Program 1 : RMI

- ⇨ set path = C:\jdk1.3\bin;
- ⇨ javac *.java
- ⇨ rmic FactImpl
- ⇨ start rmiregistry
- ⇨ java FactServer
- ⇨ java FactClient

Fact.java

```
import java.rmi.Remote;
import java.rmi.RemoteException;

public interface Fact extends Remote {
   int factorial(int n) throws RemoteException;
}
```

FactClient.java

```
import java.rmi.Naming;
import java.util.Scanner;

public class FactClient {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);

      try {
         Fact fact = (Fact) Naming.lookup("//localhost/FactService");
                  System.out.print("Enter a number to calculate its factorial: ");
         int number = scanner.nextInt();
                  int result = fact.factorial(number);

         System.out.println("The factorial of " + number + " is " + result);
      } catch (Exception e) {
         System.out.println("FactClient exception: " + e.getMessage());
         e.printStackTrace();
      } finally {
         scanner.close();
      }
   }
}
```

FactImpl.java

```
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;
public class FactImpl extends UnicastRemoteObject implements Fact {
   protected FactImpl() throws RemoteException {
      super();
```

```
    }
    @Override
    public int factorial(int n) throws RemoteException {
        if (n == 0) return 1;
        return n * factorial(n - 1);
    }
}
```

FactServer.java

```
import java.rmi.Naming;
import java.rmi.registry.LocateRegistry;

public class FactServer {
    public static void main(String[] args) {
        try {
            LocateRegistry.createRegistry(1981);

            FactImpl factImpl = new FactImpl();

            // Bind the implementation to the RMI registry with a name
            Naming.rebind("FactService", factImpl);

            System.out.println("FactServer is ready.");
        } catch (Exception e) {
            System.out.println("FactServer exception: " + e.getMessage());
            e.printStackTrace();
        }
    }
}
```

****

Program 2 : SERVLET

- **set path=C:\jdk1.3\bin;**
- **set classpath=%classpath%;C:\SERVLET\jsdk2.1\servlet.jar;.;**
- form action=**http://localhost:9000/examples/servlet/Register**
- Paste the class file in the directory (**T:\SERVLET\jsdk2.1\examples\WEB-INF\servlets**)
- **startserver.bat**
- **run the register.html file**

Register.java

```
import java.io.*;
import java.sql.*;
import javax.servlet.ServletException;
import javax.servlet.http.*;

public class Register extends HttpServlet {
   public void doPost(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {

      response.setContentType("text/html");
      PrintWriter out = response.getWriter();

      String n = request.getParameter("userName");
      String p = request.getParameter("userPass");
      String e = request.getParameter("userEmail");
      String c = request.getParameter("userCountry");

      try {
         Class.forName("oracle.jdbc.OracleDriver");
         Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@172.16.139.18:1521:ORCL", "ss", "ss");
         PreparedStatement ps = con.prepareStatement("insert into registeruser values(?,?,?,?)");
         ps.setString(1, n);
         ps.setString(2, p);
         ps.setString(3, e);
         ps.setString(4, c);

         int i = ps.executeUpdate();
         if (i > 0) {
            out.println("<p style='color:green;'>You are successfully registered...</p>");
            fetchAndDisplayUsers(out); // Display updated user list after registration
         } else {
            out.println("<p style='color:red;'>Failed to register user</p>");
         }

      } catch (Exception ex) {
```

```
            out.println("<p style='color:red;'>Error: " + ex.getMessage() + "</p>");
        } finally {
            out.close();
        }
    }

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

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        fetchAndDisplayUsers(out);

        out.close();
    }

    private void fetchAndDisplayUsers(PrintWriter out) {
        try {
            Class.forName("oracle.jdbc.OracleDriver");
            Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@172.16.139.18:1521:ORCL", "pg23pca101",
"pg23pca101");
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery("select * from registeruser");

            out.println("<table border='1'>");
            out.println("<tr><th>Name</th><th>Email</th><th>Country</th></tr>");

            while (rs.next()) {
                String userName = rs.getString("username");
                String userEmail = rs.getString("useremail");
                String userCountry = rs.getString("usercountry");
                out.println("<tr><td>" + userName + "</td><td>" + userEmail + "</td><td>" +
userCountry + "</td></tr>");
            }

            out.println("</table>");

            con.close();
        } catch (Exception ex) {
            out.println("<p style='color:red;'>Error: " + ex.getMessage() + "</p>");
        }
    }
}
```

Register.html

```
<!DOCTYPE html>
<html>
<head>
   <title>User Registration</title>
</head>
<body>
   <form id="registrationForm" action="http://localhost:8050/examples/servlet/Register"
method="post">
      <h2>User Registration Form</h2>
      Name: <input type="text" name="userName" required><br/><br/>
      Password: <input type="password" name="userPass" required><br/><br/>
      Email Id: <input type="email" name="userEmail" required><br/><br/>
      Country:
      <select name="userCountry">
         <option>India</option>
         <option>Pakistan</option>
         <option>Other</option>
      </select><br/><br/>
      <input type="submit" value="Register">
   </form>

</body>
</html>
```

Program 3 : JSP

http://sccsjc:8080/pg23pca101/Employee.jsp

```
<%@ page import="java.sql.*" %>
<%
   Connection conn = null;
   Statement stmt = null;
   ResultSet rs = null;

   // Get action and parameters from request
   String action = request.getParameter("action");
   String id = request.getParameter("id");
   String name = request.getParameter("name");

   try {
      // Initialize driver class
      Class.forName("oracle.jdbc.driver.OracleDriver");
   } catch (Exception e) {
      out.println("Failed to initialize Oracle JDBC driver: " + e.toString() + "<P>");
   }

   String dbUser = "pg23pca101";
   String dbPasswd = "pg23pca101";
   String dbURL = "jdbc:oracle:thin:@172.16.139.18:1521:ORCL";

   try {
      // Connect
      conn = DriverManager.getConnection(dbURL, dbUser, dbPasswd);
      conn.setAutoCommit(false); // Disable auto-commit mode
     // out.println("Connection status: " + conn + "<P>");

      // Create statement
      stmt = conn.createStatement();

      if ("insert".equals(action)) {
         // Insert
         try {
            String sql = "insert into employee (id, name) values (" + id + ", '" + name + "')";
            int numRowsAffected = stmt.executeUpdate(sql);
            out.println(numRowsAffected + " employee(s) inserted. <BR>");
         } catch (SQLException e) {
            out.println("Error encountered during row insertion for employee: " + e.toString() +
"<BR>");
         }
      } else if ("update".equals(action)) {
         // Update
         try {
            String sql = "update employee set name='" + name + "' where id=" + id;
            int numRowsAffected = stmt.executeUpdate(sql);
```

```
            out.println(numRowsAffected + " employee(s) updated. <BR>");
        } catch (SQLException e) {
            out.println("Error encountered during update for employee: " + e.toString() + "<BR>");
        }
    } else if ("delete".equals(action)) {
        // Delete
        try {
            String sql = "delete from employee where id=" + id;
            int numRowsAffected = stmt.executeUpdate(sql);
            out.println(numRowsAffected + " employee(s) deleted. <BR>");
        } catch (SQLException e) {
            out.println("Error encountered during deletion of employee: " + e.toString() + "<BR>");
        }
    }

    // Select (View all)
    String sql = "select id, name from employee";
    rs = stmt.executeQuery(sql);
    while (rs.next()) {
        out.println("Id = " + rs.getString("ID") + ", Name = " + rs.getString("NAME") + "<BR>");
    }
    out.println("<P>");

    // Commit
    conn.commit();
  } catch (Exception e) {
    out.println("Connection failed: " + e.toString() + "<P>");
    if (conn != null) {
        try {
            conn.rollback();
        } catch (SQLException rollbackException) {
            out.println("Rollback failed: " + rollbackException.toString() + "<P>");
        }
    }
  } finally {
    try {
        if (rs != null) rs.close();
        if (stmt != null) stmt.close();
        if (conn != null) conn.close();
    } catch (SQLException e) {
        out.println("Error closing resources: " + e.toString() + "<P>");
    }
  }
%>
```

Index.html

```
<HTML>
<BODY>
  <form method="post" action="employee.jsp">
    Action: <select name="action">
      <option value="insert">Insert</option>
```

```
            <option value="update">Update</option>
            <option value="delete">Delete</option>
        </select><br>
        ID: <input type="text" name="id"><br>
        Name: <input type="text" name="name"><br>
        <input type="submit" value="Submit">
    </form>
    <br>
    Bye bye! The system time is now <%= new java.util.Date() %>
</BODY>
</HTML>
```

Program 4 : JSP - BEAN

College.java

```
package col;
public class College{
   String name;
   String rector;
   String secretary;
   String principal;

public void setName(String n) {name=n;}
public void setRector(String r) {rector = r;}
public void setSecretary(String s) {secretary = s;}
public void setPrincipal(String p) {principal = p;}

public String getName(){return name;}
public String getRector(){return rector;}
public String getSecretary(){return secretary;}
public String getPrincipal(){return principal;}
}
```

****

Usebean.jsp

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

<jsp:useBean id="college"  class="col.College">
<%-- intialize bean properties --%>
<jsp:setProperty name="college" property="name" value="St. Joseph's College (Autonomous),
Tiruchirappalli-620002"/>
<jsp:setProperty name="college" property="rector" value="Rev. Dr. Pavulraj Michael SJ"/>
<jsp:setProperty name="college" property="secretary" value="Rev. Fr. K. Amal SJ"/>
<jsp:setProperty name="college" property="principal" value="Rev. Dr. S. Mariadoss SJ"/>
</jsp:useBean>

<html>
        <body>
        <h2>
           College.....  <jsp:getProperty name="college" property="name"/><br>
Rector.....    <jsp:getProperty name="college" property="rector"/><br>
Secretary.....    <jsp:getProperty name="college" property="secretary"/><br>
           Principal.....  <jsp:getProperty name="college" property="principal"/>
        </body>
</html>
```

****

OUTPUT

JSP-Bean Running from the Server (Tomcat 4.0)

- Compile College.java with **JDK 1.3 (set path=C:\jdk1.3\bin;)**
- Create a folder named **"col"** for the package and copying the compiled `College.class` file into this directory is correct. **(Ex:pg23pca101\03_JSP\WEB-INF\classes\col\College.class)**
- However, the folder should be inside the `WEB-INF/classes` directory of your web application to ensure that Tomcat can find and load the class.
- Finally run the JSP file from the server **"http://sccsjc:8080/pg23pca101/Usebean.jsp"**

5. XSLT

transform.xslt

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" indent="yes"/>
    <!-- Root template -->
  <xsl:template match="/">
    <html>
      <head>
        <title>Library Book List</title>
        <style>
          body { font-family: Arial, sans-serif; margin: 20px; }
          table { width: 100%; border-collapse: collapse; }
          th, td { border: 1px solid #ddd; padding: 8px; }
          th { background-color: #f2f2f2; }
        </style>
      </head>
      <body>
        <h1>Library Book List</h1>
        <table>
          <tr>
            <th>Title</th>
            <th>Author</th>
            <th>Year</th>
          </tr>
          <xsl:apply-templates select="library/book"/>
```

```
            </table>
        </body>
    </html>
</xsl:template>

<!-- Template for book elements -->
<xsl:template match="book">
    <tr>
        <td><xsl:value-of select="title"/></td>
        <td><xsl:value-of select="author"/></td>
        <td><xsl:value-of select="year"/></td>
    </tr>
</xsl:template>
</xsl:stylesheet>
```

data.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<library>
    <book>
        <title>The Great Gatsby</title>
        <author>F. Scott Fitzgerald</author>
        <year>1925</year>
    </book>
    <book>
        <title>To Kill a Mockingbird</title>
        <author>Harper Lee</author>
        <year>1960</year>
```

```
    </book>
    <book>
        <title>1984</title>
        <author>George Orwell</author>
        <year>1949</year>
    </book>
</library>
```

index.jsp

```
<%@ page import="javax.xml.transform.TransformerFactory,
javax.xml.transform.Transformer, javax.xml.transform.stream.StreamSource,
javax.xml.transform.stream.StreamResult, java.io.FileInputStream,
java.io.StringWriter, java.io.File" %>
<%@ page contentType="text/html;charset=UTF-8" %>
<!DOCTYPE html>
<html>
<head>
    <title>Library Book List</title>
</head>
<body>

    <%
        try {
            // Get the real path to the XML and XSLT files
            String xmlPath = application.getRealPath("/data.xml");
            String xsltPath = application.getRealPath("/transform.xslt");
```

```
// Check if the paths are valid
        if (xmlPath == null || xsltPath == null) {
            throw new RuntimeException("File paths are null. Ensure that the files are in the correct location.");
        }
                // Create File objects for the paths
        File xmlFile = new File(xmlPath);
        File xsltFile = new File(xsltPath);

        // Check if files exist
        if (!xmlFile.exists()) {
            throw new RuntimeException("XML file does not exist: " + xmlPath);
        }
        if (!xsltFile.exists()) {
            throw new RuntimeException("XSLT file does not exist: " + xsltPath);
        }
        // Create a TransformerFactory and load the XSLT file
        TransformerFactory factory = TransformerFactory.newInstance();
        javax.xml.transform.Source xmlSource = new StreamSource(new FileInputStream(xmlFile));
        javax.xml.transform.Source xsltSource = new StreamSource(new FileInputStream(xsltFile));
        StringWriter resultWriter = new StringWriter();

        Transformer transformer = factory.newTransformer(xsltSource);
        transformer.transform(xmlSource, new StreamResult(resultWriter));
```

```
            // Output the result
            out.println(resultWriter.toString());
        } catch (Exception e) {
            // Handle exceptions and print error messages
            e.printStackTrace();
            out.println("Error: " + e.getMessage());
        }
    %>
</body>
</html>
```

Output

Library Book List

Title	Author	Year
The Great Gatsby	F. Scott Fitzgerald	1925
To Kill a Mockingbird	Harper Lee	1960
1984	George Orwell	1949

6. Department Website

data.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<department>
  <staff>
    <member>Dr. L. Arockiam</member>
    <member>Dr. S. Britto Ramesh Kumar</member>
    <member>Dr. A. Aloysius</member>
    <member>Dr. V. Jude Nirmal</member>
    <member>Rev. Dr. S. Arul Oli SJ</member>
    <member>Rev. Dr. S. Santiago SJ</member>
    <member>Dr. A. Vimal Jearld</member>
    <member>Dr. George Gabriel Richard Roy</member>
    <member>Dr. A. Angel Preethi</member>
    <member>Dr. G. Arockiya Sagaya Sheela</member>
    <member>Dr. P. Joseph Charles</member>
    <member>Dr. K. Maheswaran</member>
  </staff>
  <motto>Pro Bono Et Vero</motto>
  <syllabus>
    <course>
      <code>CS101</code>
      <name>Introduction to Computer Science</name>
    </course>
    <course>
      <code>CS102</code>
      <name>Data Structures and Algorithms</name>
    </course>
    <course>
      <code>CS201</code>
      <name>Database Management Systems</name>
    </course>
    <course>
      <code>CS301</code>
      <name>Software Engineering</name>
    </course>
    <course>
```

```
        <code>CS302</code>
        <name>Operating Systems</name>
      </course>
    </syllabus>
    <contact>
      <email>mca@example.com</email>
      <phone>+1234567890</phone>
    </contact>
</department>
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title> Department  of Computer Science</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <header>
        <h1>Department of Computer Science (MCA)</h1>
        <nav>
            <ul>
                <li><a href="index.html">Home</a></li>
                <li><a href="staff.html">Staff List</a></li>
                <li><a href="syllabus.html">Syllabus</a></li>
                <li><a href="contact.html">Contact Us</a></li>
            </ul>
        </nav>
    </header>
    <main>
        <section id="home">
            <h2>St. Joseph's College (Autonomous), Tiruchirappalli-620002</h2>
            <section id="department-profile">
```

<h2>Profile of the Department</h2>

The Department of Computer Science was started in the year 1983. The Department offers B.Sc. (Computer Science) from 1983, MCA from 1984, M.Phil. and Ph.D. programmes from 2002. St. Joseph`s College was the first Arts and Science College to offer MCA in India and first to produce the PhD in Computer Science in the Bharathidasan University. There are 11 aided and 14 unaided staff members. Among 17 PhD holders, 7 staff members are guiding research scholars for Ph.D, 4 staff members are qualified NET and 6 qualified SET. The faculty of the Department serve as members of inspection committee of the Bharathidasan University for approval of Computer Courses, members of staff selection committee of other colleges, members of organizing committee of various conferences of international and national level and members of editorial board of research journals. In 2005 and 2007 a national level conference was organized. In 2014 and 2016 International Conferences were held. The Department has produced 15 Ph.Ds. Seven minor research projects were completed, funded by UGC. To adhere to the motto of the Department "Professional Excellence", the Training and Placement Cell of the Department has helped to place 53 students in 2007, 2 in 2008, 9 in 2009, 23 in 2010 and 23 in 2011 in various MNCs. The companies like Asian Paints, Patni Computer Systems, Fulcrum Technologies, HCL, Infosys, Wipro Technologies, Satyam, TCS, Sutherland Global Services, iGate, Savvion, NetMagnus, Kumaran Systems and ISG Novasoft prefer to provide project cum placement to our students for their values and professional commitment.

```
        </section>
    </main>
    <script>
        // Optionally, you can fetch XML data here if needed
    </script>
</body>
</html>
```

staff.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>MCA Department - Staff List</title>
  <link rel="stylesheet" href="styles.css">
  <script>
    document.addEventListener('DOMContentLoaded', function() {
      fetch('data.xml')
        .then(response => response.text())
        .then(data => {
          const parser = new DOMParser();
          const xmlDoc = parser.parseFromString(data, 'text/xml');
          const staffList = xmlDoc.getElementsByTagName('member');
          const staffSection = document.getElementById('staff-list');
          staffSection.innerHTML = ''; // Clear existing content
          Array.from(staffList).forEach(member => {
            const memberName = member.textContent || 'N/A';
            const listItem = document.createElement('li');
            listItem.textContent = memberName;
            staffSection.appendChild(listItem);
          });
        })
        .catch(error => {
          console.error('Error fetching or parsing XML data:', error);
          document.getElementById('staff-list').innerHTML = 'Failed to load
staff data.';
        });
    });
  </script>
</head>
<body>
  <header>
    <h1>Department of Computer Science (MCA)</h1>
    <nav>
      <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="staff.html">Staff List</a></li>
        <li><a href="syllabus.html">Syllabus</a></li>
        <li><a href="contact.html">Contact Us</a></li>
```

```
        </ul>
      </nav>
    </header>
    <main>
      <section id="staff">
        <h2>Staff Members</h2>
        <ul id="staff-list">
          <!-- Staff names will be inserted here by JavaScript -->
        </ul>
      </section>
    </main>
</body>
</html>
```

syllabus.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>MCA Department - Syllabus</title>
    <link rel="stylesheet" href="styles.css">
    <script>
        document.addEventListener('DOMContentLoaded', function() {
            fetch('data.xml')
                .then(response => response.text())
                .then(data => {
                    const parser = new DOMParser();
                    const xmlDoc = parser.parseFromString(data, 'text/xml');
                    const courses = xmlDoc.getElementsByTagName('course');
                    const syllabusSection = document.getElementById('syllabus-list');
                    syllabusSection.innerHTML = ''; // Clear existing content
                    Array.from(courses).forEach(course => {
```

```
            const code = course.getElementsByTagName('code')[0]?.textContent || 'N/A';
            const name = course.getElementsByTagName('name')[0]?.textContent || 'N/A';
            const courseDiv = document.createElement('div');
            courseDiv.className = 'course';
            courseDiv.innerHTML = `<strong>Code:</strong> ${code} <br><strong>Name:</strong> ${name}`;
            syllabusSection.appendChild(courseDiv);
          });
        })
        .catch(error => {
          console.error('Error fetching or parsing XML data:', error);
          document.getElementById('syllabus-list').innerHTML = 'Failed to load syllabus data.';
        });
    });
  </script>
</head>
<body>
  <header>
    <h1>Department of Computer Science (MCA)</h1>
    <nav>
      <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="staff.html">Staff List</a></li>
        <li><a href="syllabus.html">Syllabus</a></li>
        <li><a href="contact.html">Contact Us</a></li>
      </ul>
    </nav>
  </header>
  <main>
    <section id="syllabus">
      <h2>Syllabus</h2>
      <div id="syllabus-list">
        <!-- Course details will be inserted here by JavaScript -->
      </div>
```

```
    </section>
  </main>
</body>
</html>
```

contact.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>MCA Department - Contact Us</title>
  <link rel="stylesheet" href="styles.css">
  <script>
    document.addEventListener('DOMContentLoaded', function() {
      fetch('data.xml')
        .then(response => {
          if (!response.ok) {
            throw new Error('Network response was not ok: ' + response.statusText);
          }
          return response.text();
        })
        .then(data => {
          const parser = new DOMParser();
          const xmlDoc = parser.parseFromString(data, 'text/xml');

          // Fetch and display contact information
          const email = xmlDoc.getElementsByTagName('email')[0]?.textContent || 'N/A';
          const phone = xmlDoc.getElementsByTagName('phone')[0]?.textContent || 'N/A';
          document.getElementById('contact-email').textContent = email;
          document.getElementById('contact-phone').textContent = phone;
        })
        .catch(error => {
          console.error('Error fetching or parsing XML data:', error);
```

```
            document.getElementById('contact-info').innerHTML = 'Failed to load contact information.';
        });
    });
  </script>
</head>
<body>
  <header>
    <h1>Department of Computer Science (MCA)</h1>
    <nav>
      <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="staff.html">Staff List</a></li>
        <li><a href="syllabus.html">Syllabus</a></li>
        <li><a href="contact.html">Contact Us</a></li>
      </ul>
    </nav>
  </header>
  <main>
    <section id="contact">
      <h2>Contact Us</h2>
      <div id="contact-info">
        <p><strong>Email:</strong> <span id="contact-email"></span></p>
        <p><strong>Phone:</strong> <span id="contact-phone"></span></p>
      </div>
    </section>
  </main>
</body>
</html>
```

styles.css

```
/* General Reset */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
```

```
body {
    font-family: Arial, sans-serif;
    line-height: 1.6;
    color: #333;
    background-color: #f4f4f4;
    padding: 20px; /* Added padding for better spacing */
}

header {
    background-color: #333;
    color: #fff;
    padding: 20px;
    text-align: center;
}

header h1 {
    margin-bottom: 10px;
}

nav ul {
    list-style: none;
    padding: 0;
}

nav ul li {
    display: inline;
    margin-right: 15px;
}

nav ul li a {
    color: #fff;
    text-decoration: none;
    font-weight: bold;
    transition: color 0.3s;
}
nav ul li a:hover {
```

```
    color: #f4f4f4;
}
main {
    padding: 20px;
    background: #fff;
    max-width: 1200px;
    margin: 20px auto;
    box-shadow: 0 0 15px rgba(0, 0, 0, 0.1);
    border-radius: 8px;
}
section {
    margin-bottom: 20px;
}
h2 {
    color: #444;
    border-bottom: 2px solid #333;
    padding-bottom: 10px;
    margin-bottom: 20px;
    text-align: center; /* Center-aligns the heading */
}
p#motto {
    text-align: justify; /* Justify text alignment for paragraphs */
    margin: 0;
    font-size: 1.1em;
    line-height: 1.8;
    padding: 0 10px; /* Added padding for better readability */
}
ul {
    list-style-type: disc;
    padding-left: 20px;
}

ul li {
    margin-bottom: 10px;
}
.course {
    background: #e9ecef;
```

```
    border: 1px solid #ddd;
    border-radius: 5px;
    padding: 15px;
    margin-bottom: 15px;
}
.course strong {
    display: block;
    margin-bottom: 5px;
}

footer {
    background-color: #333;
    color: #fff;
    padding: 10px;
    text-align: center;
    position: fixed;
    width: 100%;
    bottom: 0;
}

@media (max-width: 768px) {
    nav ul {
        text-align: center;
    }

    nav ul li {
        display: block;
        margin-bottom: 10px;
    }

    /* Ensure main content is responsive */
    main {
        padding: 10px;
    }

    /* Ensure h2 heading is responsive */
    h2 {
```

```
        font-size: 1.5em;
    }
}
```

7. AJAX (Asynchronous JavaScript and XML)

index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<title>Product Calculator</title>
<script>
functioncalculateTotal() {
varproductName = document.getElementById('productName').value;
var quantity = document.getElementById('quantity').value;
var price = document.getElementById('price').value;
// AJAX Script
varxhr = new XMLHttpRequest();  //XMLHttpRequest Object
xhr.open('POST', 'calculateTotal.jsp', true); // Request is Asynchronous
xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');

xhr.onreadystatechange = function() {//Event Handling
if (xhr.readyState === 4 &&xhr.status === 200) {
                // Extract only the total amount from the response
varresponseText = xhr.responseText.trim();// Handling the Response
document.getElementById('totalAmount').value = responseText;
            }
        };

varparams = 'productName=' + encodeURIComponent(productName) +
```

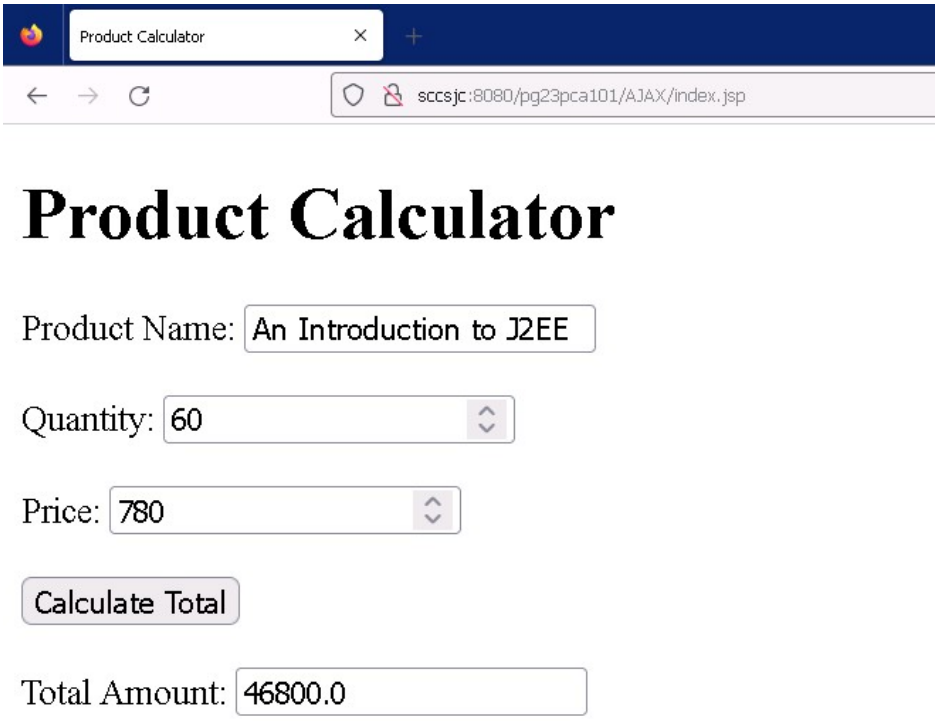
```
                    '&quantity=' + encodeURIComponent(quantity) +
                    '&price=' + encodeURIComponent(price);
xhr.send(params);// Sending Data
        }
</script>
</head>
<body>
<h1>Product Calculator</h1>
<form id="productForm" onsubmit="event.preventDefault();
calculateTotal();">
<label for="productName">Product Name:</label>
<input type="text" id="productName" name="productName"
required><br><br>
<label for="quantity">Quantity:</label>
<input type="number" id="quantity" name="quantity" required
min="1"><br><br>
<label for="price">Price:</label>
<input type="number" id="price" name="price" required step="0.01"
min="0"><br><br>
<button type="submit">Calculate Total</button>
</form>

<br>
<label for="totalAmount">Total Amount:</label>
<input type="text" id="totalAmount" readonly><br><br>
</body>
</html>
```

CalculateTotalServlet.java

```
importjava.io.IOException;
importjavax.servlet.ServletException;
importjavax.servlet.annotation.WebServlet;
importjavax.servlet.http.HttpServlet;
importjavax.servlet.http.HttpServletRequest;
importjavax.servlet.http.HttpServletResponse;

@WebServlet("/calculateTotal")
public class CalculateTotalServlet extends HttpServlet {
private static final long serialVersionUID = 1L;

   @Override
protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
response.setContentType("application/json");
response.setCharacterEncoding("UTF-8");

      // Read parameters from the request
      String productName = request.getParameter("productName");
int quantity = Integer.parseInt(request.getParameter("quantity"));
double price = Double.parseDouble(request.getParameter("price"));

      // Calculate total amount
doubletotalAmount = quantity * price;
```

```
        // Create JSON response
        String jsonResponse = "{\"productName\": \"" + productName + "\",
\"quantity\": " + quantity + ", \"price\": " + price + ", \"totalAmount\": " +
totalAmount + "}";


        // Send the response
response.getWriter().write(jsonResponse);
    }
}
```

Output :



8. Student Profile

student.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<student>
  <personalInfo>
    <name>John Doe</name>
    <id>123456</id>
    <major>Computer Science</major>
    <year>2023</year>
  </personalInfo>
  <contact>
    <email>john.doe@example.com</email>
    <phone>+1234567890</phone>
  </contact>
  <courses>
    <course>
      <title>Introduction to Computer Science</title>
      <code>CS101</code>
      <credits>3</credits>
    </course>
    <course>
      <title>Data Structures and Algorithms</title>
      <code>CS201</code>
      <credits>4</credits>
    </course>
  </courses>
</student>
```

profile.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Student Profile</title>
  <link rel="stylesheet" href="styles.css">
  <script>
    document.addEventListener('DOMContentLoaded', function() {
      fetch('student.xml')
        .then(response => {
          if (!response.ok) {
            throw new Error('Network response was not ok ' +
response.statusText);
          }
          return response.text();
        })
        .then(data => {
          const parser = new DOMParser();
          const xmlDoc = parser.parseFromString(data, 'text/xml');

          // Personal Info
          const name =
xmlDoc.getElementsByTagName('name')[0]?.textContent || 'N/A';
          const id = xmlDoc.getElementsByTagName('id')[0]?.textContent ||
'N/A';
          const major =
xmlDoc.getElementsByTagName('major')[0]?.textContent || 'N/A';
          const year =
xmlDoc.getElementsByTagName('year')[0]?.textContent || 'N/A';

          document.getElementById('name').textContent = name;
          document.getElementById('id').textContent = id;
          document.getElementById('major').textContent = major;
          document.getElementById('year').textContent = year;

          // Contact Info
          const email =
xmlDoc.getElementsByTagName('email')[0]?.textContent || 'N/A';
          const phone =
xmlDoc.getElementsByTagName('phone')[0]?.textContent || 'N/A';
```

```
            document.getElementById('email').textContent = email;
            document.getElementById('phone').textContent = phone;

            // Courses
            const courses = xmlDoc.getElementsByTagName('course');
            const courseList = document.getElementById('course-list');
            courseList.innerHTML = ''; // Clear existing content

            Array.from(courses).forEach(course => {
              const title =
course.getElementsByTagName('title')[0]?.textContent || 'N/A';
              const code =
course.getElementsByTagName('code')[0]?.textContent || 'N/A';
              const credits =
course.getElementsByTagName('credits')[0]?.textContent || 'N/A';

              const courseDiv = document.createElement('div');
              courseDiv.className = 'course';
              courseDiv.innerHTML = `<h3>${title}</h3><p>Code:
${code}</p><p>Credits: ${credits}</p>`;
              courseList.appendChild(courseDiv);
            });
          })
          .catch(error => {
            console.error('Error fetching XML data:', error);
            document.getElementById('profile-content').innerHTML = 'Failed to
load profile data.';
          });
      });
  </script>
</head>
<body>
  <header>
    <h1>Student Profile</h1>
  </header>
  <main>
    <section id="personal-info">
```

```
        <h2>Personal Information</h2>
        <p><strong>Name:</strong> <span id="name"></span></p>
        <p><strong>ID:</strong> <span id="id"></span></p>
        <p><strong>Major:</strong> <span id="major"></span></p>
        <p><strong>Year:</strong> <span id="year"></span></p>
      </section>
      <section id="contact-info">
        <h2>Contact Information</h2>
        <p><strong>Email:</strong> <span id="email"></span></p>
        <p><strong>Phone:</strong> <span id="phone"></span></p>
      </section>
      <section id="course-info">
        <h2>Courses</h2>
        <div id="course-list"></div>
      </section>
    </main>
</body>
</html>
```