemple de déroulement

- **Username** : "Alice".

- **Choix** : "nom".

- Système choisit : "Emma".

- Tentatives :

1. "Bob" → "Incorrect !" (Tableau : "1 - Bob").

2. "Liam" → "Incorrect ! Indice : La 2e lettre est 'm.'" (Tableau : "1 - Bob", "2 - Liam").

3. "Sam" → "Incorrect !" (Tableau : "1 - Bob", "2 - Liam", "3 - Sam").

4. "Grace" → "Incorrect ! Indice : La dernière lettre est 'a.'" (Tableau : "1 - Bob", "2 - Liam", "3 - Sam", "4 - Grace").

5. "Olivia" → "Perdu ! Le kidnappeur était : Emma" (Tableau : "1 - Bob", "2 - Liam", "3 - Sam", "4 - Grace", "5 - Olivia").

- Cookies : `Alice\_attempts`=5`, `Alice\_score`=0`.

- **Choix** : "nombre".

- Système choisit : 42.

- Tentatives :

1. 50 → "Trop grand !" (Tableau : "1 - 50").

2. 30 → "Trop petit !" (Tableau : "1 - 50", "2 - 30").

3. 42 → "Gagné ! Score : 60" (Tableau : "1 - 50", "2 - 30").

- Cookies : `Alice\_attempts`=2`, `Alice\_score`=60`.

---

### ### Déploiement

Déployez sur Tomcat à `http://localhost:8080/projet-jeu/JeuServlet`. Les sessions séparent les jeux par utilisateur, les cookies stockent les résultats, et le tableau affiche les erreurs.

Si vous avez besoin d'ajustements ou d'explications supplémentaires, dites-le-moi !

