

Index

S. No.	Objective	Page No.	Date	Sign.
1.	Write HTML/JavaScript to display your CV in navigator, your Institute Website, Department Website and Tutorial Website for specific subject.			
2.	Write an HTML program to design an entry form of student details and send it to store at database server like SQL, Oracle or MS Access.			
3.	Write program using Java Script for Web Page to display browsers information.			
4.	Write a Java applet to display the Application Program screen i.e. calculator and other.			
5.	Writing program in XML for creation of DTD, which specifies set of rules. Create a style sheet in CSS/XSL & display the document in internet explorer.			
6.	Program to illustrate JDBC connectivity. Program for maintaining database by sending queries. Design and implement a simple servlet book query with the help of JDBC & SQL. Create MS Access database, create ODBC link, compile & execute java JDVC socket.			
7.	Install TOMCAT web server and APACHE. Access the above developed static web pages for books web site, using these servers by putting the web pages developed.			

8.	Assume four users user1, user2, user3 and user4 having the passwords pwd1, pwd2, pwd3 and pwd4 respectively. Write a servlet for doing the following. Create a Cookie and add these four user id's and passwords to this Cookie. Read the user id and passwords entered in the login form and authenticate with the values available in the cookies.			
9.	Install a database (MySQL or Oracle). Create a table which should contain at least the following fields: name, password, email-id, phone number. Write a java program/servlet/JSP to connect to that database and extract data from the tables and display them. Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page.			

Objective:- Write HTML/JavaScript to display your CV in navigator, your Institute Website, Department Website and Tutorial Website for specific subject.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CV</title>
  </head>
  <body style="margin: auto; max-width: 800px; padding: 1rem;">
    <div>
      <p style="font-size: 12pt; text-align: center; border: 1px solid;">
        <a href="https://www.dbgisre.edu.in/">Dev Bhoomi Group of Institutions</a>
        | <a href="CV.html">Curriculum Vitae</a> |
        <a href="https://aktu.ac.in/">Dr. APJ. Abdul Kalam Technical University</a>
      </p>
      <section style="position: absolute; top: 50px;">
        <div style="position: absolute; left: 500px;">
          <p>
            <br>
          </p>
        </div>
        <div>
          <h1 style="font-size: 30pt"><i>Sanjay Singh</i></h1>
          <h4>
            8630570916 <br>
```

sanjaysingh26112000@gmail.com

Fatehpur Jutt, Nawada Road, Saharanpur

PIN Code - 247001

</h4>

</div>

<hr>

<hr>

<div style="float: left; width: 700px; height: 1800px;">

<h1 style="font-size: 20pt">Objective</h1>

<p style="font-size: 15pt; font-family: Bell MT;">

A motivated individual with in-depth knowledge of languages and
development tools,

seeking a position in a growth-oriented company where I can use my
skills to the

advantage of the company while having the scope to develop my own
skills.

</p>

<hr>

<h1 style="font-size: 15pt">EDUCATION AND QUALIFICATIONS:</h1>

<table style="border: 1px solid black; text-align: left; border-spacing: 10px; padding: 10px; ">

<tr>

<th>Qualification</th>

<th>Maximum Marks</th>

<th>Obtained Marks</th>

<th>Division</th>

<th>Institution</th>

<th>Board/University</th>

</tr>

<tr>

<td>B.Tech.</td>
<td>....</td>
<td>....</td>
<td>None</td>
<td>Dev Bhoomi Group of Institutions, Saharanpur</td>
<td>AKTU University</td>

<td>12th</td>
<td>500</td>
<td>428</td>
<td>First</td>
<td>Dr. BR. Ambedkar Inter College, Gayanagar, Saharanpur</td>
<td>UP Board</td>

<td>10th</td>
<td>600</td>
<td>517</td>
<td>First</td>
<td>N.S.P. Inter College, Fatehpur Jutt, Saharanpur</td>
<td>UP Board</td>

</table>

TECHNICAL EXPERIENCE:</h1> Languages: HTML, CSS, JAVASCRIPT, C, C++, PYTHON, JAVA
 Platforms: Windows 10, Linux


```

        Concepts: Networking, Operating Systems <br>
    </p>
    <hr>
    <h1 style="font-size: 15pt">PERSONAL SKILLS</h1>
    <p>
        1. Computer Proficiency <br>
        2. Problem Solving <br>
        3. Attention to Details
    </p>
    <hr>
    <h1 style="font-size: 15pt">LANGUAGES:</h1>
    <p>1. Hindi 2. English</p>
    <hr>
    <h1 style="font-size: 15pt">INTEREST:</h1>
    <p>
        1. Learning new programming language <br>
        2. Reading Books <br>
        3. Website development
    </p>
    <hr>
    <h1 style="font-size: 15pt">REFERENCE:</h1>
    <p>Will be provided on demand</p>
    <hr>
</div>
</section>
</div>
</body>
</html>
```

Objective: Write an HTML program to design an entry form of student details and send it to store at database server like SQL, Oracle or MS Access.

StudentDetailsForm.html

```
<!DOCTYPE html>

<html>

    <head>
        <title>Student Details Form</title>
    </head>

    <body>
        <h2>Student Details Form</h2>
        <form action="process.php" method="post">
            <label for="name">Name:</label>
            <input type="text" id="name" name="name"><br><br>

            <label for="rollno">Roll No:</label>
            <input type="text" id="rollno" name="roll_no"><br><br>

            <label for="email">Email:</label>
            <input type="email" id="email" name="email"><br><br>

            <label for="phone">Phone:</label>
            <input type="tel" id="phone" name="phone"><br><br>

            <label for="dob">Date of Birth:</label>
            <input type="date" id="dob" name="dob"><br><br>
```

```
<label for="gender">Gender:</label>
<input type="radio" id="male" name="gender" value="male">
<label for="male">Male</label>
<input type="radio" id="female" name="gender" value="female">
<label for="female">Female</label> <br><br>

<label for="address">Address:</label><br>
<textarea id="address" name="address" rows="4"
cols="30"></textarea><br><br>

<input type="submit" value="Submit">
<input type="reset" value="Reset">
</form>
</body>

</html>
```

process.php

```
<!DOCTYPE html>
<html>

  <head>
    <title>Process Details</title>
  </head>

  <body>
    <center>
      <?php
```



```

$conn = mysqli_connect("localhost", "root", "", "student");

if ($conn === false) {
    die("ERROR: Could not connect. "
        . mysqli_connect_error());
}

$name = $_REQUEST['name'];
$roll_no = $_REQUEST['roll_no'];
$email = $_REQUEST['email'];
$phone = $_REQUEST['phone'];
$dob = $_REQUEST['dob'];
$gender = $_REQUEST['gender'];
$address = $_REQUEST['address'];

$sql = "INSERT INTO studentdetails VALUES ('$name',
'$roll_no','$email','$phone','$dob', '$gender', '$address')";

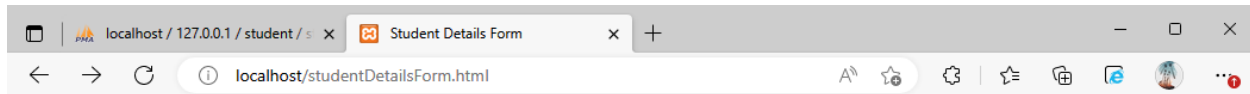
if (mysqli_query($conn, $sql)) {
    echo "<h3>data stored in a database successfully."
        . " Please browse your localhost php my admin"
        . " to view the updated data</h3>";

    echo nl2br("\n$name\n $roll_no\n "
        . "$email\n $phone\n $dob\n $gender\n $address");
} else {
    echo "ERROR: Hush! Sorry $sql. "
        . mysqli_error($conn);
}

```

```
        // Close connection
        mysqli_close($conn);
    ?>
</center>
</body>

</html>
```



Student Details Form

Name:

Roll No:

Email:

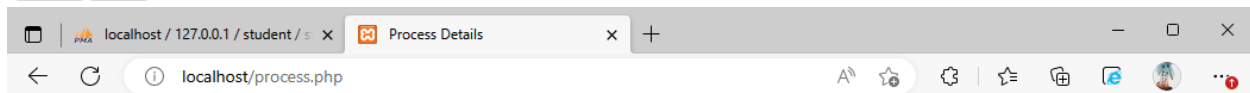
Phone:

Date of Birth:

Gender: ☒ Male ☐ Female

Address:

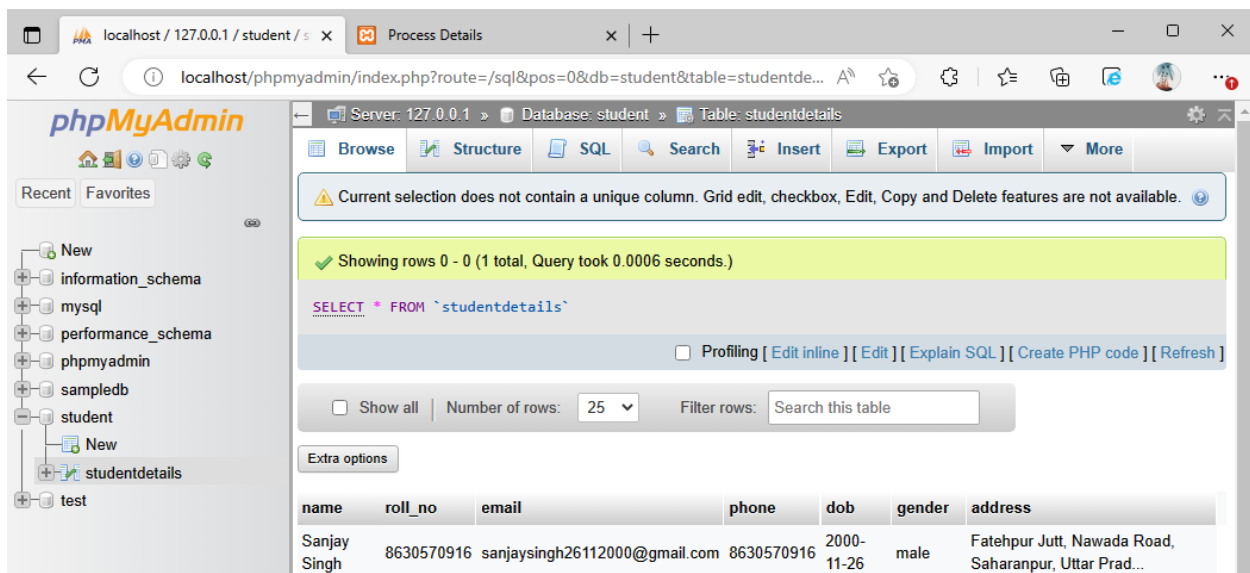
Fatehpur Jutt, Nawada Road,
Saharanpur, Uttar Pradesh



data stored in a database successfully. Please browse your localhost php my admin to view the updated data

Sanjay Singh
8630570916
sanjaysingh26112000@gmail.com
8630570916
2000-11-26
male

Fatehpur Jutt, Nawada Road, Saharanpur, Uttar Pradesh



Objective: Write program using Java Script for Web Page to display browsers information.

```
<!DOCTYPE html>

<html>

  <head>

    <title>Browser Information</title>

    <script type="text/javascript">

      function displayBrowserInfo() {

        document.getElementById("browser-info").innerHTML =

          "Codename: " + navigator.appCodeName + "<br>" +

          "Browser name: " + navigator.appName + "<br>" +

          "Version: " + navigator.appVersion + "<br>" +

          "Cookie enable: " + navigator.cookieEnabled + "<br>" +

          "Java Enable: " + navigator.javaEnabled + "<br>" +

          "Platform: " + navigator.platform + "<br>" +

          "Plugins: " + navigator.plugins + "<br>" +

          "Language: " + navigator.language + "<br>" +

          "Online: " + navigator.onLine + "<br>" +

          "User agent: " + navigator.userAgent;

      }

    </script>

  </head>

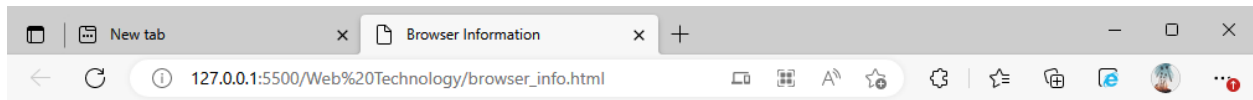
  <body onload="displayBrowserInfo()">
```

```
<h1>Browser Information</h1>
```

```
<div id="browser-info"></div>
```

```
</body>
```

```
</html>
```



Browser Information

Codename: Mozilla
Browser name: Netscape
Version: 5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.0.0 Safari/537.36 Edg/110.0.1587.41
Cookie enable: true
Java Enable: function javaEnabled() { [native code] }
Platform: Win32
Plugins: [object PluginArray]
Language: en-US
Online: false
User agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.0.0 Safari/537.36 Edg/110.0.1587.41

Objective: Write a Java applet to display the Application Program screen i.e. calculator and other.

Calculator.html

```
<!DOCTYPE html>

<html>

    <body>

        <applet code="Calculator.class" width="260" height="310"></applet>

    </body>

</html>
```

Calculator.java

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;

/* <applet code = "Calculator.class" width = "260" height = "310"></applet>
*/

public class Calculator extends Applet implements ActionListener {

    TextField t1;

    Button button1, button2, button3, button4, button5, button6, button7,
    button8, button9, button0;

    Button add, sub, mul, div, eql, dot;
```

```
String msg = "", tmp;
```

```
int a, b;
```

```
public void init() {
```

```
    setLayout(null);
```

```
    t1 = new TextField(20);
```

```
    button1 = new Button("1");
```

```
    button2 = new Button("2");
```

```
    button3 = new Button("3");
```

```
    button4 = new Button("4");
```

```
    button5 = new Button("5");
```

```
    button6 = new Button("6");
```

```
    button7 = new Button("7");
```

```
    button8 = new Button("8");
```

```
    button9 = new Button("9");
```

```
    button0 = new Button("0");
```

```
    add = new Button("+");
```

```
    sub = new Button("-");
```

```
    div = new Button("/");
```

```
    mul = new Button("*");
```

```
    dot = new Button(".");
```

```
    eql = new Button("=");
```



```
add(t1);
```

```
add(button7);
```

```
add(button8);
```

```
add(button9);
```

```
add(div);
```

```
add(button4);
```

```
add(button5);
```

```
add(button6);
```

```
add(mul);
```

```
add(button1);
```

```
add(button2);
```

```
add(button3);
```

```
add(sub);
```

```
add(dot);
```

```
add(button0);
```

```
add(eq1);
```

```
add(add);
```

```
t1.setBounds(30, 30, 200, 40);
```

```
button7.setBounds(30, 80, 44, 44);
```

```
button8.setBounds(82, 80, 44, 44);  
button9.setBounds(134, 80, 44, 44);  
button4.setBounds(30, 132, 44, 44);  
button5.setBounds(82, 132, 44, 44);  
button6.setBounds(134, 132, 44, 44);  
button1.setBounds(30, 184, 44, 44);  
button2.setBounds(82, 184, 44, 44);  
button3.setBounds(134, 184, 44, 44);  
dot.setBounds(30, 236, 44, 44);  
button0.setBounds(82, 236, 44, 44);  
eq1.setBounds(134, 236, 44, 44);  
add.setBounds(186, 236, 44, 44);  
sub.setBounds(186, 184, 44, 44);  
mul.setBounds(186, 132, 44, 44);  
div.setBounds(186, 80, 44, 44);
```

```
button0.addActionListener(this);  
button1.addActionListener(this);  
button2.addActionListener(this);  
button3.addActionListener(this);  
button4.addActionListener(this);  
button5.addActionListener(this);  
button6.addActionListener(this);  
button7.addActionListener(this);  
button8.addActionListener(this);
```

```

button9.addActionListener(this);

// button0.addActionListener(this);

// button0.addActionListener(this);

div.addActionListener(this);

mul.addActionListener(this);

add.addActionListener(this);

sub.addActionListener(this);

eq1.addActionListener(this);

}

```

```

public void actionPerformed(ActionEvent ae) {

    String str = ae.getActionCommand();

    if (str.equals("+") || str.equals("-") || str.equals("*") ||
str.equals("/")) {

        String str1 = t1.getText();

        tmp = str;

        a = Integer.parseInt(str1);

        msg = "";

    } else if (str.equals("=")) {

        String str2 = t1.getText();

        b = Integer.parseInt(str2);

        int sum = 0;

        if (tmp == "+")

            sum = a + b;

        else if (tmp == "-")

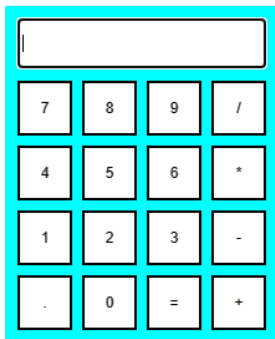
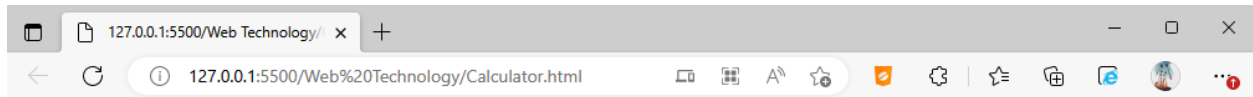
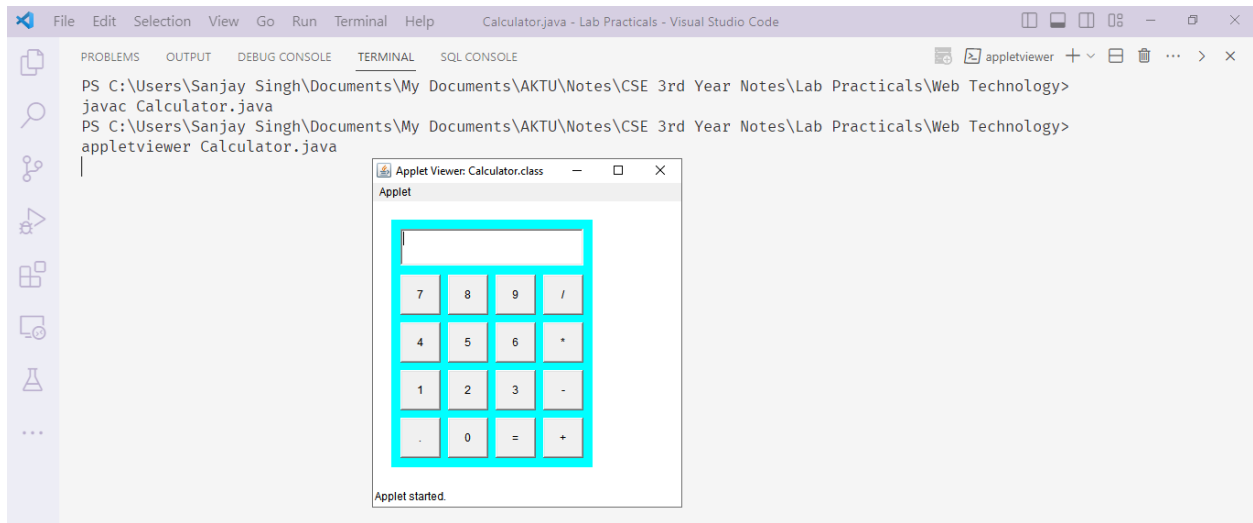
```

```

        sum = a - b;
    else if (tmp == "*")
        sum = a * b;
    else if (tmp == "/")
        sum = a / b;
    String str1 = String.valueOf(sum);
    t1.setText("" + str1);
    msg = "";
} else {
    // String ae.getActionCommand();
    // str += ae.getActionCommand();
    msg += str;
    t1.setText("" + msg);
}
}

public void paint(Graphics g) {
    g.setColor(Color.cyan);
    g.fillRect(20, 20, 220, 270);
}
}

```



Objective: Writing program in XML for creation of DTD, which specifies set of rules. Create a style sheet in CSS/XSL & display the document in internet explorer.

book.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="book.xsl"?>

<!DOCTYPE bookstore [
<!ELEMENT bookstore (book+)>
<!ELEMENT book (title, author, publisher, year, price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT publisher (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT price (#PCDATA)>
]>

<bookstore>
  <book>
    <title>The Cat in the Hat</title>
    <author>Dr. Seuss</author>
    <publisher>Random House</publisher>
    <year>1957</year>
    <price>$10.99</price>
  </book>
  <book>
    <title>Green Eggs and Ham</title>
    <author>Dr. Seuss</author>
    <publisher>Random House</publisher>
```

```

        <year>1960</year>
        <price>$8.99</price>
    </book>
    <book>
        <title>Where the Wild Things Are</title>
        <author>Maurice Sendak</author>
        <publisher>Harper & Row</publisher>
        <year>1963</year>
        <price>$12.99</price>
    </book>
</bookstore>

```

book.xsl

```

<?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>
            <body>
                <h3>Book Collection</h3>
                <link rel="stylesheet" type="text/css" href="book.css"/>
                <table>
                    <tr>
                        <th>Title</th>
                        <th>Author</th>
                        <th>Publisher</th>
                        <th>Year</th>
                        <th>Price</th>
                    </tr>
                    <xsl:for-each select="bookstore/book">
                        <tr>
                            <td><xsl:value-of select="title"></xsl:value-of></td>
                            <td><xsl:value-of select="author"></xsl:value-of></td>

```

```

        <td><xsl:value-of select="publisher"></xsl:value-of></td>
        <td><xsl:value-of select="year"></xsl:value-of></td>
        <td><xsl:value-of select="price"></xsl:value-of></td>
    </tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

book.css

```

table {
    border-collapse: collapse;
    width: 100%;
}

th,
td {
    text-align: left;
    padding: 8px;
}

tr:nth-child(even) {
    background-color: #f2f2f2;
}

th {
    background-color: #4caf50;
    color: white;
}

```


C:\Users\Sanjay Singh\Documents\My Documents\AKTU\Notes\CSE 3rd Year Notes\Lab Practicals\Web Technology\xml\book.xml

Search...

C:\Users\Sanjay Singh\Doc... ×

Book Collection

Title	Author	Publisher	Year	Price
The Cat in the Hat	Dr. Seuss	Random House	1957	\$10.99
Green Eggs and Ham	Dr. Seuss	Random House	1960	\$8.99
Where the Wild Things Are	Maurice Sendak	Harper & Row	1963	\$12.99

Objective: Program to illustrate JDBC connectivity. Program for maintaining database by sending queries. Design and implement a simple servlet book query with the help of JDBC & SQL. Create MS Access database, create ODBC link, compile & execute java JDVC socket.

JdbcConnectivityExample.java

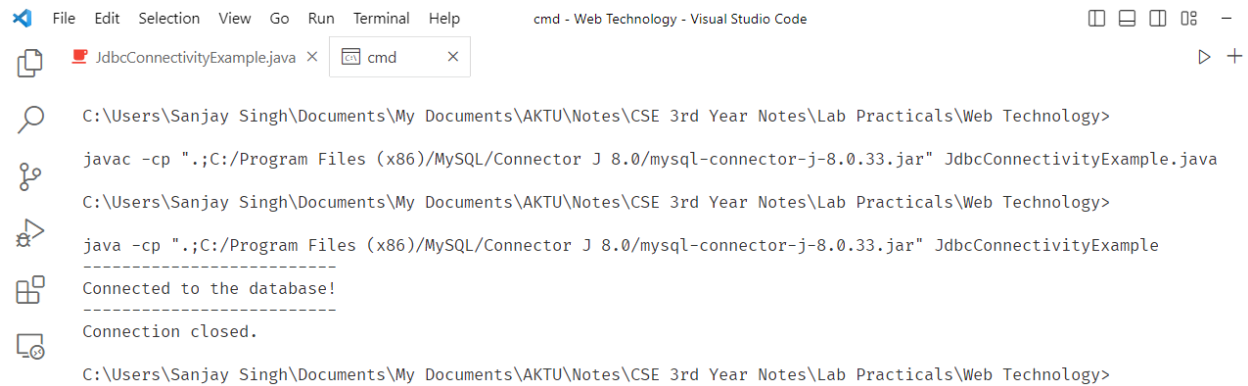
```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class JdbcConnectivityExample {

    public static void main(String[] args) {

        Connection connection = null;
        try {
            // Establish a connection
            connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydatabase",
"root", "pass");
            System.out.println("-----");
            System.out.println("Connected to the database!");
            System.out.println("-----");
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            // Close the connection in a finally block
            try {
                if (connection != null) {
                    connection.close();
                    System.out.println("Connection closed.");
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}
```

Output



```
cmd - Web Technology - Visual Studio Code

C:\Users\Sanjay Singh\Documents\My Documents\AKTU\Notes\CSE 3rd Year Notes\Lab Practicals\Web Technology>
javac -cp ".;C:/Program Files (x86)/MySQL/Connector J 8.0/mysql-connector-j-8.0.33.jar" JdbcConnectivityExample.java
C:\Users\Sanjay Singh\Documents\My Documents\AKTU\Notes\CSE 3rd Year Notes\Lab Practicals\Web Technology>
java -cp ".;C:/Program Files (x86)/MySQL/Connector J 8.0/mysql-connector-j-8.0.33.jar" JdbcConnectivityExample
-----
Connected to the database!
-----
Connection closed.

C:\Users\Sanjay Singh\Documents\My Documents\AKTU\Notes\CSE 3rd Year Notes\Lab Practicals\Web Technology>
```

DatabaseMaintenanceExample.java

```
import java.sql.*;

public class DatabaseMaintenanceExample {
    static final String JDBC_DRIVER = "com.mysql.cj.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost:3306/mydatabase";
    static final String USER = "root";
    static final String PASS = "pass";

    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;
        try {
            Class.forName(JDBC_DRIVER);
            conn = DriverManager.getConnection(DB_URL, USER, PASS);
            stmt = conn.createStatement();

            // Example queries
            String createTableQuery = "CREATE TABLE customers (id INT, name
VARCHAR(50))";
            String insertDataQuery = "INSERT INTO customers (id, name) VALUES (1, 'John
Doe')";
            String updateDataQuery = "UPDATE customers SET name = 'Jane Smith' WHERE id =
1";
            String deleteDataQuery = "DELETE FROM customers WHERE id = 1";

            // Execute queries
            stmt.executeUpdate(createTableQuery);
            stmt.executeUpdate(insertDataQuery);
            stmt.executeUpdate(updateDataQuery);
```

```

stmt.executeUpdate(deleteDataQuery);

System.out.println("-----");
System.out.println("Database maintenance operations completed successfully.");

stmt.close();
conn.close();
} catch (SQLException se) {
    se.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
} finally {
    try {
        if (stmt != null)
            stmt.close();
    } catch (SQLException se2) {
    }
    try {
        if (conn != null)
            conn.close();
    } catch (SQLException se) {
        se.printStackTrace();
    }
}
}
}
}

```

Output

```

File Edit Selection View Go Run ... cmd - Web Technology - Visual Studio Code
DatabaseMaintenanceExample.java cmd
C:\Users\Sanjay Singh\Documents\My Documents\AKTU\Notes\CSE 3rd Year Notes\Lab Practicals\Web Technology>
javac -cp ".;C:/Program Files (x86)/MySQL/Connector J 8.0/mysql-connector-j-8.0.33.jar" DatabaseMaintenanceExample.java
C:\Users\Sanjay Singh\Documents\My Documents\AKTU\Notes\CSE 3rd Year Notes\Lab Practicals\Web Technology>
java -cp ".;C:/Program Files (x86)/MySQL/Connector J 8.0/mysql-connector-j-8.0.33.jar" DatabaseMaintenanceExample
-----
Database maintenance operations completed successfully.
C:\Users\Sanjay Singh\Documents\My Documents\AKTU\Notes\CSE 3rd Year Notes\Lab Practicals\Web Technology>

```

BookQueryServlet.java

```
package BookQueryServlet;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.*;
import java.util.ArrayList;
import java.util.List;

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("/bookquery")
public class BookQueryServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

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

        Connection conn = null;
        Statement stmt = null;
        ResultSet rs = null;

        try {
            // Establish JDBC connection
            Class.forName("com.mysql.cj.jdbc.Driver");
            String url =
"jdbc:mysql://localhost:3306/mydatabase?useSSL=false&serverTimezone=UTC";
            String username = "root";
            String password = "pass";
            conn = DriverManager.getConnection(url, username, password);

            // Execute book query
            stmt = conn.createStatement();
            String query = "SELECT * FROM books";
            rs = stmt.executeQuery(query);
```

```

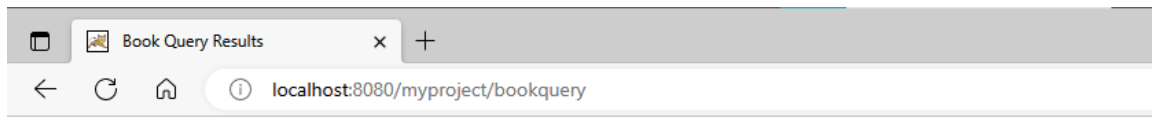
// Store book query results in a list
List<String> books = new ArrayList<>();
while (rs.next()) {
    String bookTitle = rs.getString("title");
    books.add(bookTitle);
}

// Generate HTML response
out.println("<html>");
out.println("<head><title>Book Query Results</title></head>");
out.println("<body>");
out.println("<h1>Book Query Results</h1>");
out.println("<ul>");
for (String book : books) {
    out.println("<li>" + book + "</li>");
}
out.println("</ul>");
out.println("</body></html>");

} catch (SQLException | ClassNotFoundException e) {
    e.printStackTrace();
} finally {
    // Close JDBC resources
    try {
        if (rs != null)
            rs.close();
        if (stmt != null)
            stmt.close();
        if (conn != null)
            conn.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}
}

```

Output



Book Query Results

- The Great Gatsby
- To Kill a Mockingbird
- 1984

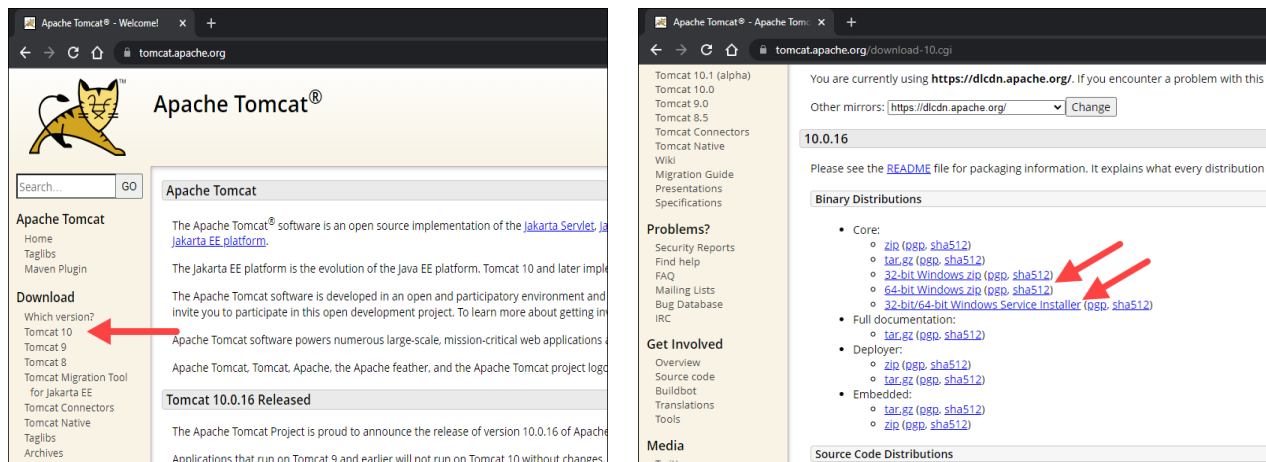
Objective: Install TOMCAT web server and APACHE. Access the above developed static web pages for books web site, using these servers by putting the web pages developed.

To install Apache Tomcat and access the static web pages for your books website, follow these steps:

1. Download Tomcat for Windows

To download the Tomcat installation file, follow the steps below:

1. Browse to the official Apache Tomcat website. Locate the Download section and click the latest Tomcat version available. At the time of writing this article, the latest Tomcat version was version 10.
2. On the Download page, scroll down and locate the Binary Distributions area. In the Core list, depending on the installation type you prefer, click the download link for the Windows Service Installer or the 32bit/64bit Windows zip file.



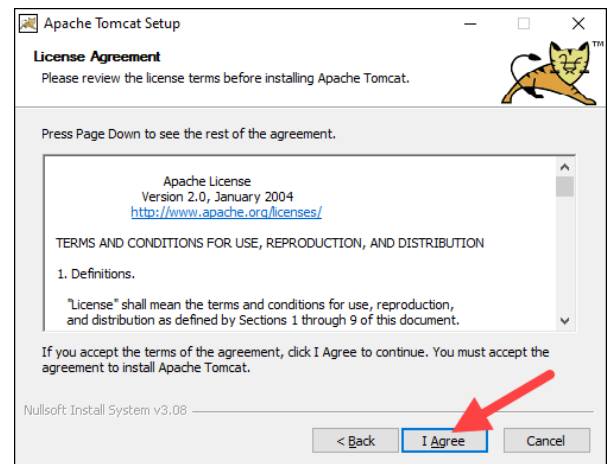
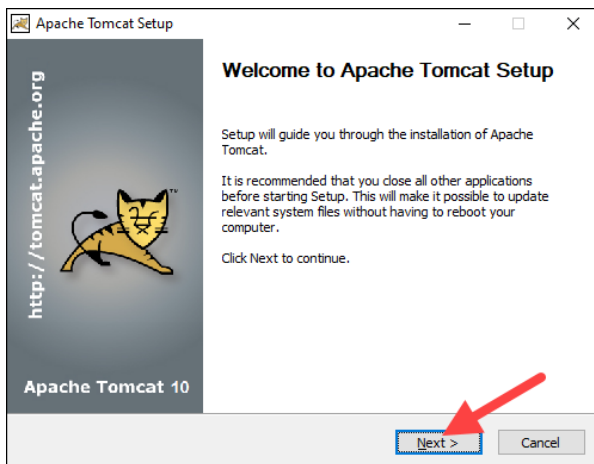
2. Install Tomcat

Install Tomcat via the Windows Service Installer for an automated and wizard-guided experience. The service installer installs the Tomcat service and runs it automatically when the system boots. For a portable experience, install Tomcat using the zip file and avoid installing the service. Easily uninstall Tomcat when it is no longer needed by deleting the Tomcat directory, or move it around when necessary.

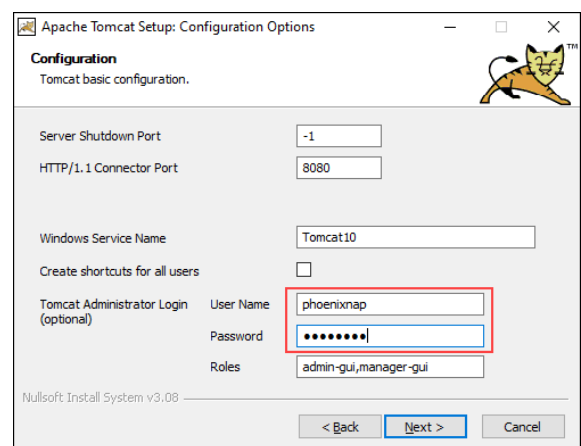
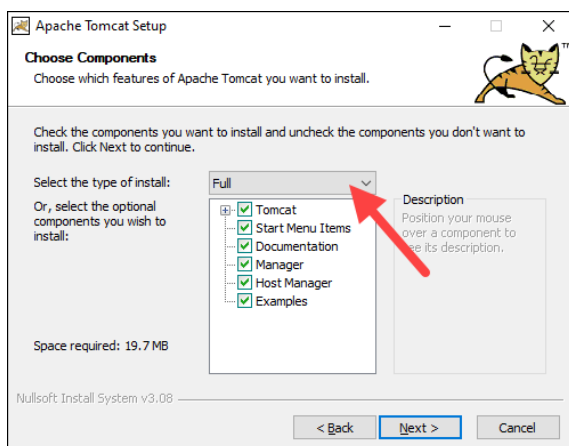
Follow the steps below to install Tomcat using the Windows Service Installer.

1. Open the downloaded Windows Service Installer file to start the installation process.
2. In the Tomcat Setup welcome screen, click Next to proceed.

3. Read the License Agreement and if you agree to the terms, click I Agree to proceed to the next step.

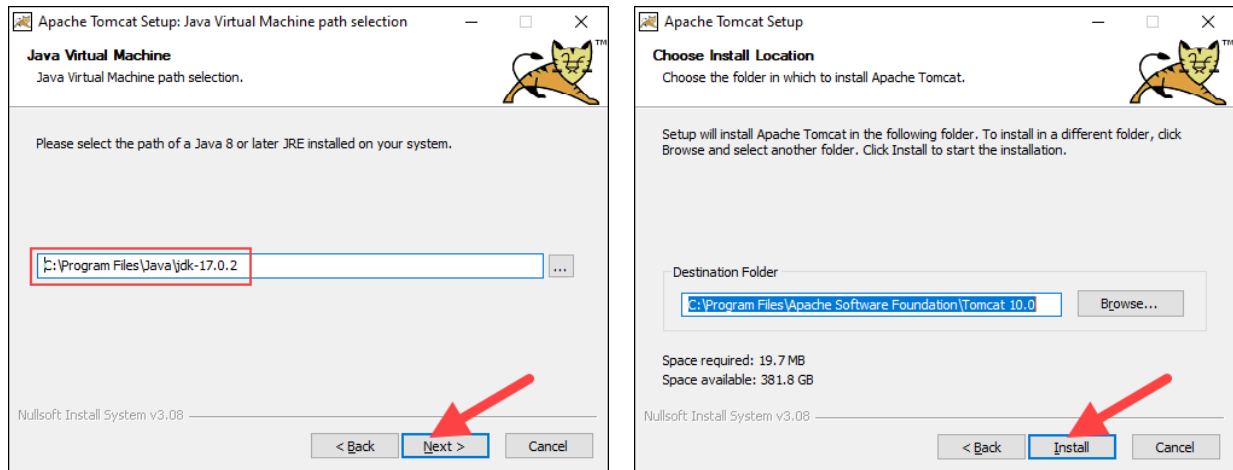


4. In the Tomcat component selection screen, choose Full in the dropdown menu to ensure the wizard installs the Tomcat Host Manager and Servlet and JSP examples web applications. Alternatively, keep the default Normal installation type and click **Next**.
5. The next step configures the Tomcat server. For instance, enter the Administrator login credentials or choose a different connection port. When finished, click Next to proceed to the next step.

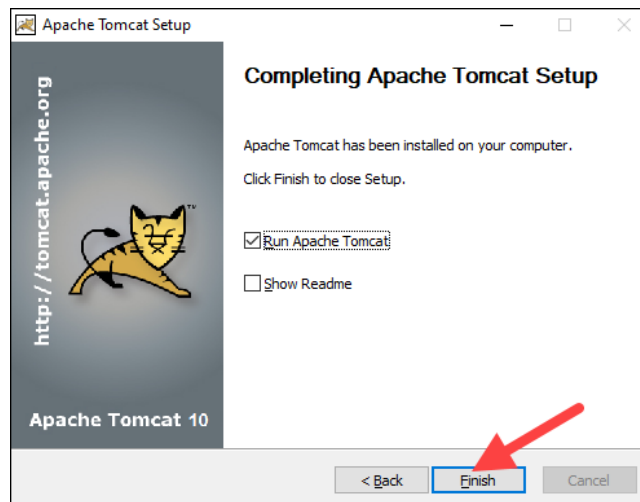


6. The next step requires you to enter the full path to the JRE directory on your system. The wizard auto-completes this if you have previously set up the Java environment variables. Click Next to proceed to the next step.

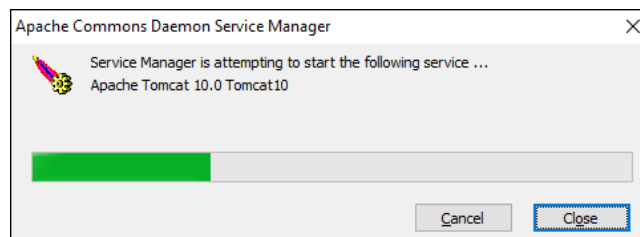
7. Choose the Tomcat server install location or keep the default one and click Install.



8. Check the Run Apache Tomcat box to start the service after the installation finishes. Optionally, check the Show Readme box to see the Readme file. To complete the installation, click Finish.

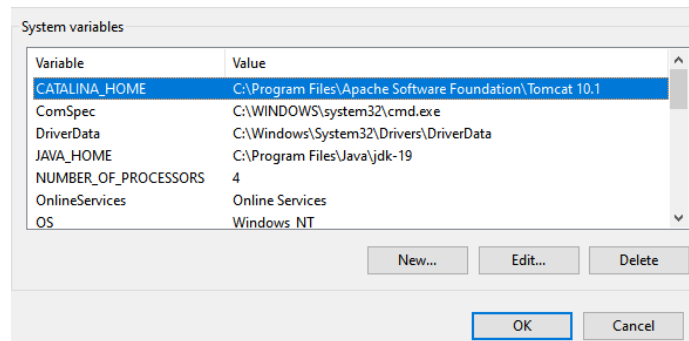


9. A popup window appears that starts the Tomcat service. After the process completes, the window closes automatically. The Apache Tomcat web server is now successfully installed.



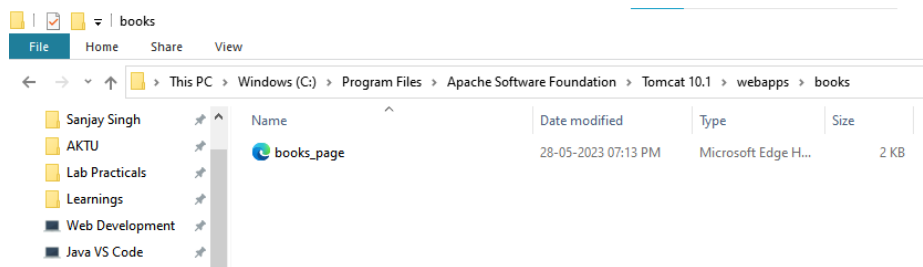
10. Optionally, set the CATALINA_HOME environment variable to point to the Tomcat installation directory. This step is not mandatory, but it can make it easier to manage and run Tomcat from the command line.

- Open the Control Panel and go to System and Security > System > Advanced system settings.
- Click on the "Environment Variables" button.
- In the "System Variables" section, click on the "New" button.
- Enter **CATALINA_HOME** as the variable name and the path to the Tomcat installation directory as the variable value.
- Click "OK" to save the variable.



3. Place your static web pages

- Copy your static web pages (HTML, CSS, JavaScript files, etc.) to the **webapps** directory inside your Tomcat installation directory.
- Create a subdirectory inside the **webapps** directory to hold your website files. For example, you can create a directory called books and place your web pages inside it.



4. Start Apache Tomcat

- Open a command prompt or terminal window. Run the startup.bat (Windows) script to start Tomcat.

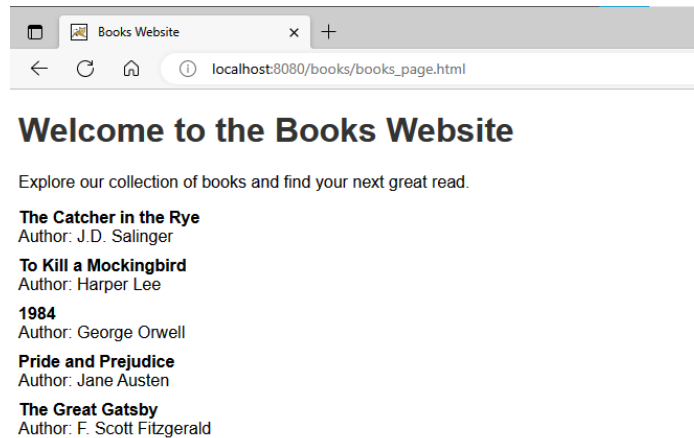
```

C:\Users\Sanjay Singh>startup.bat
Using CATALINA_BASE:  "C:\Program Files\Apache Software Foundation\Tomcat 10.1"
Using CATALINA_HOME:  "C:\Program Files\Apache Software Foundation\Tomcat 10.1"
Using CATALINA_TMPDIR: "C:\Program Files\Apache Software Foundation\Tomcat 10.1\temp"
Using JRE_HOME:       "C:\Program Files\Java\jdk-19"
Using CLASSPATH:      "C:\Program Files\Apache Software Foundation\Tomcat 10.1\bin\b
e Software Foundation\Tomcat 10.1\bin\tomcat-juli.jar"
Using CATALINA_OPTS:  ""
C:\Users\Sanjay Singh>

```

5. Access the books website

- Open a web browser.
- Enter this URL in the address bar: **http://localhost:8080/books/books_page.html**
- Press Enter to load the webpage.



Objective: Assume four users user1, user2, user3 and user4 having the passwords pwd1, pwd2, pwd3 and pwd4 respectively. Write a servlet for doing the following. Create a Cookie and add these four user id's and passwords to this Cookie. Read the user id and passwords entered in the login form and authenticate with the values available in the cookies.

CookieServlet.java

```
package com.example;

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.Cookie;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;

@WebServlet(urlPatterns = "/cookies")
public class CookieServlet extends HttpServlet {
    private static final String COOKIE_NAME = "UserCredentials";

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        String userCredentials = "user1:pwd1,user2:pwd2,user3:pwd3,user4:pwd4";

        try {
            String encodedCredentials = URLEncoder.encode(userCredentials, "UTF-8");

            Cookie cookie = new Cookie(COOKIE_NAME, encodedCredentials);
            cookie.setMaxAge(24 * 60 * 60); // Set cookie expiration time (in seconds)
            response.addCookie(cookie);
            response.getWriter().println("Cookie created successfully!");
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
    }
}
```

login.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Login</title>
</head>
<body>
  <form action="login" method="POST">
    <label for="userId">User ID:</label>
    <input type="text" id="userId" name="userId"><br><br>

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

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

LoginServlet.java

```
package com.example;

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.Cookie;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.net.URLDecoder;

@WebServlet(urlPatterns = "/login")
public class LoginServlet extends HttpServlet {
    private static final String COOKIE_NAME = "UserCredentials";

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
    ServletException, IOException {
        String userId = request.getParameter("userId");
        String password = request.getParameter("password");

        // Retrieve the stored cookie
```

```

Cookie[] cookies = request.getCookies();
Cookie userCredentials = findCookie(cookies, COOKIE_NAME);

// Authenticate the login credentials with the values available in the cookie
if (userCredentials != null) {
    String storedCredentials = URLDecoder.decode(userCredentials.getValue(), "UTF-8");
    String[] credentialsArray = storedCredentials.split(",");

    for (String credentials : credentialsArray) {
        String[] userAndPassword = credentials.split(":");
        String storedUserId = userAndPassword[0].trim();
        String storedPassword = userAndPassword[1].trim();

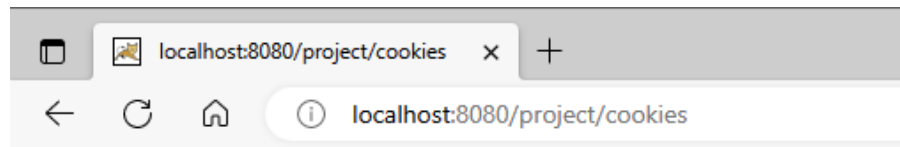
        if (storedUserId.equals(userId) && storedPassword.equals(password)) {
            // Authentication successful
            response.getWriter().println("Login Successful!");
            return;
        }
    }
}

// Authentication failed
response.getWriter().println("Invalid credentials. Please try again.");
}

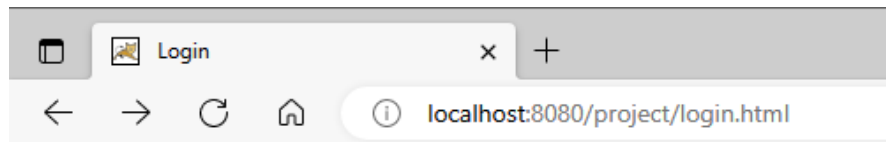
private Cookie findCookie(Cookie[] cookies, String name) {
    if (cookies != null) {
        for (Cookie cookie : cookies) {
            if (cookie.getName().equals(name)) {
                return cookie;
            }
        }
    }
    return null;
}
}

```

Output

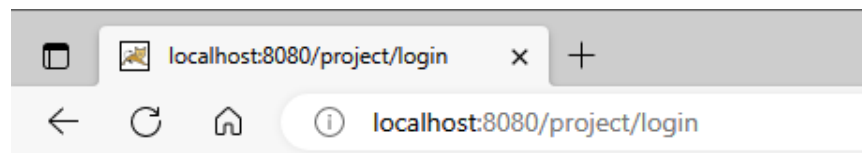


Cookie created successfully!



User ID:

Password:



Login Successful!

Objective: Install a database (MySQL or Oracle). Create a table which should contain at least the following fields: name, password, email-id, phone number. Write a java program/servlet/JSP to connect to that database and extract data from the tables and display them. Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page.

Creating table in MySQL

```
CREATE TABLE users (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(50) NOT NULL,  
    password VARCHAR(50) NOT NULL,  
    email_id VARCHAR(100) NOT NULL,  
    phone_number VARCHAR(20) NOT NULL  
);
```

Index.jsp

```
<!DOCTYPE html>  
<html>  
<head>  
    <title>User Registration</title>  
</head>  
<body>  
    <h2>User Registration</h2>  
    <form action="UserServlet" method="post">  
        <label for="name">Name:</label>  
        <input type="text" id="name" name="name" required><br><br>  
  
        <label for="password">Password:</label>  
        <input type="password" id="password" name="password" required><br><br>  
  
        <label for="email">Email:</label>  
        <input type="email" id="email" name="email" required><br><br>  
  
        <label for="phone">Phone Number:</label>  
        <input type="text" id="phone" name="phone" required><br><br>  
  
        <input type="submit" value="Register">
```

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

UserServlet.java

```
package UserServlet;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.*;

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("/UserServlet")
public class UserServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static final String DB_URL = "jdbc:mysql://localhost:3306/mydatabase";
    private static final String DB_USERNAME = "root";
    private static final String DB_PASSWORD = "pass";

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

        try {
            // Connect to the database
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection connection = DriverManager.getConnection(DB_URL, DB_USERNAME,
DB_PASSWORD);

            // Execute SELECT query to retrieve user data
            Statement statement = connection.createStatement();
            ResultSet resultSet = statement.executeQuery("SELECT * FROM users");

            // Display retrieved user data
            out.println("<html><body>");
```

```

        out.println("<h2>User Details:</h2>");
        out.println("<table
border=\"1\"><tr><th>ID</th><th>Name</th><th>Password</th><th>Email</th><th>Phone
Number</th></tr>");

        while (resultSet.next()) {
            int id = resultSet.getInt("id");
            String name = resultSet.getString("name");
            String password = resultSet.getString("password");
            String email = resultSet.getString("email_id");
            String phoneNumber = resultSet.getString("phone_number");

            out.println("<tr><td>" + id + "</td><td>" + name + "</td><td>" + password +
"</td><td>" + email
                + "</td><td>" + phoneNumber + "</td></tr>");
        }

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

        // Clean up resources
        resultSet.close();
        statement.close();
        connection.close();
    } catch (ClassNotFoundException | SQLException e) {
        e.printStackTrace();
        out.println("An error occurred while processing the request.");
    }
}

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

    String name = request.getParameter("name");
    String password = request.getParameter("password");
    String email = request.getParameter("email");
    String phoneNumber = request.getParameter("phone");

    try {
        // Connect to the database

```

```

        Class.forName("com.mysql.jdbc.Driver");
        Connection connection = DriverManager.getConnection(DB_URL, DB_USERNAME,
DB_PASSWORD);

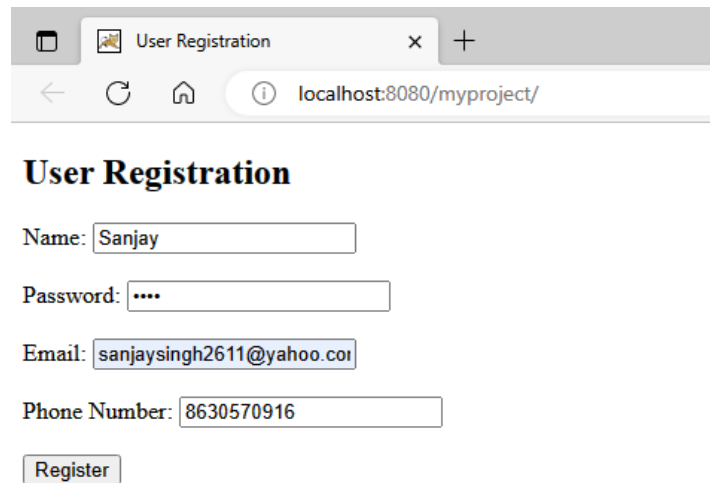
        // Execute INSERT query to add new user details
        PreparedStatement preparedStatement = connection
            .prepareStatement("INSERT INTO users (name, password, email_id, phone_number)
VALUES (?, ?, ?, ?)");
        preparedStatement.setString(1, name);
        preparedStatement.setString(2, password);
        preparedStatement.setString(3, email);
        preparedStatement.setString(4, phoneNumber);

        int rowsInserted = preparedStatement.executeUpdate();
        if (rowsInserted > 0) {
            out.println("User details inserted successfully!");
        } else {
            out.println("Failed to insert user details.");
        }

        // Clean up resources
        preparedStatement.close();
        connection.close();
    } catch (ClassNotFoundException | SQLException e) {
        e.printStackTrace();
        out.println("An error occurred while processing the request.");
    }
}
}

```

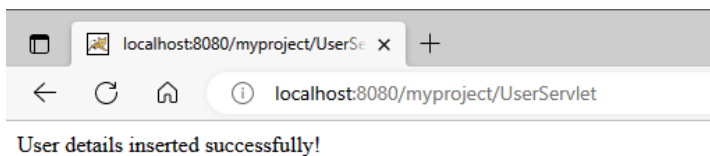
Output



A screenshot of a web browser window with the title "User Registration". The address bar shows "localhost:8080/myproject/". The page contains a registration form with the following fields:

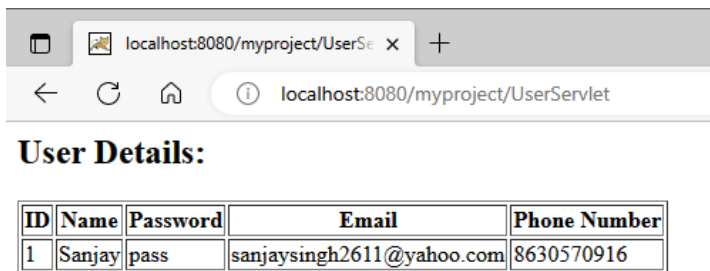
- Name:
- Password:
- Email:
- Phone Number:

Below the fields is a "Register" button.



A screenshot of a web browser window with the title "localhost:8080/myproject/UserSe". The address bar shows "localhost:8080/myproject/UserServlet". The page displays the message:

User details inserted successfully!



A screenshot of a web browser window with the title "localhost:8080/myproject/UserSe". The address bar shows "localhost:8080/myproject/UserServlet". The page displays the heading "User Details:" followed by a table:

ID	Name	Password	Email	Phone Number
1	Sanjay	pass	sanjaysingh2611@yahoo.com	8630570916