

Ex 8	JSP application using scripting elements, EL and JSTL
-------------	--

Aim

To create a JSP application using scripting elements, EL for data display, and JSTL for conditional and loop-based rendering.

Definitions

JSP

JSP, or Jakarta Server Pages (formerly JavaServer Pages), is a server-side technology used to create dynamic web pages by embedding Java code within HTML. It allows developers to separate presentation logic (HTML) from business logic (Java), making it easier to manage and update web applications.

Scripting Elements

Scripting elements are code blocks or tags used to insert code, like Java or JavaScript, into a document to add dynamic behavior. In web development, they are used to perform calculations, manipulate objects, and generate content that changes based on user actions or other data.

EL

The JSP Expression Language (EL) is a powerful feature in JavaServer Pages (JSP) that simplifies access to application data stored in various scopes and JavaBeans components. It was introduced in JSP 2.0 to provide a more concise and readable way to embed dynamic content in JSPs, reducing the need for scriptlets and Java code directly within the page.

JSTL

JSTL, or JavaServer Pages Standard Tag Library, is a set of custom JSP tags that provide common functionality for web development, such as conditional statements, iteration, database access (SQL), and internationalization. It was created to standardize common tasks in JSP pages, reducing the need for developers to write custom tags or use Java code directly in their JSPs, which improves code readability and consistency.

Procedure

Open NetBeans IDE.

To create a Project go to File Menu → choose New Project → choose Java Web from Categories → choose Web Application from Projects → click next → specify the project name as JSTL-EL → Click Next → Click Next → click Finish.

Expand the project JSTL-EL → Right click on Libraries folder → Select Add Library → Choose JSTL 1.2.1 from the list → Click Add Library.

Right Click on Web Pages Folder → new → JSP → specify the file name as index → click finish → type the following codes:

Index.jsp

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

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

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>JSP Application</title>
```

```
</head>
```

```
<body>
```

```
    <h1>JSP Application Example</h1>
```

```
<!-- Scripting Element: Declaration --%>
```

```
<%! int visitorCount = 0; %>
```

```
<!-- Scripting Element: Scriptlet --%>
```

```
<%
```

```
    visitorCount++;
```

```
    String message = "Welcome to our JSP page!";
```

```
    request.setAttribute("username", "John Doe");
```

```
    request.setAttribute("items", new String[]{"Apple", "Banana", "Orange"});
```

```
    request.setAttribute("isAdmin", true);
```

```
%>
```

```

<%-- Scripting Element: Expression --%>

<p>Current visitor count: <%= visitorCount %></p>

<p>Current time: <%= new java.util.Date() %></p>


<%-- EL for Data Display --%>

<p>${message}</p>

<p>User: ${username}</p>


<%-- JSTL for Conditional Rendering --%>

<c:if test="${isAdmin}">
    <p>You have administrative privileges.</p>
</c:if>

<c:choose>
    <c:when test="${ visitorCount % 2 == 0}">
        <p>This is an even-numbered visit.</p>
    </c:when>
    <c:otherwise>
        <p>This is an odd-numbered visit.</p>
    </c:otherwise>
</c:choose>


<%-- JSTL for Loop-based Rendering --%>

<h2>Available Items:</h2>

<ul>
    <c:forEach var="item" items="${items}">
        <li>${item}</li>
    </c:forEach>
</ul>

</body>

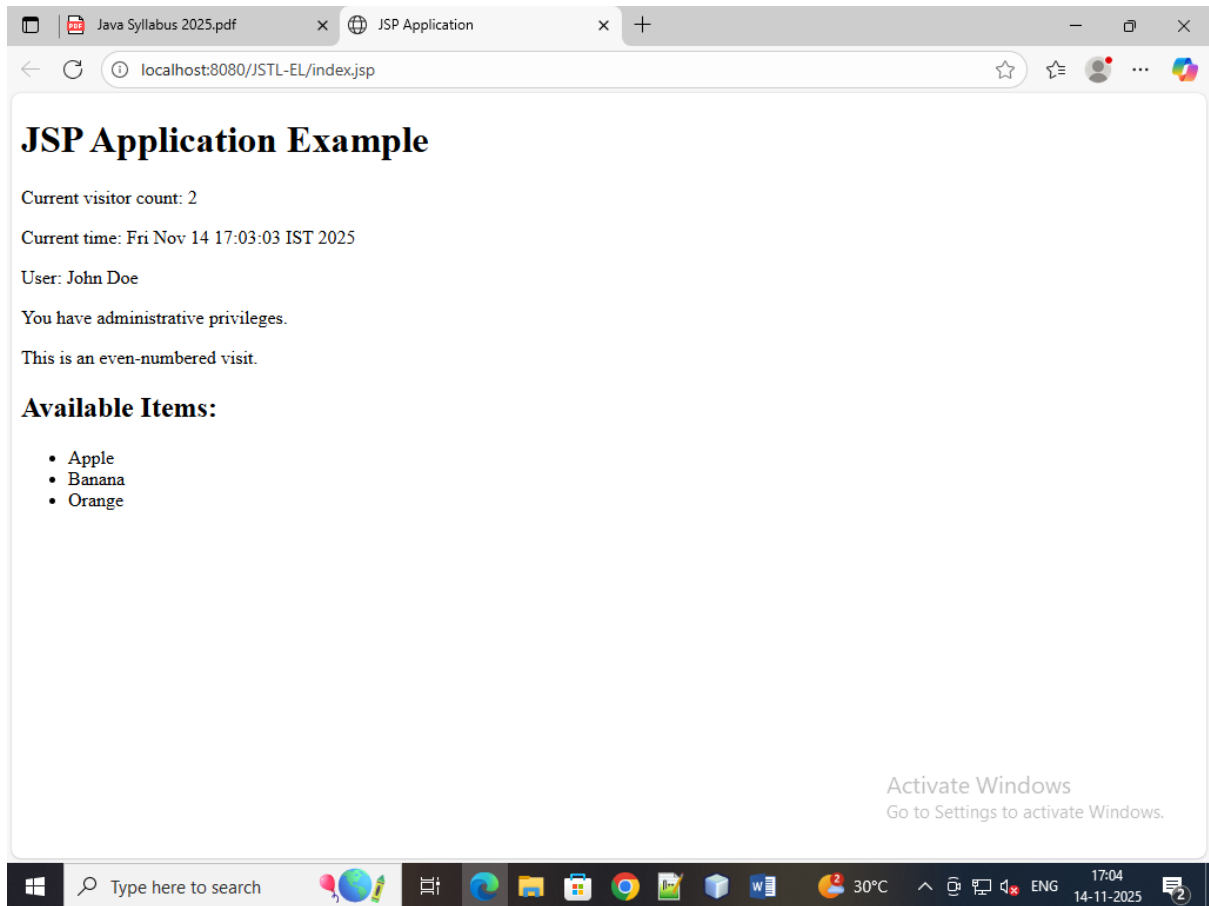
```

</html>

Right click on index.jsp → choose Run File.

Output

You can see the following result in the Browser.



Result

Thus, a JSP application using scripting elements, EL for data display, and JSTL for conditional and loop-based rendering has been created.