Web Technologies Lab Record

- Question 1: Online Book Store

```
Write from WT Lab Observation
```

- Question 2: Demostrate different types of CSS:

  - Inline

```
<!DOCTYPE html>
<html>
    <head>
        <title>Inline CSS</title>
    </head>
    <body>
        <h1 style="color:blue; text-align:center;">Inline CSS</h1>
        <p style="color:red;">
            An inline style may be used to apply a unique style for a single element.
            To use inline styles, add the style attribute to the relevant element. The style
attribute can contain any CSS property.
        </p>
    </body>
</html>
```

  - Internal

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Internal CSS</title>
        <style>
            body {
                background-color: linen;
            }

            h1 {
                color: maroon;
                text-align: center;
            }
        </style>
    </head>
    <body>
        <h1>Internal CSS</h1>
        <p>
            An internal style sheet maybe used if one single HTML page has a unique style.
            The internal style is defined inside the &lt;style&lt; element, inside the head section.
        </p>
    </body>
</html>
```

- External

external.html

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>External CSS</title>
        <link rel="stylesheet" type="text/css" href="external.css">
```

```
            </head>
            <body>
                <h1>External CSS</h1>
                <p>
                    With an external style sheet, you can change the look of an entire website by changing
just one file!
                    Each HTML page must include a reference to the external style sheet file inside the
&lt;link&lt; element, inside the head section.
                </p>
            </body>
        </html>
```

external.css

```
        body {
            background-color: linen;
        }

        h1 {
            color: navy;
            text-align: center;
        }
```

- Question 3: Validations using Regular Expressions for:

    - Registration Page

```
        <!DOCTYPE html>
        <html lang="en">
            <head>
                <title>Registration Page</title>
```

```html
    <style>
        html, body {
            display: flex;
            height: 100%;
            width: 100%;
            font-family: -apple-system, BlinkMacSystemFont;
        }
        h1 {
            text-align: center;
        }
        .form {
            margin: 20px auto;
        }
        input {
            display: block;
            margin-bottom: 20px;
            width: 300px;
            height: 30px;
            font-size: 14px;
        }
    </style>
</head>
<body>
    <div class="form">
        <h1>Registration Page</h1>
        <form>
            <input type="text" name="name" id="name" placeholder="Name">
            <input type="text" name="email" id="email" placeholder="E-Mail">
            <input type="password" name="pwd" id="pwd" placeholder="Password">
            <input type="text" name="phone" id="phone" placeholder="Phone Number">
            <input type="button" value="Register" onclick="validation()">
        </form>
        <span id="output"></span>
    </div>

    <script>
```

```
                function validation () {
                    var out = document.getElementById('output')
                    var name = document.getElementById('name').value
                    var email = document.getElementById('email').value
                    var pwd = document.getElementById('pwd').value
                    var ph = document.getElementById('phone').value

                    var nameValid = name.match(/^[A-Za-z ]+$/)
                    var emailValid = email.match(/^[\w\.-]+@[\w\.]+[A-Za-z]{2,3}$/)
                    var pwdValid = pwd.match(/^.{6,}$/)
                    var phValid = ph.match(/^\d{10}$/)

                    if(!nameValid) {
                        output.innerHTML = 'Invalid Name'
                    } else if(!emailValid) {
                        output.innerHTML = 'E-Mail is not valid'
                    } else if(!pwdValid) {
                        output.innerHTML = 'Password should be atlest 6 characters long'
                    } else if(!phValid) {
                        output.innerHTML = 'Invalid Phone number'
                    } else {
                        output.innerHTML = 'Welcome, ' + name
                    }
                }
            </script>
        </body>
    </html>
```

- Login Page

```
    <!DOCTYPE html>
    <html lang="en">
        <head>
            <title>Login Page</title>
```

```html
        <style>
            html, body {
                display: flex;
                height: 100%;
                width: 100%;
                font-family: -apple-system, BlinkMacSystemFont;
            }
            h1 {
                text-align: center;
            }
            .form {
                margin: 20px auto;
            }
            input {
                display: block;
                margin-bottom: 20px;
                width: 300px;
                height: 30px;
                font-size: 14px;
            }
        </style>
    </head>
    <body>
        <div class="form">
            <h1>Login Page</h1>
            <form>
                <input type="text" name="email" id="email" placeholder="E-Mail">
                <input type="password" name="pwd" id="pwd" placeholder="Password">
                <input type="button" value="Login" onclick="validation()">
            </form>
            <span id="output"></span>
        </div>

        <script>
            function validation() {
                var out = document.getElementById('output')
```

```
                    var email = document.getElementById('email').value
                    var pwd = document.getElementById('pwd').value

                    var emailValid = email.match(/^[\w\.-]+@[\w\.]+[A-Za-z]{2,3}$/)
                    var pwdValid = pwd.match(/^.{6,}$/)

                    if(!emailValid) {
                        output.innerHTML = 'E-Mail is not valid'
                    } else if(!pwdValid) {
                        output.innerHTML = 'Password should be atlest 6 characters long'
                    } else {
                        output.innerHTML = 'Welcome, ' + email
                    }
                }
            </script>
        </body>
    </html>
```

○ Payment Page

```
    <!DOCTYPE html>
    <html lang="en">
        <head>
            <title>Payment Page</title>
            <style>
                html, body {
                    display: flex;
                    height: 100%;
                    width: 100%;
                    font-family: -apple-system, BlinkMacSystemFont;
                }
                h1 {
                    text-align: center;
                }
```

```
            .form {
                margin: 20px auto;
            }
            input {
                display: block;
                margin-bottom: 20px;
                width: 300px;
                height: 30px;
                font-size: 14px;
            }
        </style>
    </head>
    <body>
        <div class="form">
            <h1>Payment Page</h1>
            <form>
                <input type="text" name="name" id="name" placeholder="Name">
                <input type="text" name="card" id="card" placeholder="Credit Card Number">
                <input type="text" name="date" id="date" placeholder="Expiry Date (DD / MM)">
                <input type="password" name="cvv" id="cvv" placeholder="CVV">
                <input type="text" name="phone" id="phone" placeholder="Phone Number">
                <input type="button" value="Pay" onclick="validation()">
            </form>
            <span id="output"></span>
        </div>

        <script>
            function validation () {
                var out = document.getElementById('output')
                var name = document.getElementById('name').value
                var card = document.getElementById('card').value
                var date = document.getElementById('date').value
                var cvv = document.getElementById('cvv').value
                var ph = document.getElementById('phone').value

                var nameValid = name.match(/^[A-Za-z ]+$/)
```

```
                var cardValid = card.match(/^\d{16}$/)
                var dateValid = date.match(/^\d{2}\/\d{2}$/)
                var cvvValid = cvv.match(/^.{3,4}$/)
                var phValid = ph.match(/^\d{10}$/)

                if(!nameValid) {
                    out.innerHTML = 'Invalid Name'
                } else if(!cardValid) {
                    out.innerHTML = 'Invalid Card Number'
                } else if(!dateValid) {
                    out.innerHTML = 'Invalid Date Format'
                } else if(!cvvValid) {
                    out.innerHTML = 'Invalid CVV'
                } else if(!phValid) {
                    out.innerHTML = 'Invalid Phone number'
                } else {
                    out.innerHTML = 'Payment Successful'
                }
            }
        </script>
    </body>
</html>
```

- Question 4: Demonstrate following HTML DOM functions

  - getElementById()

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>getElementById() function</title>
    </head>
    <body>
        <p id="demo">Click the button to change the color of this paragraph.</p>
```

```
        <button id="btn">Click ME!</button>

        <script>
            var button = document.getElementById('btn')

            button.onclick = function() {
                var p = document.getElementById('demo')
                p.style.color = 'red'
            }
        </script>
    </body>
</html>
```

- getElementByClassName()

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>getElementByClassName() function</title>
    </head>
    <body>
        <div class="example">
            A div with class="example"
        </div>

        <div class="example">
            Another div with class="example"
        </div>

        <p class="example">This is a p element with class="example".</p>
        <button id="btn">Click ME!</button>

        <script>
            var button = document.getElementById('btn')
```

```
                button.onclick = function() {
                    var x = document.getElementsByClassName('example')
                    for (var i = 0; i < x.length; i++) {
                        x[i].style.color = "red";
                    }
                }
            </script>
        </body>
    </html>
```

- getElementByTagName()

```
    <!DOCTYPE html>
    <html lang="en">
        <head>
            <title>getElementByTagName() function</title>
        </head>
        <body>
            <p>This is a p element</p>
            <p>This is also a p element.</p>
            <p>This is also a p element - Click the button to change the color of all p elements in this
document.</p>
            <button id="btn">Click ME!</button>

            <script>
                var button = document.getElementById('btn')

                button.onclick = function() {
                    var x = document.getElementsByTagName('p')
                    for (var i = 0; i < x.length; i++) {
                        x[i].style.color = "red"
                    }
                }
```

```
            </script>
        </body>
    </html>
```

- Question 5(a): Write an XML Page for books catalog with Title of Book, Author, ISBN, Publisher, Edition, Price

```xml
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
    <book id="1" category="science">
        <title>A Brief History of Time</title>
        <author>Stephen Hawking</author>
        <isbn>978-05-53109-53-5</isbn>
        <publisher>Bantam Dell Publishing Group</publisher>
        <edition>1</edition>
        <price>239.00</price>
    </book>
    <book id="2" category="Biography">
        <title>Elon Musk: Tesla, SpaceX, and the Quest for a Fantastic Future</title>
        <author>Ashlee Vance</author>
        <isbn>978-00-62301-23-9</isbn>
        <publisher>Virgin Books</publisher>
        <edition>1</edition>
        <price>1796.00</price>
    </book>
    <book id="3" category="Biography">
        <title>Steve Jobs</title>
        <author>Walter Isaacson</author>
        <isbn>978-03-49140-43-8</isbn>
        <publisher>Simon &amp; Schuster</publisher>
        <edition>1</edition>
        <price>440.00</price>
    </book>
```

```
        <book id="4" category="Memoir">
            <title>Becoming</title>
            <author>Michelle Obama</author>
            <isbn>978-15-24763-13-8</isbn>
            <publisher>Crown</publisher>
            <edition>1</edition>
            <price>230</price>
        </book>
        <book id="5" category="Textbook">
            <title>Beginning HTML, XHTML, CSS and Javascript</title>
            <author>Jon Duckett</author>
            <isbn>978-81-26525-51-5</isbn>
            <publisher>Wiley</publisher>
            <edition>8</edition>
            <price>424.00</price>
        </book>
    </bookstore>
```

- Question 5(b): Write both Internal and External DTD with Schema for above XML Page

  - Internal DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE bookstore [
    <!ELEMENT bookstore (book+)>
    <!ELEMENT book (title, author, isbn, publisher, edition, price)>
    <!ELEMENT title (#PCDATA)>
    <!ELEMENT author (#PCDATA)>
    <!ELEMENT isbn (#PCDATA)>
    <!ELEMENT publisher (#PCDATA)>
    <!ELEMENT edition (#PCDATA)>
    <!ELEMENT price (#PCDATA)>
    <!ATTLIST book id ID #REQUIRED>
    <!ATTLIST book category CDATA #IMPLIED>
```

```
]>
<bookstore>
    <book id="1" category="science">
        <title>A Brief History of Time</title>
        <author>Stephen Hawking</author>
        <isbn>978-05-53109-53-5</isbn>
        <publisher>Bantam Dell Publishing Group</publisher>
        <edition>1</edition>
        <price>239.00</price>
    </book>
    <book id="2" category="Biography">
        <title>Elon Musk: Tesla, SpaceX, and the Quest for a Fantastic Future</title>
        <author>Ashlee Vance</author>
        <isbn>978-00-62301-23-9</isbn>
        <publisher>Virgin Books</publisher>
        <edition>1</edition>
        <price>1796.00</price>
    </book>
</bookstore>
```

- External DTD

external.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE bookstore SYSTEM "external.dtd">
<bookstore>
    <book id="1" category="science">
        <title>A Brief History of Time</title>
        <author>Stephen Hawking</author>
        <isbn>978-05-53109-53-5</isbn>
        <publisher>Bantam Dell Publishing Group</publisher>
        <edition>1</edition>
        <price>239.00</price>
```

```
        </book>
        <book id="2" category="Biography">
            <title>Elon Musk: Tesla, SpaceX, and the Quest for a Fantastic Future</title>
            <author>Ashlee Vance</author>
            <isbn>978-00-62301-23-9</isbn>
            <publisher>Virgin Books</publisher>
            <edition>1</edition>
            <price>1796.00</price>
        </book>
    </bookstore>
```

external.dtd

```
    <!ELEMENT bookstore (book+)>
    <!ELEMENT book (title, author, isbn, publisher, edition, price)>
    <!ELEMENT title (#PCDATA)>
    <!ELEMENT author (#PCDATA)>
    <!ELEMENT isbn (#PCDATA)>
    <!ELEMENT publisher (#PCDATA)>
    <!ELEMENT edition (#PCDATA)>
    <!ELEMENT price (#PCDATA)>
    <!ATTLIST book id ID #REQUIRED>
    <!ATTLIST book category CDATA #IMPLIED>
```

- XML Schema

  schema.xml

```
    <?xml version="1.0" encoding="UTF-8"?>
    <bookstore
        xmlns="https://www.w3schools.com"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```xml
        xsi:schemaLocation="https://www.w3schools.com/xml schema.xsd">
    <book id="1" category="science">
        <title>A Brief History of Time</title>
        <author>Stephen Hawking</author>
        <isbn>978-05-53109-53-5</isbn>
        <publisher>Bantam Dell Publishing Group</publisher>
        <edition>1</edition>
        <price>239.00</price>
    </book>
    <book id="2" category="Biography">
        <title>Elon Musk: Tesla, SpaceX, and the Quest for a Fantastic Future</title>
        <author>Ashlee Vance</author>
        <isbn>978-00-62301-23-9</isbn>
        <publisher>Virgin Books</publisher>
        <edition>1</edition>
        <price>1796.00</price>
    </book>
</bookstore>
```

schema.xsd

```xml
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="https://www.w3schools.com">

    <xs:element name="bookstore">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="book">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="title" type="xs:string"/>
                            <xs:element name="author" type="xs:string"/>
```

```
                                <xs:element name="isbn" type="xs:string"/>
                                <xs:element name="publisher" type="xs:string"/>
                                <xs:element name="edition" type="xs:string"/>
                                <xs:element name="price" type="xs:string"/>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:schema>
```

- Question 6: Write XSLT page for XML file in Question 5

books.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="books.xsl"?>
<bookstore>
    <book id="1" category="science">
        <title>A Brief History of Time</title>
        <author>Stephen Hawking</author>
        <isbn>978-05-53109-53-5</isbn>
        <publisher>Bantam Dell Publishing Group</publisher>
        <edition>1</edition>
        <price>239.00</price>
    </book>
    <book id="2" category="Biography">
        <title>Elon Musk: Tesla, SpaceX, and the Quest for a Fantastic Future</title>
        <author>Ashlee Vance</author>
        <isbn>978-00-62301-23-9</isbn>
        <publisher>Virgin Books</publisher>
        <edition>1</edition>
        <price>1796.00</price>
```

```xml
        </book>
        <book id="3" category="Biography">
            <title>Steve Jobs</title>
            <author>Walter Isaacson</author>
            <isbn>978-03-49140-43-8</isbn>
            <publisher>Simon &amp; Schuster</publisher>
            <edition>1</edition>
            <price>440.00</price>
        </book>
        <book id="4" category="Memoir">
            <title>Becoming</title>
            <author>Michelle Obama</author>
            <isbn>978-15-24763-13-8</isbn>
            <publisher>Crown</publisher>
            <edition>1</edition>
            <price>230.00</price>
        </book>
        <book id="5" category="Textbook">
            <title>Beginning HTML, XHTML, CSS and Javascript</title>
            <author>Jon Duckett</author>
            <isbn>978-81-26525-51-5</isbn>
            <publisher>Wiley</publisher>
            <edition>8</edition>
            <price>424.00</price>
        </book>
    </bookstore>
```

books.xsl

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

    <xsl:template match="/">
```

```html
<html>
    <head>
        <title>Books Catalog</title>
        <style>
            table, th, td {
                border: 1px solid black;
                border-collapse: collapse;
            }

            th, td {
                padding: 5px 15px;
            }
        </style>
    </head>
    <body>
        <h1>Books Catalog</h1><br />
        <table>
            <tr>
                <th>Title</th>
                <th>Author</th>
                <th>ISBN</th>
                <th>Publisher</th>
                <th>Edition</th>
                <th>Price</th>
            </tr>
            <xsl:for-each select="bookstore/book">
            <tr>
                <td><xsl:value-of select="title"/></td>
                <td><xsl:value-of select="author"/></td>
                <td><xsl:value-of select="isbn"/></td>
                <td><xsl:value-of select="publisher"/></td>
                <td align="right"><xsl:value-of select="edition"/></td>
                <td align="right">₹ <xsl:value-of select="price"/></td>
            </tr>
            </xsl:for-each>
        </table>
```

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

    </xsl:stylesheet>
```

- Question 7: Write a Program to establish a Connection with the Database and create a Statement to insert values into a table

```java
import java.sql.*;

public class DBInsert {
    public static void main(String args[]) {
        Connection con = null;
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","manager");
            System.out.println("\nConnection Successful.\n");

            Statement st = con.createStatement();
            st.execute("insert into Account values(1,'Ramu',50000)");
            st.execute("insert into Account values(2,'Ramesh',70000.25)");
            st.execute("insert into Account values(3,'Sharath',80000)");
            System.out.println("Insert Operation Successful.");

            st.close();
            con.close();
        } catch(Exception sqle) {
            System.out.println("Exception: "+sqle);
        }
    }
}
```

- Question 8: Write a Program to use PreparedStatement interface to insert values into a table

```
import java.sql.*;

public class PreparedStatementDemo {
    public static void main(String args[]) throws Exception {
        Connection con = null;
        PreparedStatement ps = null;

        Class.forName("oracle.jdbc.driver.OracleDriver");
        con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","manager");
        System.out.println("\nConnection Successful.\n");

        ps = con.prepareStatement("insert into account values(?,?,?)");
        ps.setInt(1,4);
        ps.setString(2,"Prajeet");
        ps.setInt(3,50000);
        ps.execute();
        System.out.println("Insert Operation Successful.");

        ps.close();
        con.close();
    }
}
```

- Question 9: Write a Program to use CallableStatement interface to call a procedure declared in the database

```
import java.sql.*;

public class CallableStatementDemo {
    public static void main(String args[]) throws Exception {
```

```java
            Connection con = null;
            CallableStatement cst = null;

            Class.forName("oracle.jdbc.driver.OracleDriver");
            con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","manager");
            System.out.println("\nConnection Successful.\n");

            cst = con.prepareCall("{call square(?,?)}");
            cst.setInt(1,3);
            cst.registerOutParameter(2,Types.INTEGER);
            cst.execute();
            System.out.println("\nProcedure Execution Successful.\n");
            System.out.println("\nSquare of given number is: " + cst.getInt(2) + "\n");

            cst.close();
            con.close();
        }
    }
```

- Question 10: Write a Program to use ResultSet interface to print the data stored in the Database

```java
    import java.sql.*;

    public class ResultSetDemo {
        public static void main(String args[]) throws Exception {
            Connection con = null;
            Statement st = null;
            ResultSet rs = null;

            Class.forName("oracle.jdbc.driver.OracleDriver");
            con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","manager");
            st = con.createStatement();
            System.out.println("\nConnection Successful.\n");
```

```
            rs=st.executeQuery("select * from Account");

            while(rs.next())
                System.out.println("|"+rs.getInt(1)+"|"+rs.getString(2)+"|"+rs.getInt(3)+"|");
            System.out.println("\nEnd of Information Recieved.\n");

            rs.close();
            st.close();
            con.close();
        }
    }
```

- Question 11: Write a Program to use ResultSetMetaData interface to get the metadata about the ResultSet object

```
import java.sql.*;

public class ResultSetMeta {
    public static void main(String args[]) throws Exception {
        Connection con = null;
        Statement st = null;
        ResultSet rs = null;
        ResultSetMetaData rsmd = null;

        Class.forName("oracle.jdbc.driver.OracleDriver");
        con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","manager");
        System.out.println("\nConnection Successful.\n");

        st=con.createStatement();
        rs=st.executeQuery("select * from account");
        rsmd=rs.getMetaData();

        for(int i=0;i<rsmd.getColumnCount();i++)
```

```
            System.out.print("|"+rsmd.getColumnName(i+1)+"|");

        System.out.println();
        while(rs.next())
            System.out.println("|"+rs.getInt(1)+"|"+rs.getString(2)+"|"+rs.getInt(3)+"|");

        rs.close();
        st.close();
        con.close();


    }
}
```

- Question 12: Write a Servlet Program with Webpage to ask User's Name and show a Hello Message.

  index.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>Hello</title>
    </head>
    <body>
        <h1>Welcome.</h1>

        <form method="POST" action="hello">
            Name: <input type="text" name="name" > <br>
            <input type="submit" value="Display">
        </form>
    </body>
</html>
```

WEB-INF/classes/HelloMessage.java

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

public class HelloMessage extends HttpServlet {

    public void doPost(HttpServletRequest req, HttpServletResponse res) throws IOException,
ServletException {
        res.setContentType("text/html");

        String name = req.getParameter("name");

        PrintWriter out = res.getWriter();
        out.println("<html><head><title>Hello</title></head><body><h1>Welcome, " + name + "</h1></body>
</html>");
    }
}
```

WEB-INF/web.xml

```xml
<web-app>
    <servlet>
        <servlet-name>Hello</servlet-name>
        <servlet-class>HelloMessage</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>Hello</servlet-name>
        <url-pattern>/hello</url-pattern>
    </servlet-mapping>
</web-app>
```

- Question 13: Write a Servlet Program to insert data into a database table

index.html

```html
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Database Insert</title>
    </head>
    <body>
        <form action="register" method="post">
            <input type="text" name="name" placeholder="Name" required> <br>
            <input type="text" name="email" placeholder="E-Mail" required> <br>
            <input type="password" name="pwd" placeholder="Password" required> <br>
            <input type="password" name="cpwd" placeholder="Confirm Password" required> <br>
            <input type="submit" value="Register"> <br>
        </form>
    </body>
</html>
```

WEB-INF/classes/RegisterUser.java

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

public class RegisterUser extends HttpServlet {

    public void doPost(HttpServletRequest req, HttpServletResponse res) throws IOException,
ServletException {
            res.setContentType("text/html");
            PrintWriter out = res.getWriter();
```

```java
            out.println("<html><head><title>Database Insert</title></head><body><h1>");

            String name = req.getParameter("name");
            String email = req.getParameter("email");
            String pwd = req.getParameter("pwd");
            String cpwd = req.getParameter("cpwd");

            if(!pwd.equals(cpwd)) {
                out.println("Passwords don't match");
                return;
            }

            try {
                Class.forName("oracle.jdbc.driver.OracleDriver");
                Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","manager");
                PreparedStatement ps = con.prepareStatement("INSERT INTO users(name, email, pwd) VALUES (?,
?, ?)");
                ps.setString(1, name);
                ps.setString(2, email);
                ps.setString(3, pwd);

                if(ps.executeUpdate() > 0)
                    out.println("Registration Successful");
                else
                    out.println("An error occured while registration");

                ps.close();
                con.close();
            } catch(Exception ex) {
                out.println("An error occured while registration.<br>" + ex.getMessage());
            }

            out.println("</h1></body></html>");
        }
    }
```

```
    /**
    * Before running, create table in Oracle DB using,
    * create table users(id NUMBER GENERATED ALWAYS AS IDENTITY, name VARCHAR2(255), email varchar2(255), pwd
varchar2(255));
    */
```

WEB-INF/web.xml

```
<web-app>
    <servlet>
        <servlet-name>Register</servlet-name>
        <servlet-class>RegisterUser</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>Register</servlet-name>
        <url-pattern>/register</url-pattern>
    </servlet-mapping>
</web-app>
```