

When installing tomcat, if ports are not visible (in Windows), go to “C->Program files->Apache software foundation -> Tomcat -> conf”. Copy all files here. Then go to eclipse workspace -> servers and paste those files here and check in eclipse under servers, tomcat, if they are pasted.

Experiment 1

Develop static pages (using HTML and CSS) of an online book store. The pages should resemble: www.flipkart.com The website should consist of the following pages.

- a. Home Page b. Registration and User Login c. User Profile Page d. Books Catalog**
- e. Shopping Cart f. Payment by Credit Card g. Order Confirmation**

Main.html:

```
<html>
<head>
  <title>Amazon</title>
</head>
<body bgcolor="pink">
  <center>
    <strong><h1>Welcome to AMAZON</h1></strong>
    <form method="post" action="login.html" target="_blank">
      <h4>for books</h4>
      <input type="submit" value="click here">
    </form>
  </center>
</body>
</html>
```

login.html:

```
<html>
<head>
  <title>Login</title>
</head>
<body bgcolor="pink">
```

```

<center>
  <strong><h1>AMAZON</h1></strong>
</center>
<right>
  <table align="right">
    <tr>
      <td><h4>user name</td>
      <td><input type="text" ></td>
      <td></td>
    </tr>
    <tr>
      <td><h4>password</h4></td>
      <td><input type="password"></td>
    </tr>
    <tr>
      <td>
        <form method="post" action="catalog.html">
          <input type="submit" value="submit">
        </form>
      </td>
      <td>
        <form method="post" action="reg.html">
          <input type="submit" value="register">
          <input type="reset" value="reset">
        </form>
      </td>
    </tr>
  </table>
</right>
</body>
</html>

```

reg.html:

```

<html>
<head>
  <title>Login Page</title>
</head>
<body bgcolor="pink">
  <center><strong><h1>AMAZON</h1></strong></center>
  <form method="post" action="catalog.html">
    <table align="left">
      <tr>

```



```

<table>
  <tr>
    <td><b><h3>Frontend books</h3></b></td>
    <td></td>
  </tr>
  <tr>
    <td></td>
    <td><h4>C & Ds</h4></td>
  </tr>
  <tr>
    <td></td>
    <td><h4>Ads</h4></td>
  </tr>
  <tr>
    <td></td>
    <td><h4>Java</h4></td>
  </tr>
  <tr>
    <td><b><h3></h3>Backend books</b></td>
    <td></td>
  </tr>
  <tr>
    <td></td>
    <td><h4>Oracle</h4></td>
  </tr>
  <tr>
    <td></td>
    <td><h4>My Sql</h4></td>
  </tr>
</table>
<b>To buy any one of these books<br></b>
<input type="submit" value="click here">
</left>
</form>
</body>
</html>

```

shopping.html:

```

<html>
<head>
  <title>Shopping Cart</title>
</head>
<body bgcolor="pink">

```

```

<br><br><br><br><br>
    <table align="center">
        <tr>
            <td>Text Books</td>
            <td>
                <select>
                    <optgroup label="select the book">
                        <option value="C & Ds">
                            C & Ds
                        </option>
                        <option value="Java">
                            Java
                        </option>
                        <option value="MySql">
                            MySql
                        </option>
                    </optgroup>
                </select>
            </td>
        </tr>
        <tr>
            <td>Quantity</td>
            <td>
                <input type="text" id="q">
            </td>
        </tr>
        <tr>
            <td></td>
            <td>
                <form method="post" action="payment.html">
                    <input type="submit" value="ok">
                </form>
            </td>
        </tr>
    </table>
    <center>
        <pre>Cost of one book is "500" + shipping "100"</pre>
    </center>
</body>
</html>

```

payment.html:

```

<html>
<head>
  <title>Payment</title>
</head>
<body bgcolor="pink">
  <center><h1>Payment by credit card</h1></center>
  <form method="post" action="orderconform.html">
    <br><br><br><br><br>
    <table align="center">
      <tr>
        <td><h4>Total Amount</h4></td>
        <td><input type="text"></td>
      </tr>
      <tr>
        <td><h4>Credit card number</h4></td>
        <td><input type="text"></td>
      </tr>
      <tr>
        <td></td>
        <td><input type="submit" value="ok">
      </td>
    </tr>
  </table>
</form>
</body>
</html>

```

orderconform.html:

```

<html>
<head>
  <title>Order Confirmation</title>
</head>
<body bgcolor="pink">
  <center>
    <h1>BOOK SHOPPING</h1>
    <pre><strong>Your order Is Conformed</strong></pre>
    <h2>THANK YOU</h2>
  </center>
</body>
</html>

```

Experiment 2

Create and save an XML document on the server, which contains 10 user's information. Write a program, which takes User Id as an input and returns the user details by taking the user information from the XML document

// users.xml must be within webapp in eclipse

users.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<users>
  <user>
    <id>1</id>
    <name>Gladys</name>
    <email>Gladys@gmail.com</email>
    <location>USA</location>
  </user>
  <user>
    <id>2</id>
    <name>Daniel</name>
    <email>Dany@gmail.com</email>
    <location>USA</location>
  </user>
  <user>
    <id>3</id>
    <name>Emily</name>
    <email>Emily@example.com</email>
    <location>Canada</location>
  </user>
  <user>
    <id>4</id>
    <name>John</name>
    <email>John@example.com</email>
    <location>UK</location>
  </user>
  <user>
    <id>5</id>
    <name>Sophia</name>
    <email>Sophia@example.com</email>
```

```
        <location>Germany</location>
    </user>
    <user>
        <id>6</id>
        <name>Michael</name>
        <email>Michael@example.com</email>
        <location>Australia</location>
    </user>
    <user>
        <id>7</id>
        <name>Olivia</name>
        <email>Olivia@example.com</email>
        <location>France</location>
    </user>
    <user>
        <id>8</id>
        <name>William</name>
        <email>William@example.com</email>
        <location>Brazil</location>
    </user>
    <user>
        <id>9</id>
        <name>Ava</name>
        <email>Ava@example.com</email>
        <location>Spain</location>
    </user>
    <user>
        <id>10</id>
        <name>James</name>
        <email>James@example.com</email>
        <location>Italy</location>
    </user>
</users>
```

index.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>Retrieve User Details</title>
</head>
<body>
    <label for="userId">Enter User ID: </label>
```



```

<input type="text" id="userId">
<button onclick="getUserDetails()">Get Details</button>
<div id="output"></div>
<script>
function getUserDetails() {
    // Get user ID input
    var inputUserId = document.getElementById("userId").value;
    // Fetch XML data (replace 'data.xml' with your XML file path)
    fetch('users.xml')
        .then(response => response.text())
        .then(data => {
            // Parse XML data
            var parser = new DOMParser();
            var xmlDoc = parser.parseFromString(data, "text/xml");
            // Find user nodes based on ID
            var users = xmlDoc.getElementsByTagName('user');
            var userDetails = [];
            // Loop through users to find matching user ID
            for (var i = 0; i < users.length; i++) {
                var userId = users[i].getElementsByTagName('id')[0].childNodes[0].nodeValue;

                if (userId === inputUserId) {
                    // Extract user details including ID
                    var userName = users[i].getElementsByTagName('name')[0].childNodes[0].nodeValue;
                    var userEmail = users[i].getElementsByTagName('email')[0].childNodes[0].nodeValue;
                    var userLocation =
users[i].getElementsByTagName('location')[0].childNodes[0].nodeValue;
                    userDetails.push({
                        'ID': userId,
                        'Name': userName,
                        'Email': userEmail,
                        'Location': userLocation
                    });
                }
            }
            // Display user details
            var outputDiv = document.getElementById("output");
            outputDiv.innerHTML = "<h3>User Details:</h3>";
            if (userDetails.length > 0) {
                userDetails.forEach(function(user) {
                    outputDiv.innerHTML += "<p><strong>ID:</strong> " + user.ID + "</p>";
                    outputDiv.innerHTML += "<p><strong>Name:</strong> " + user.Name + "</p>";
                    outputDiv.innerHTML += "<p><strong>Email:</strong> " + user.Email + "</p>";
                    outputDiv.innerHTML += "<p><strong>Location:</strong> " + user.Location + "</p>";
                });
            }
        });
}
}

```

```

    });
  } else {
    outputDiv.innerHTML += "<p>No user found with ID: " + inputUserId + "</p>";
  }
})
).catch(error => console.error('Error:', error));
}
</script>
</body>
</html>

```

Experiment 3

Write a PHP script to merge two arrays and sort them as numbers, in descending order

```

<?php
$a1 = array(22, 15, 37, 62, 34);
$a2 = array(23, 17, 36, 67, 38);
$n = array_merge($a1, $a2);
rsort($n);
print_r($n);
?>

```

Experiment 4

Write a PHP script that reads data from one file and write into another file.

```

<?php
$file = "data1.txt"; // Corrected file names with proper extensions
$copy = "data2.txt"; // Corrected file names with proper extensions

if (copy($file, $copy)) {
    echo "Copied $file to $copy";
} else {
    echo "Failed to copy $file to $copy";
}
?>

```

Experiment 5

Write a PHP script to print prime numbers between 1 -50

```
<?php
$num = 1;
while ($num <= 50) {
    $count = 0;
    for($i = 1; $i <= $num; $i++) {
        if($num % $i == 0) {
            $count++;
        }
    }
    if($count < 3) {
        echo $num." ";
    }
    $num = $num + 1;
}
?>
```

Experiment 6

Validate the Registration, user login, user profile and payment by credit card pages using JavaScript.

// Login id and password must only be letters and the first letters must be capital

main.html:

```
<html>
<frameset rows="25%,*">
<frame src="top.html" name="top" scrolling="no" frameborder="0">
<frameset cols="25%,75%">
<frame src="left.html" name="left" scrolling="no" frameborder="0">
<frame src="right.html" name="right" scrolling="auto" frameborder="0">
</frameset>
</frameset>
```

</html>

top.html:

```
<html>
<body bgcolor="pink">
<br><br>
<marquee><h1
align="center"><b><u>ONLINE BOOK
STORAGE</u></b></h1></marquee>
</body>
</html>
```

right.html:

```
<html>
<body>
<br><br><br><br><br>
<h2 align="center">
<b><p> welcome to online book storage. Press login if you are having id otherwise press
registration.
</p></b></h2>
</body></html>
```

left.html:

```
<html>
<body bgcolor="pink">
<h3>
<ul>
<li><a href="login.html" target="right">
    <font color="black">LOGIN</font></a></li><br><br>
<li><a href="reg.html" target="right">
    <font color="black"> REGISTRATION </font></a></li><br><br>
<li><a href="profile.html" target="right">
    <font color="black"> USER PROFILE</font></a></li><br><br>
```

```

<li><a href="catalog.html" target="right">
    <font color="black"> BOOKS CATALOG</font></a></li><br><br>
<li><a href="scart.html" target="right">
    <font color="black"> SHOPPING CART</font></a></li><br><br>
<li><a href="payment.html" target="right">
    <font color="black">PAYMENT</font></a></li><br><br>
<br><br>
</ul>
</body>
</html>

```

login.html:

```

<!DOCTYPE html>
<html>
<head>
    <title>Login Page</title>
</head>
<body bgcolor="pink">
    <br><br><br>
    <script language="javascript">
        function validate() {
            var flag = 1;
            var id = document.myform.id.value;
            var pwd = document.myform.pwd.value;

            if (id === "" || pwd === "") {
                alert("Login ID and Password must be filled.");
                flag = 0;
            }

            if (flag === 1) {
                alert("VALID INPUT");
                window.location.href = "catalog.html"; // Change to the desired page.
            } else {
                alert("INVALID INPUT");
            }
        }
    </script>
    <form name="myform">
        <div align="center">
            <pre>

```

```

LOGIN ID: <input type="text" name="id"><br>
PASSWORD: <input type="password" name="pwd"><br><br>
</pre>
<input type="button" value="OK" onClick="validate()">
<input type="reset" value="Clear">
</div>
</form>
</body>
</html>

```

reg.html:

```

<!DOCTYPE html>
<html>
<head>
  <title>User Registration</title>
</head>
<body bgcolor="pink">
  <br><br>
  <script type="text/javascript">
    function validate() {
      var flag = 1;
      var name = document.myform.name.value;
      var addr = document.myform.addr.value;
      var phno = document.myform.phno.value;
      var id = document.myform.id.value;
      var pwd = document.myform.pwd.value;

      if (name === "" || addr === "" || phno === "" || id === "" || pwd === "") {
        alert("Please fill in all the details.");
        flag = 0;
      }

      var str = document.myform.phno.value;
      var x = new RegExp("\\d", "g");
      if (!(str.match(x)) || str.length !== 10) {
        flag = 0;
      }

      var str1 = document.myform.id.value;
      var x1 = new RegExp("[A-Z][a-zA-Z]+$", "g");
      if (!str1.match(x1)) {
        flag = 0;
      }
    }
  </script>

```

```

        alert("Invalid UserID");
    }

    var str2 = document.myform.pwd.value;
    var x2 = new RegExp("^[A-Z][a-zA-Z]+$", "g");
    if (!str2.match(x2)) {
        flag = 0;
        alert("Invalid password");
    }

    if (flag === 1) {
        alert("VALID INPUT");
        window.location.href = "login.html";
    } else {
        alert("INVALID INPUT");
        document.myform.focus();
    }
}
</script>

<form name="myform">
    <div align="center">
        <pre>
            NAME :<input type="text" name="name"><br>
            ADDRESS :<input type="text" name="addr"><br>
            CONTACT NUMBER :<input type="text" name="phno"><br>
            LOGIN ID :<input type="text" name="id"><br>
            PASSWORD :<input type="password" name="pwd">
        </pre>
        <br><br>
    </div>
    <div align="center">
        <input type="button" value="Register" onClick="validate()">
        <input type="reset" value="Clear">
    </div>
</form>
</body>
</html>

```

profile.html:

```

<!DOCTYPE html>
<html>
<head>

```

```

<title>User Profile</title>
</head>
<body bgcolor="pink">
  <br><br>
  <script type="text/javascript">
    function validate() {
      var name = document.myform.name.value;
      var addr = document.myform.addr.value;
      var phno = document.myform.phno.value;
      var id = document.myform.id.value;
      var pwd = document.myform.pwd.value;

      if (name === "" || addr === "" || phno === "" || id === "" || pwd === "") {
        alert("Please fill in all the details.");
      } else if (!/^d{10}$/.test(phno)) {
        alert("Invalid contact number. It must be 10 digits.");
      } else {
        alert("Profile updated successfully.");
      }
    }
  </script>

  <form name="myform">
    <div align="center">
      <pre>
        NAME: <input type="text" name="name"><br>
        ADDRESS: <input type="text" name="addr"><br>
        CONTACT NUMBER: <input type="text" name="phno"><br>
        LOGIN ID: <input type="text" name="id"><br>
        PASSWORD: <input type="password" name="pwd">
      </pre>
      <br><br>
    </div>
    <div align="center">
      <input type="button" value="Update Profile" onClick="validate()">
      <input type="reset" value="Clear">
    </div>
  </form>
</body>
</html>

```

scart.html:


```
<!DOCTYPE html>
<html>
<head>
  <title>Form Validation</title>
</head>
<body bgcolor="pink">
  <br><br><br>
  <form name="myform" onsubmit="return validateForm();">
    <label for="title">Title: </label>
    <input type="text" id="title" name="title" required>
    <br><br>
    <input type="submit" value="Submit">
  </form>

  <script>
    function validateForm() {
      var flag = 1;
      var title = document.myform.title.value;

      if (title === "") {
        flag = 0;
        alert("Please enter a value in the 'Title' field.");
      } else if (title.toLowerCase() === "c") {
        alert("Valid input: 'Title' is 'C'");
        window.location.href = "payment.html"; // Change to the desired page.

      } else if (title.toLowerCase() === "jsp") {
        alert("Valid input: 'Title' is 'JSP'");
        window.location.href = "payment.html"; // Change to the desired page.

      } else {
        flag = 0;
        alert("Invalid input. 'Title' must be 'C' or 'JSP'.");
      }

      return flag === 1;
    }
  </script>
</body>
</html>
```

catalog.html:

```
<!DOCTYPE html>
<html>
<head>
  <title>Book Information</title>
</head>
<body bgcolor="pink">
  <br><br><br>
  <script language="javascript">
    function validate() {
      var flag = 1;
      var id = document.myform.id.value;
      var title = document.myform.title.value;
      var no = document.myform.no.value;
      var cost = document.myform.cost.value;

      if (id === "" || title === "" || no === "" || cost === "") {
        flag = 0;
      }

      var str = title.toLowerCase();
      var str1 = parseInt(cost);

      if (!(str === "c" && str1 === 444) || (str === "jsp" && str1 === 555)) {
        flag = 0;
      }

      if (flag === 1) {
        alert("VALID INPUT");
      } else {
        alert("INVALID INPUT");
        document.myform.focus();
      }
    }
  </script>

  <form name="myform" action="payment.html" target="right">
    <div align="center">
      <pre>
        LOGIN ID: <input type="text" name="id"><br>
        TITLE: <input type="text" name="title"><br>
        NO. OF BOOKS: <input type="text" name="no"><br>
        COST OF BOOK: <input type="text" name="cost"><br>
```

```

        </pre>
        <br><br>
    </div>

    <div align="center">
        <input type="submit" value="OK" onClick="validate()">
        <input type="reset" value="Clear">
    </div>
</form>
</body>
</html>

```

payment.html:

```

<!DOCTYPE html>
<html>
<head>
    <title>Login and Payment</title>
</head>
<body bgcolor="pink">
    <br><br><br>
    <script language="javascript">
        function validate() {
            var flag = 1;
            var id = document.myform.id.value;
            var pwd = document.myform.pwd.value;
            var amount = document.myform.amount.value;
            var num = document.myform.num.value;

            if (id === "" || pwd === "" || amount === "" || num === "") {
                flag = 0;
                alert("Please fill in all the fields.");
            }

            var str = amount;
            var x = new RegExp("^\\d+$");

            if (!str.match(x)) {
                flag = 0;
                alert("Amount must be a positive number.");
            }
        }
    </script>

```

```

        if (flag === 1) {
            alert("VALID INPUT");
            window.location.href = "order.html"; // Change to the desired page.
        } else {
            alert("INVALID INPUT");
            document.myform.focus();
        }
    }
}
</script>

<form name="myform">
    <div align="center">
        <pre>
            LOGIN ID: <input type="text" name="id"><br>
            PASSWORD: <input type="password" name="pwd"><br>
            AMOUNT: <input type="text" name="amount"><br>
            CREDIT CARD NUMBER: <input type="password" name="num">
        </pre>
        <br><br>
    </div>
    <div align="center">
        <input type="button" value="OK" onClick="validate()">
        <input type="reset" value="Clear">
    </div>
</form>
</body>
</html>

```

order.html:

```

<!DOCTYPE html>
<html>
<head>
    <title>Order Confirmation</title>
</head>
<body bgcolor="pink">
    <center>
        <h1><b>AMAZON</b></h1>
        <pre><strong>
            <b>Your Order Is Confirmed
        </strong></pre>
        <h2><b>THANK YOU</b></h2>
    </center>

```

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

Experiment 7

i) **Find the length of a string.**

```
<!Doctype html>  
<html>  
<body>  
<?php  
echo strlen("Hello World!");  
>  
</body>  
</html>  
</html>
```

ii) **Count no of words in a string.**

```
<!Doctype html>  
<html>  
<body>  
<?php  
echo str_word_count("Hello world!");  
>  
</body>  
</html>
```

iii) **Reverse a string.**

```
<!doctype html>  
<html>  
<body>  
<?php  
echo strrev("Hello world!");  
>
```

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

iv) **Search for a specific string.**

```
<?php
$string = "This is a sample string to search.";
$wordToFind = "sample";

if (strpos($string, $wordToFind) !== false) {
    echo strpos($string, $wordToFind);
} else {
    echo "The word '$wordToFind' was not found in the string.";
}
?>
```

Experiment 8

Install TOMCAT web server. Convert the static web pages of assignments into dynamic web pages using servlets and cookies. Hint: User's information (user id, password, credit card number) would be stored in web.xml. Each user should have a separate Shopping Cart.

index.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>Login Form</title>
</head>
<body>
    <h2>Login</h2>
    <form action="LoginServlet" method="post">
        <label for="username">Enter username:</label>
        <input type="text" id="username" name="User" required><br><br>
        <label for="password">Enter password:</label>
        <input type="password" id="password" name="password" required><br><br>
        <label for="cardID">Enter card ID:</label>
        <input type="text" id="cardID" name="CardID" required><br><br>
```

```

        <input type="submit" value="Login">
    </form>
</body>
</html>

```

LoginServlet.java:

```

import java.io.*;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.Cookie;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
public class LoginServlet extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        String usr = req.getParameter("User");
        String pwd = req.getParameter("password");
        String card = req.getParameter("CardID");
        boolean flag = true;
        String[] userID = getInitParameter("usernames").split(",");
        String[] password = getInitParameter("passwords").split(",");
        String[] cardids = getInitParameter("cardIDs").split(",");
        for (int i = 0; i < userID.length; i++) {
            // Comparing the credentials. Consider using a secure authentication mechanism.
            if (userID[i].equals(usr) && password[i].equals(pwd) && cardids[i].equals(card)) {
                flag = false;
                // Creating cookies for the user session
                Cookie cookie1 = new Cookie("CurrentUser", usr);
                Cookie cookie2 = new Cookie("CreditCard", card);
                res.addCookie(cookie1);
                res.addCookie(cookie2);
                // Displaying welcome message and options for book purchase
                out.println("<h2>Welcome " + usr + "</h2><hr>");
                out.println("<h2>Select the book you would like to purchase</h2><hr>");
                out.println("<form action='LoginSuccess'>");
                out.println("<input type=radio name='book' checked value='Intro to Machine
Learning'/>Intro to Machine Learning<br>");
                out.println("<input type=radio name='book' value='Exploring Python'/>Exploring
Python<br>");
            }
        }
    }
}

```

```

        out.println("<input type=radio name='book' value='Mastering C'/>Mastering C<br>");
        out.println("<input type=submit value='purchase'><hr>");
        out.println("</form>"); // Close the form
    }
}
if (flag) {
    // Display an error message for invalid credentials
    out.println("<h4>Invalid user name or password or card number, please try again by
clicking the following link</h4>");
    out.println("<a href='index.html'>index.html</a>");
}
}
}

```

LoginSuccess.java:

```

import java.io.*;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.Cookie;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class LoginSuccess extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();

        // Checking for null values in cookies and parameters
        Cookie[] myCookies = req.getCookies();
        if (myCookies != null && myCookies.length >= 2 && req.getParameter("book") != null) {
            String book = req.getParameter("book");
            String userName = myCookies[0].getValue();
            String creditCard = myCookies[1].getValue();

            out.print("<h2>Welcome " + userName + "</h2><hr>");
            out.print("<h3>Thank you for purchasing book: " + book + "</h3><hr>");
            out.print("<h3>Rs.250 debited from credit card: " + creditCard + "</h3>");
        } else {

```



```

        // Error message in case of missing cookies or book parameter
        out.print("<h3>Error: Unable to process the request.</h3>");
    }
    out.close();
}
}

```

web.xml:

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="https://jakarta.ee/xml/ns/jakartaee"
xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee
https://jakarta.ee/xml/ns/jakartaee/web-app_6_0.xsd" version="6.0">
  <display-name>Experiment 8</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
  <servlet>
    <description></description>
    <display-name>LoginServlet</display-name>
    <servlet-name>LoginServlet</servlet-name>
    <servlet-class>LoginServlet</servlet-class>
    <init-param>
      <param-name>usernames</param-name>
      <param-value>sonu,vassu,krish</param-value>
    </init-param>
    <init-param>
      <param-name>passwords</param-name>
      <param-value>sonu,vassu,krish</param-value>
    </init-param>
    <init-param>
      <param-name>cardIDs</param-name>
      <param-value>111,222,333</param-value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name>LoginServlet</servlet-name>

```

```

    <url-pattern>/LoginServlet</url-pattern>
</servlet-mapping>
<servlet>
    <description></description>
    <display-name>LoginSuccess</display-name>
    <servlet-name>LoginSuccess</servlet-name>
    <servlet-class>LoginSuccess</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>LoginSuccess</servlet-name>
    <url-pattern>/LoginSuccess</url-pattern>
</servlet-mapping>
</web-app>

```

Experiment 9

Redo the previous task using JSP by converting the static web pages of assignments into dynamic web pages. Create a database with user information and books information. The books catalogue should be dynamically loaded from the database. Follow the MVC architecture while doing the website.

BookDAO.java:

```

package test;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class BookDAO {
    private Connection connection;

    public BookDAO() {
        String jdbcURL = "jdbc:mysql://localhost:3306/Exp9";
        String username = "root";
        String password = "Star_gazer0";

        try {

```

```

        Class.forName("com.mysql.cj.jdbc.Driver");
        connection = DriverManager.getConnection(jdbcURL, username, password);

        // Check if connection is valid
        if (connection.isValid(5)) {
            System.out.println("Connected!");
        } else {
            System.out.println("Failed to connect to the database.");
        }
    } catch (ClassNotFoundException | SQLException e) {
        e.printStackTrace();
    }
}

public String getBookDetailsByName(String bookName) {
    StringBuilder details = new StringBuilder();
    try {
        PreparedStatement preparedStatement = connection.prepareStatement("SELECT *
FROM books WHERE title = ?");
        preparedStatement.setString(1, bookName);
        ResultSet rs = preparedStatement.executeQuery();
        if (rs.next()) {
            details.append("<td>").append(rs.getInt("book_id")).append("</td>");
            details.append("<td>").append(rs.getString("title")).append("</td>");
            details.append("<td>").append(rs.getString("author")).append("</td>");
            details.append("<td>").append(rs.getString("publisher")).append("</td>");
            details.append("<td>").append(rs.getInt("publication_year")).append("</td>");
        } else {
            details.append("<td colspan='5'>Book not found</td>");
        }
        rs.close();
        preparedStatement.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return details.toString();
}

// Here, you can implement methods to perform CRUD operations for books
// Example methods: getBooks(), addBook(Book book), updateBook(Book book),
deleteBook(int bookId)

// Close the connection when not needed

```

```

public void closeConnection() {
    try {
        if (connection != null && !connection.isClosed()) {
            connection.close();
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}

```

BookServlet.java:

```

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 test.BookDAO;

```

```

@WebServlet("/BookServlet")
public class BookServlet extends HttpServlet {
    private BookDAO bookDAO;

```

```

    public void init() {
        bookDAO = new BookDAO();
    }

```

```

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // Get the book name entered by the user
        String bookName = request.getParameter("bookName");
        if (bookName != null && !bookName.isEmpty()) {
            // Retrieve book details based on the name from the DAO
            String bookDetails = bookDAO.getBookDetailsByName(bookName);
            request.setAttribute("bookDetails", bookDetails);
        }
        request.getRequestDispatcher("books.jsp").forward(request, response);
    }

```

```

    public void destroy() {
        bookDAO.closeConnection();
    }

```

```
}  
}
```

books.jsp:

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"  
%>  
<!DOCTYPE html>  
<html>  
<head>  
    <title>Book Catalog</title>  
</head>  
<body>  
    <h2>Book Catalog</h2>  
    <form action="BookServlet" method="get">  
        Enter Book Name: <input type="text" name="bookName">  
        <input type="submit" value="Search">  
    </form>  
    <br>  
    <table cellpadding="15" border="1" style="background-color: pink;">  
        <tr>  
            <th>Book ID</th>  
            <th>Title</th>  
            <th>Author</th>  
            <th>Publisher</th>  
            <th>Publication Year</th>  
        </tr>  
        <!-- Display book details retrieved from Servlet -->  
        <%  
            // Retrieve book details sent from the Servlet  
            String bookDetails = (String) request.getAttribute("bookDetails");  
            if (bookDetails != null) {  
                out.println("<tr>" + bookDetails + "</tr>");  
            }  
        %>  
    </table>  
</body>  
</html>
```

Experiment 10

Install a database (MySQL/Oracle). Create a table which should contain at least the following fields: Name, Password, Email-id, Phone Number (these should hold the data from the registration form). Practice 'JDBC' connectivity. Write a java program/servlet/JSP to connect to that database and extract data from the tables and display them. Experiment with various SQL queries. Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page.

register.html:

```
<html>
<body>
<form action="Register" method="post">
  Name:<input type="text" name="userName"/><br/><br/>
  Password:<input type="password" name="userPass"/><br/><br/>
  Email Id:<input type="text" name="userEmail"/><br/><br/>
  Phone no.:<input type="text" name="userPhone"/><br/><br/>
  <input type="submit" value="register"/>
</form>
</body>
</html>
```

Register.java:

```
package test;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
@WebServlet("/Register")
public class Register extends HttpServlet {
    @Override
    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("userPhone");
Connection con = null;
PreparedStatement ps = null;
try {
    Class.forName("com.mysql.cj.jdbc.Driver");
    con = DriverManager.getConnection(
        "jdbc:mysql://localhost:3306/Exp9", "root", "Star_gazer0");
    ps = con.prepareStatement(
        "INSERT INTO user (username, password, email, phone) 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.print("You are successfully registered...");
} catch (ClassNotFoundException | SQLException e2) {
    out.println("Error: " + e2.getMessage());
} finally {
    try {
        if (ps != null) {
            ps.close();
        }
        if (con != null) {
            con.close();
        }
    } catch (SQLException e3) {
        out.println("Error in closing resources: " + e3.getMessage());
    }
}
out.close();
}
}

```

web.xml:

```

<web-app>
<servlet>
<servlet-name>Register</servlet-name>
<servlet-class>Register</servlet-class>

```

```

</servlet>
<servlet-mapping>
<servlet-name>Register</servlet-name>
<url-pattern>/servlet/Register</url-pattern>
</servlet-mapping>
<welcome-file-list>
  <welcome-file>register.html</welcome-file>
</welcome-file-list>
</web-app>

```

Experiment 11

Write a JSP which does the following job: Insert the details of the 3 or 4 users who register with the web site by using registration form. Authenticate the user when he submits the login form using the user's name and password from the database.

register.html:

```

<!-- register.html -->
<html>
<body>
  <form action="Register" method="post">
    <p>Registration page</p>
    <label for="username">Username:</label>
    <input type="text" id="username" name="username" required><br/><br/>

    <label for="password">Password:</label>
    <input type="password" id="password" name="password" required><br/><br/>

    <input type="submit" value="Register">
  </form>
</body>
</html>

```

login.html:


```

<!-- login.html -->
<html>
<body>
    <form action="Login" method="post">
        <p>Login page</p>
        <label for="loginUsername">Username:</label>
        <input type="text" id="loginUsername" name="loginUsername" required><br/><br/>

        <label for="loginPassword">Password:</label>
        <input type="password" id="loginPassword" name="loginPassword" required><br/><br/>

        <input type="submit" value="Login">
    </form>
</body>
</html>

```

Login.java:

```

// Imports...
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
@WebServlet("/Login")
public class Login extends HttpServlet {
    @Override
    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String enteredUsername = request.getParameter("loginUsername");
        String enteredPassword = request.getParameter("loginPassword");
        // Code to check if the received username and password are in the database
        boolean isValidLogin = validateLogin(enteredUsername, enteredPassword);
        if (isValidLogin) {
            out.println("Login successful!"); // Display success message

```

```

    } else {
        out.println("Login failed. Invalid username or password."); // Display failure message
    }
}

private boolean validateLogin(String username, String password) {
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/Exp9", "root", "Star_gazer0");
        // Prepared statement to check if username and password exist in 'user' table
        PreparedStatement ps = con.prepareStatement("SELECT * FROM user WHERE
username=? AND password=?");
        ps.setString(1, username);
        ps.setString(2, password);
        ResultSet rs = ps.executeQuery();
        boolean isValidLogin = rs.next(); // If the result set has a record, login is valid
        con.close();
        return isValidLogin;
    } catch (ClassNotFoundException | SQLException e) {
        e.printStackTrace(); // Handle exceptions properly in production code
    }
    return false; // Login failed
}
}

```

Register.java:

```

// Register.java - Servlet for user registration
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
@WebServlet("/Register")
public class Register extends HttpServlet {
    @Override
    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

```

```

response.setContentType("text/html");
PrintWriter out = response.getWriter();
String username = request.getParameter("username");
String password = request.getParameter("password");
// Code to store the received username and password in the database
boolean registrationSuccess = saveUserToDatabase(username, password);
if (registrationSuccess) {
    // Registration successful
    response.sendRedirect("login.html"); // Redirect to login page
} else {
    out.println("Registration failed. Please try again.");
}
}

private boolean saveUserToDatabase(String username, String password) {
    // Implement database insertion logic here
    // Use PreparedStatement to safely insert data into the database
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/Exp9", "root", "Star_gazer0");
        // Prepared statement to insert username and password into 'user' table
        PreparedStatement ps = con.prepareStatement("INSERT INTO user (username,
password) VALUES (?, ?)");
        ps.setString(1, username);
        ps.setString(2, password);
        int rowsInserted = ps.executeUpdate();
        con.close();
        return rowsInserted > 0; // Registration successful if rows inserted > 0
    } catch (ClassNotFoundException | SQLException e) {
        e.printStackTrace(); // Handle exceptions properly in production code
    }
    return false; // Registration failed
}
}

```

Experiment 12

```

package test;
import javax.swing.*;
import java.awt.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

```

```

public class CustomVisualBean extends JPanel {
    private boolean shape; // Determines the shape: square (true) or circle (false)
    private Color color; // Color of the area
    public CustomVisualBean() {
        shape = false; // Default shape is square
        color = Color.RED; // Default color
        // Add mouse click listener to change color dynamically
        addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                color = getRandomColor();
                repaint();
            }
        });
    }
    public boolean isShape() {
        return shape;
    }
    public void setShape(boolean shape) {
        this.shape = shape;
        repaint();
    }
    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        int width = getWidth();
        int height = getHeight();
        g.setColor(color);
        if (shape) {
            g.fillRect(10, 10, width - 20, height - 20); // Draw square
        } else {
            g.fillOval(10, 10, width - 20, height - 20); // Draw circle
        }
    }
    private Color getRandomColor() {
        // Generate a random color
        int red = (int) (Math.random() * 256);
        int green = (int) (Math.random() * 256);
        int blue = (int) (Math.random() * 256);
        return new Color(red, green, blue);
    }
    public static void main(String[] args) {
        // For testing the bean
        JFrame frame = new JFrame();
        CustomVisualBean visualBean = new CustomVisualBean();
        visualBean.setPreferredSize(new Dimension(200, 200));
        frame.add(visualBean);
    }
}

```

```
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.pack();
frame.setVisible(true);
    }
}
```