

c) Aim: Event bubbling

```
<html>
<head>
  <title>Event Bubbling</title>
  <style>
    .parent {
      background-color: lightblue;
      padding: 20px;
    }

    .child {
      background-color: lightgreen;
      padding: 10px;
    }
  </style>
  <script>
    function handleClick(event) {
      var target = event.target || event.srcElement;
      var id = target.id;
      var message = id === "parent" ? "Outer" : "Inner";
      document.getElementById("result").innerHTML = "Clicked: " + message;
    }
  </script>
</head>
<body>
  <h1>Event Bubbling</h1>
  <p>Name :G Reguvel</p>
  <p>Roll No.: Y21CS043</p>
  <div class="parent" id="parent" onclick="handleClick(event)">
    <p>Parent Element (Click Me)</p>
    <div class="child" id="child" onclick="handleClick(event)">
      <p>Child Element (Click Me)</p>
    </div>
  </div>

  <p id="result"></p>
</body>
</html>
```

```
▼<student>  
  <name>NAME</name>  
  <rollno>REGD.NO</rollno>  
  <year>II/IV BTech</year>  
  <college>RVR&JCCE</college>  
</student>
```

LABCYCLE ~3

PROGRAM – 1: Write a valid XML document using DTD

STEPS:

1. Open a text editor or an XML editor.
2. Create a new file with a .xml extension.
3. Add the XML declaration at the beginning of the file: `<?xml version="1.0" encoding="UTF-8"?>`.
4. Define the structure of the XML document using the DTD syntax.
5. Add the `<!DOCTYPE>` declaration at the beginning of the XML file to link the DTD file to the XML file.
6. Define the structure of the XML document in the DTD file using the DTD syntax.
7. Save the file with a .dtd extension.
8. Copy the contents of the DTD file and paste them into the XML file after the `<!DOCTYPE>` declaration.
9. Open a web browser.
10. Drag and drop the XML file into the browser window.
11. If the XML document is valid, the browser will display the contents of the XML file. If the XML document is invalid, the browser will display an error message.

SOURCE CODE:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE student[
<!ELEMENT student (name, rollno, year, college)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT rollno (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT college (#PCDATA)>
]>

<student>
<name>NAME</name>
<rollno>REGD.NO</rollno>
<year>II/IV BTech</year>
<college>RVR&JCCE</college>
</student>
```

Regd. No.: Y21CS043
Output:

NAME:

Reguvel

PASS:

.....



login

Login Success...

PROGRAM – 2: Write a servlet program to validate a user**STEPS:**

1. Open NetBeans IDE.
2. Create a new project or open an existing project.
3. Right-click on the project name in the **Projects** window and select **New > Servlet**.
4. Enter a name for the servlet and click **Finish**.
5. Override the **doGet()** or **doPost()** method to handle the HTTP request and validate the user input.
6. Build the project by selecting **Build Project** from the **Run** menu.
7. Deploy the servlet to a web server or a servlet container such as Apache Tomcat.
8. Test the servlet by sending an HTTP request to its URL using a web browser or a tool such as **curl**.

SOURCE CODE:**Index.jsp:**

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>JSP Page</title>
</head>
<body>
  <form action="./log" method="post">
    <table>
      <tr>
        <td><b>NAME:</b></td>
        <td><input type="text" name="username"></td>
      </tr>
      <tr>
        <td><b>PASS:</b></td>
        <td><input type="password" name="passcode"></td>
      </tr>
    </table>
    <input type="submit" value="log">
  </form>
</body>
</html>
```

Log.java:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package login;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class log extends HttpServlet {
protected void doPost(HttpServletRequest req,HttpServletResponse res) throws ServletException, IOException {
    res.setContentType("text/html");
    PrintWriter pw=res.getWriter();
    res.setContentType("text/html");
    String user=req.getParameter("username");
    String pass=req.getParameter("passcode");
    if(user.equals("aditya") &&pass.equals("aditya"))
        pw.println("Login Success...!");
    else
        pw.println("Login Failed...!");
    pw.close();}
}
```

Output:

NAME:

Reguvel

PASS:

.....



login

STEPS:

1. Open NetBeans IDE.
2. Create a new project or open an existing project.
3. Right-click on the project name in the **Projects** window and select **New > JSP**.
4. Enter a name for the JSP file and click **Finish**.
5. Write the JSP code using the implicit objects such as **request**, **response**, **session**, **application**, **out**, **config**, and **pageContext**.
6. Save the JSP file in the **Web Pages** folder of the project.
7. Deploy the project to a web server or a servlet container such as Apache Tomcat.
8. Test the JSP by sending an HTTP request to its URL using a web browser or a tool such as **curl**.

SOURCE CODE:LOGIN.HTML:

```
<!DOCTYPE html>
<html>
<head>
<title>JSP Login</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<h1 align="center">Login Page Using JSP</h1>
<form method="POST" action="./Valid.jsp" , " autocomplete="off">
<label>Username: </label>
<input type="text" name="NAME">
<br /><br />
<label>Password: </label>
<input type="password" name="PASS">
<br /><br />
<input type="submit" value="Login">&nbsp;
<input type="reset" value="Clear">
</form>
</body>
</html>
```

VALID.JSP:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Implicit Objects</title>
</head>
<body>
<%
    String uname = request.getParameter("username");
    String pwd = request.getParameter("passwd");

    if (uname.equalsIgnoreCase("aditya") && pwd.equalsIgnoreCase("aditya"))
    {
        out.println("<h1>Login success</h1>");
    }
    else
    {
        out.println("<h1>Login failed</h1>");
    }
%>
</body>
</html>
```

Action Tag Example

Name:	<input type="text" value="Reguvel"/>
Email:	<input type="text" value="reguvel07@gmail.com"/>
<input type="submit" value="Submit"/>	

Form submitted Successfully

Name: Reguvel

Email: reguvel07@gmail.com

PROGRAM – 4: Write a JSP program on Action tags.

STEPS:

1. Open NetBeans IDE.
2. Create a new project or open an existing project.
3. Right-click on the project name in the **Projects** window and select **New > JSP**.
4. Enter a name for the JSP file and click **Finish**.
5. Write the JSP code using the action tags such as `<jsp:useBean>`, `<jsp:setProperty>`, `<jsp:getProperty>`, `<jsp:forward>`, and `<jsp:include>`.
6. Save the JSP file in the **Web Pages** folder of the project.
7. Deploy the project to a web server or a servlet container such as Apache Tomcat.
8. Test the JSP by sending an HTTP request to its URL using a web browser or a tool such as **curl**.

SOURCE CODE:

ACC.JSP:

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<title>Action Tag Example</title>
</head>
<body>
<h1>Action Tag Example</h1>

<form action="act.jsp" method="post">
<label for="name">Name:</label>
<input type="text" name="name" id="name" required>
<br>
<label for="email">Email:</label>
<input type="email" name="email" id="email" required>
<br>
<input type="submit" value="Submit">
</form>

</body>
</html>
```

ACT.JSP:

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<title>ACT JSP</title>
</head>
<body>
```