ALGORITHM:

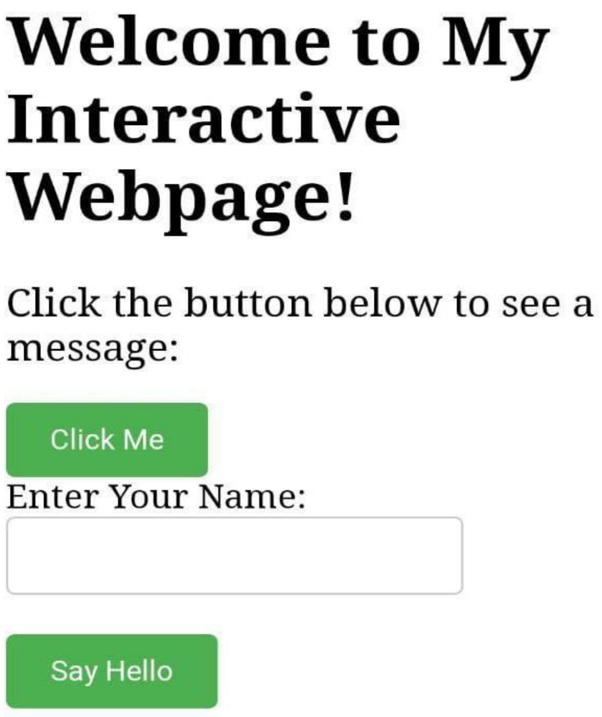
1. Define the purpose of the website and identify the target audience.
2. Plan the layout of the website, including the pages and navigation.
3. Choose an authoring tool, such as Dream weaver or Word Press, to create the website.
4. Use HTML to design the pages of the website.
5. Add interactive elements, such as forms and buttons, to engage the user.
6. Optimize the website for search engines and accessibility.
7. Test the website on different devices and browsers.
8. Publish the website to a web hosting service

PROGRAM:

```
<!DOCTYPE html>
<html>
<head>
  <title>Interactive Webpage</title>
  <style>
    button {
      background-color: #4CAF50;
      color: white;
      padding: 10px 20px;
      border: none;
      border-radius: 4px;
      cursor: pointer;
    }
    input[type=text] {
      padding: 10px;
      border: 1px solid #ccc;
      border-radius: 4px;
      box-sizing: border-box;
    }
  </style>
</head>
<body>
```

```
  <h1>Welcome to My Interactive Webpage!</h1>

  <p>Click the button below to see a message:</p>

  <button onclick="alert('Hello World!')">Click Me</button>

  <form>

    <label for="name">Enter Your Name:</label>

    <input type="text" id="name" name="name"><br><br>

    <button type="button" onclick="alert('Hello ' +
document.getElementById('name').value)">Say Hello</button>

  </form>

</body>

</html>
```

OUTPUT:



Welcome to My Interactive Webpage!

Click the button below to see a message:

Enter Your Name:

ALGORITHM:

1. Create an HTML file with buttons for each operation.
2. Use JavaScript functions for addition, subtraction, multiplication, and division.
3. Prompt the user to enter two numbers.
4. Convert the input values to integers using parseInt().
5. Perform the selected arithmetic operation.
6. Display the result using an alert() box.
7. Save and open the file in a web browser to view the output.

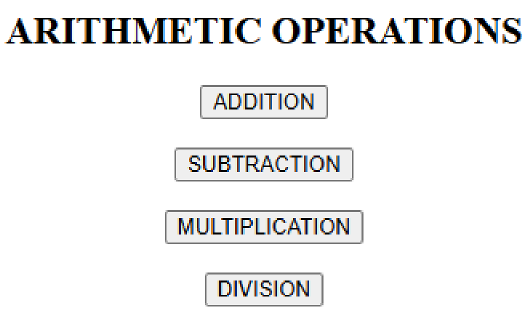
PROGRAM:

Arithmetic.html:

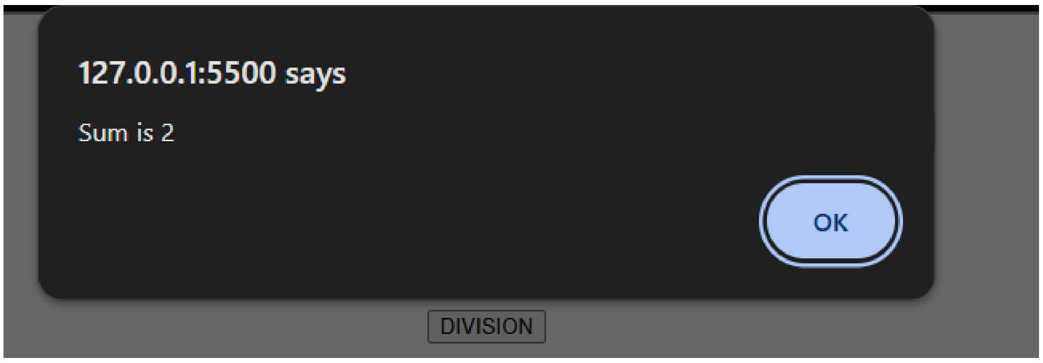
```
<!DOCTYPE html>

<html>

<head>

<title>Arithmetic Operations</title>

<script type="text/javascript">

        function add() {

            var n1 = window.prompt("Enter the value of A:", "");

            var a = parseInt(n1);

            var n2 = window.prompt("Enter the value of B:", "");

            var b = parseInt(n2);

            var c = a + b;

alert("Sum is " + c);

        }

        function sub() {

            var n1 = window.prompt("Enter the value of A:", "");

            var a = parseInt(n1);

            var n2 = window.prompt("Enter the value of B:", "");

            var b = parseInt(n2);

            var c = a - b;

alert("Difference is " + c);

        }

        function mul() {

            var n1 = window.prompt("Enter the value of A:", "");
            var a = parseInt(n1);

            var n2 = window.prompt("Enter the value of B:", "");

            var b = parseInt(n2);

            var c = a * b;

alert("Product is " + c);

        }

        function div() {

            var n1 = window.prompt("Enter the value of A:", "");

            var a = parseInt(n1);

            var n2 = window.prompt("Enter the value of B:", "");

            var b = parseInt(n2);

            if (b === 0) {

alert("Division by zero is not allowed!");

            } else {

                var c = a / b;

alert("Quotient is " + c);

            }

        }

</script>

</head>

<body>

<form>

<h2 style="text-align:center;">ARITHMETIC OPERATIONS</h2>

<div style="text-align:center;">
```

```
<input type="button" value="ADDITION" onclick="add()"><br><br>

<input type="button" value="SUBTRACTION" onclick="sub()"><br><br>

<input type="button" value="MULTIPLICATION" onclick="mul()"><br><br>

<input type="button" value="DIVISION" onclick="div()"><br><br>

</div>

</form>

</body>

</html>
```

OUTPUT:



ARITHMETIC OPERATIONS



127.0.0.1:5500 says

Sum is 2

ALGORITHMS:

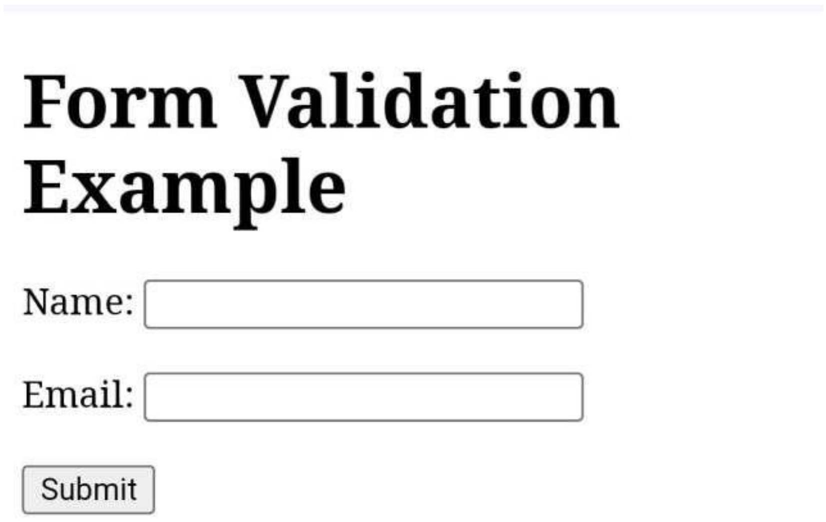
1. Get the form element from the HTML document using JavaScript.
2. Add an event listener to the form element that listens for the submit event.
3. Inside the event listener function, prevent the default form submission behavior using event.preventDefault().
4. Retrieve the values of the form fields using JavaScript and validate them using conditional statements.
5. If any field is in valid, display an error message and prevent the form from submitting.
6. If all fields are valid, submit the form.

PROGRAM:

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<head>
<title>Form Validation Example</title>
<script>
function validateForm()
{
var name = document.forms[“myForm”][“name”].value;var
email=document.forms[“myForm”][“email”].value;
if(name==””){
alert(“Name must be filled out”);
return false;
}
if(email==””)
{
alert(“Email must be filled out”);
return false;
}
if(email.indexOf(“@”) == -1 || email.indexOf(“.”) == -1) {alert(“Please enter a valid
emailaddress”);
return false;
}
}
</script>
</head>
<body>
<h1>Form Validation Example</h1>
<form name=”myForm”onsubmit=”returnvalidateForm()”>
<label for=”name”>Name:</label>
<input type=”text” id=”name” name=”name”><br><br>
<label for=”email”>Email:</label>
```

```
<input type=”text” id=”email” name=”email”><br><br>
<input type=”submit” value=”Submit”>
</form>
```

OUTPUT:



ALGORITHMS:

1. **Identify the purpose of the PHP script:**
   Determine what the script should accomplish — whether it’s processing user input, retrieving data from a database, or generating dynamic content.

2. **Set up the PHP environment:**
   Install PHP and configure your web server (e.g., Apache, Nginx, or XAMPP) to support PHP scripts.

3. **Create a new PHP file:**
   Use a text editor or an Integrated Development Environment (IDE) to create a new file with a .php extension.

4. **Start with the PHP opening tag:**
   Begin the script with the PHP opening tag: <?php

5. **Write PHP code:**
   Write the code that performs the desired task — such as processing form data, querying a database, or generating HTML content.

6. **Test the PHP script:**
   Save the file on your web server and open it in a web browser to verify that it works correctly.

7. **Debug and refine the script:**
   If it doesn’t work as expected, use error messages, echo statements, or debugging tools to identify and fix issues.

8. **Add comments and documentation:**
   Include comments in your code to explain its purpose and make it easier to understand and maintain in the future.

9. **Refactor and optimize:**
   Improve your code for efficiency and optimize it for better performance.

10. **Deploy the PHP script:**
    Upload the finalized PHP script to your production web server and integrate it into the website or application as needed.

PROGRAM:

```
<!DOCTYPE html>
<html>
<head>
   <title>Simple PHP Script</title>
</head>
<body>
   <h1>Current Date and Time</h1>

   <?php
      echo "The current date and time is: " . date("Y-m-d h:i:sa");
   ?>

</body>
</html>
```

OUTPUT:

```
Current Date and Time

The current date and time is: 2022-11-1114:45:30
```

ALGORITHM:

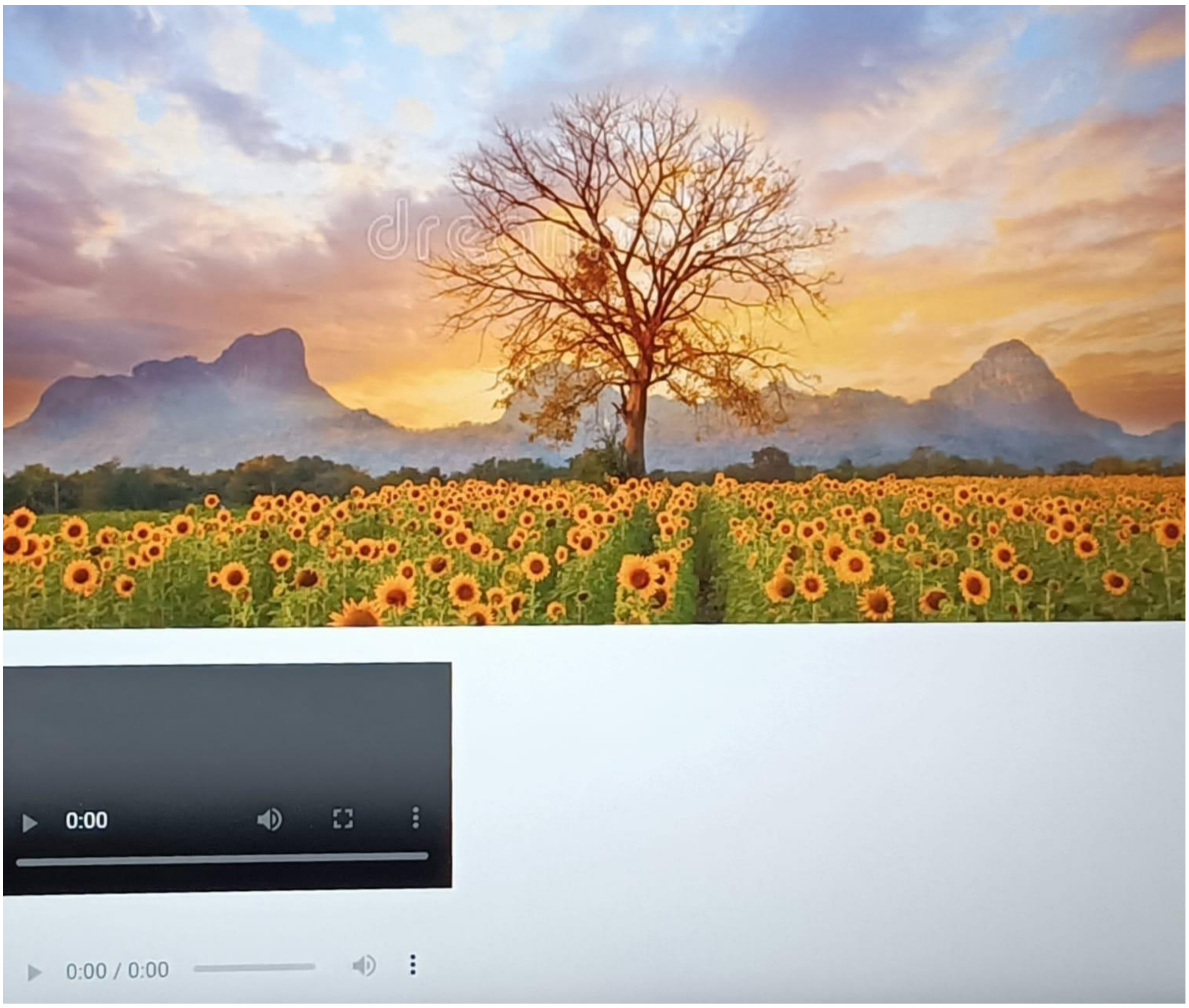
1. **Determine multimedia content types:**
   Identify the types of multimedia that will be used on the website, such as images, videos, audio files, and animations.
2. **Optimize multimedia files for the web:**
   Compress multimedia files without compromising quality. Use suitable tools such as image compression tools (e.g., TinyPNG), video compression tools (e.g., HandBrake), and audio compression tools to reduce file size and improve load time.
3. **Implement lazy loading:**
   Improve loading speed by implementing lazy loading — multimedia content is only loaded when needed, such as when the user scrolls to that section of the page.
4. **Use responsive design techniques:**
   Ensure that multimedia content displays properly on various devices and screen sizes.
   - Use responsive image and video tags.
   - Apply CSS to adjust size and positioning of multimedia elements dynamically.
5. **Provide an intuitive multimedia interface:**
   Create user-friendly controls that allow users to interact with multimedia content — such as play, pause, adjust volume, or change playback speed.
6. **Ensure accessibility:**
   Make multimedia content accessible for users with disabilities by:
   - Adding captions and transcripts for videos and audio.
   - Including descriptive alt tags for images.

7. **Monitor and optimize continuously:**
   Regularly track website performance and gather user feedback to continuously enhance multimedia content for a better user experience.

PROGRAM:

```
<!DOCTYPE html>
<html>
<head>
  <title>Media Example</title>
</head>
<body>
  <img src="https://thumbs.dreamstime.com/b/beautiful-landscape-dry-tree-branch-sun-flowers-field-against-colorful-evening-dusky-sky-use-as-natural-background-backdrop-48319427.jpg"
       alt="An example image" width="500"><br><br>
  <video controls width="500">
    <source src="video.mp4" type="video/mp4">
    <source src="video.webm" type="video/webm">
    Your browser does not support the video tag.
  </video>
  <br><br>

  <audio controls>
    <source src="audio.mp3" type="audio/mpeg">
    <source src="audio.ogg" type="audio/ogg">
    Your browser does not support the audio tag.
  </audio>
</body>
</html>
```

OUTPUT:



PROCEDURE:

client.html:

1. Create a web page using HTML form that contains the fields such as text, password and one submit button.
2. Set the URL of the server as the value of form's action attribute.
3. Run the HTML program.
4. Submit the form data to the server.

server.java:

1. Define the class server that extends the property of the class HttpServlet
2. Handle the request from the client by using the method service() of HttpServlet class.
3. Get the parameter names from the HTML form by using the method getParameterNames().
4. Get the parameter values from the HTML forms by using the method getParameter().
5. Send the response to the client by using the method of PrintWriter class.

PROGRAM:

MySrv.java:

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class MySrv extends HttpServlet {
public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html");
PrintWriter out = response.getWriter();
out.println("<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.01
Transitional//EN\">");
out.println("<HTML>");
out.println(" <HEAD><TITLE>A Servlet</TITLE></HEAD>");
out.println(" <BODY>");

//Getting HTML parameters from Servlet

String username=request.getParameter("uname");
String password=request.getParameter("pwd");
if((username.equals("user")) && (password.equals("pswd")))
{
out.println(" <h1> Welcome to "+username);
}
else
{
out.println(" <h1> Registration success ");
out.println(" <a href='./index.html'> Click for Home page </a>");
}
out.println(" </BODY>");
out.println("</HTML>");
out.close();
}
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
doPost( request,response);
}
}
```

Registration.html:

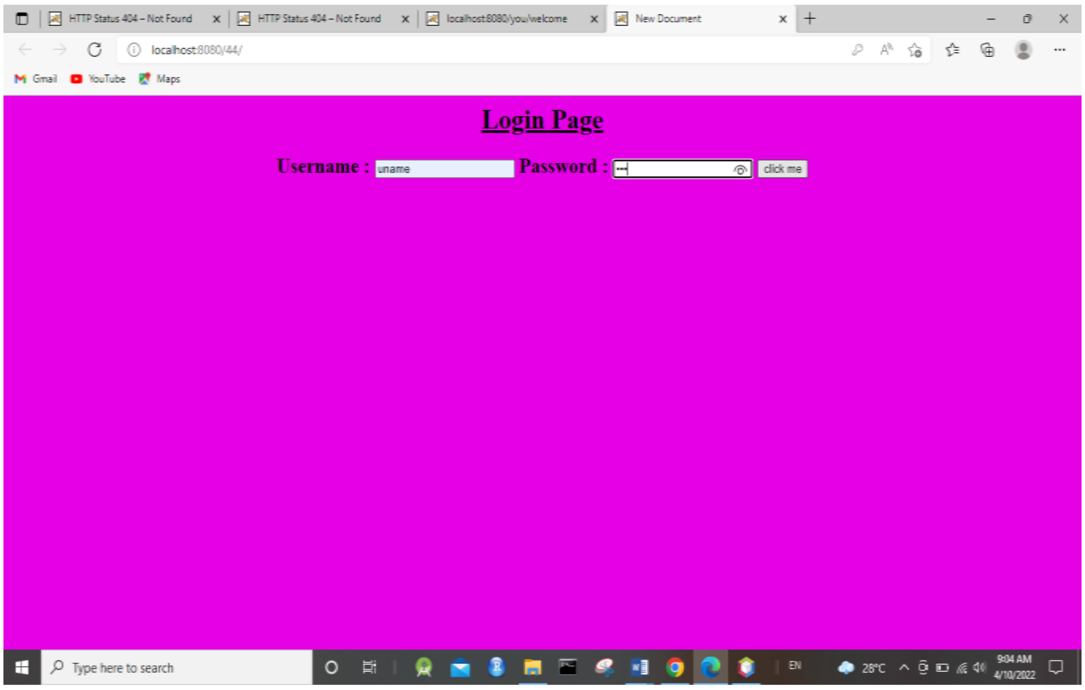
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE> New Document </TITLE>
<META NAME="Generator" CONTENT="EditPlus">
<META NAME="Author" CONTENT="">
<META NAME="Keywords" CONTENT="">
<META NAME="Description" CONTENT="">
</HEAD>
<BODY bgcolor='#e600e6'>
<form action='./MySrv' method="post">
<center><h1><u> Login Page </u></h1>
<h2> Username : <input type="text" name="uname"/>
        Password : <input type="password" name="pwd"/>
<input type="submit" value="click me"/>
</center>
</form>
</body>
</HTML>
```

web.xml:

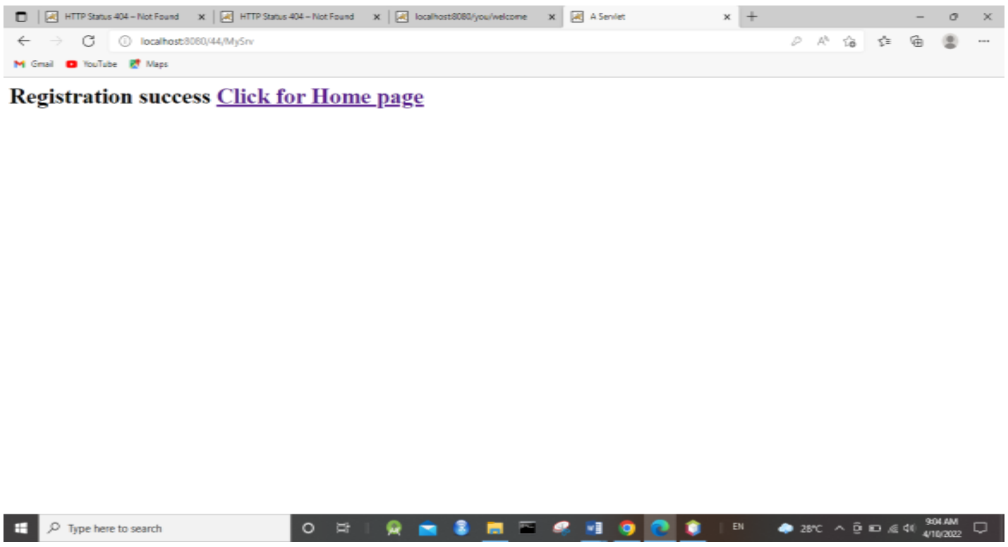
```
<web-app>
<servlet>
<servlet-name>MySrv</servlet-name>
<servlet-class>MySrv</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>MySrv</servlet-name>
<url-pattern>/MySrv</url-pattern>
</servlet-mapping>
<welcome-file-list>
```

```
<welcome-file>index.html</welcome-file>
</welcome-file-list>
</web-app>
```

OUTPUT:


Login Page

Username : uname Password : ••• click me


Registration success Click for Home page

PROCEDURE:

1. Create a web page using HTML form that contains the fields such as text, password and one submit button.
2. Set the URL of the server as the value of form's action attribute.
3. Ask if the user wants to add more items or check out.
4. Include the current items as hidden fields so they'll be passed on and submit to self.

PROGRAM:

register.html:

```
<html>
<body bgcolor = "cyan">
<center>
<h1>WELCOME TO REGISTRATION PAGE</h1>
<form action="./registerone" METHOD="post">
Name: <input type="text" name = "name"><br><br>
Password: <input type="password" name="password"><br><br>
PROFESSION:
<select name="profession">
<option value="engineer">ENGINEER</option>
<option value="teacher">TEACHER</option>
<option value="businessman">BUSINESSMAN</option>
</select><br><br>
<input type="submit" value="REGISTER">
</form>
</center>
</body>
</html>
```

web.xml

```
<web-app>
<welcome-file-list>
<welcome-file>register.html</welcome-file>
</welcome-file-list>
<servlet>
<servlet-name>RegistrationServletOne</servlet-name>
<servlet-class>RegistrationServletOne</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>RegistrationServletOne</servlet-name>
<url-pattern>/registerone</url-pattern>
</servlet-mapping>
<servlet>

<servlet-name>RegistrationServletTwo</servlet-name>
<servlet-class>RegistrationServletTwo</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>RegistrationServletTwo</servlet-name>
<url-pattern>/registertwo</url-pattern>
</servlet-mapping>
</web-app>
```

RegistrationServletOne.java:

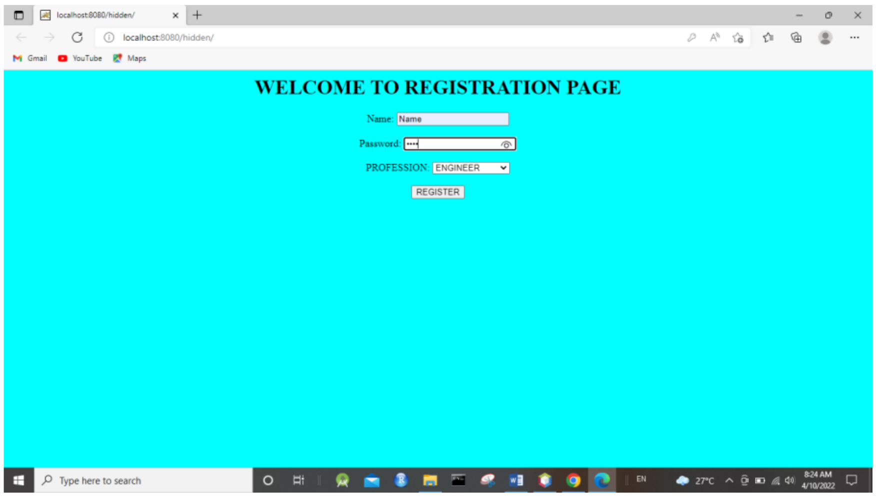
```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class RegistrationServletOne extends HttpServlet
{
public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
{
String name = request.getParameter("name");
String password = request.getParameter("password");
String profession = request.getParameter("profession");
response.setContentType("text/html");
PrintWriter out = response.getWriter();
out.println("<html><body bgcolor = wheat>");
out.println("<center>");
out.println("<h1>COMPLETE THE REGISTRATION</h1>");
out.println("<form action = ./registertwo method = post");
out.println("<input type = hidden name = name value =" + name + ">");
out.println("<input type = hidden name = password value =" + password + ">");
out.println("<input type = hidden name = profession value =" + profession + ">");
out.println("EMAIL ID:<input type =text name = email><br><br>");
out.println("PHONE NO:<input type =text name = cell><br><br>");
out.println("<input type =submit value=registernow>");
out.println("</center>");
out.println("</body></html>");
out.close();
}
}
```

RegistrationServletTwo.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;
public class RegistrationServletTwo extends HttpServlet
{
public void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
throws ServletException, IOException
{
String name = request.getParameter("name");
String password = request.getParameter("password");
String profession = request.getParameter("profession");
String email = request.getParameter("email");
String cell = request.getParameter("cell");
response.setContentType("text/html");
PrintWriter out = response.getWriter();
out.println("<html><body bgcolor = wheat>");
out.println("<center>");
out.println("<h1>REGISTRATION SUCCESSFUL..........</h1>");
out.println("</center>");
out.println("</body></html>");
out.close();
}
```

OUTPUT:



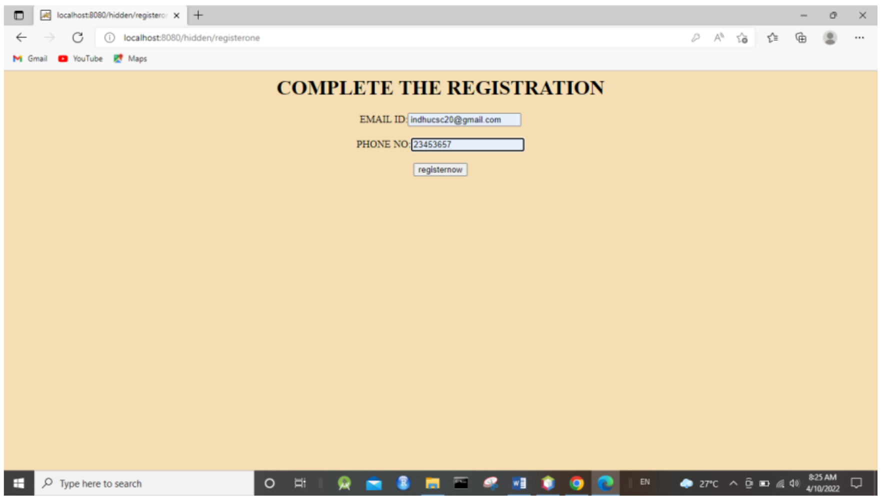
WELCOME TO REGISTRATION PAGE

Name: Name

Password: ••••

PROFESSION ENGINEER

REGISTER

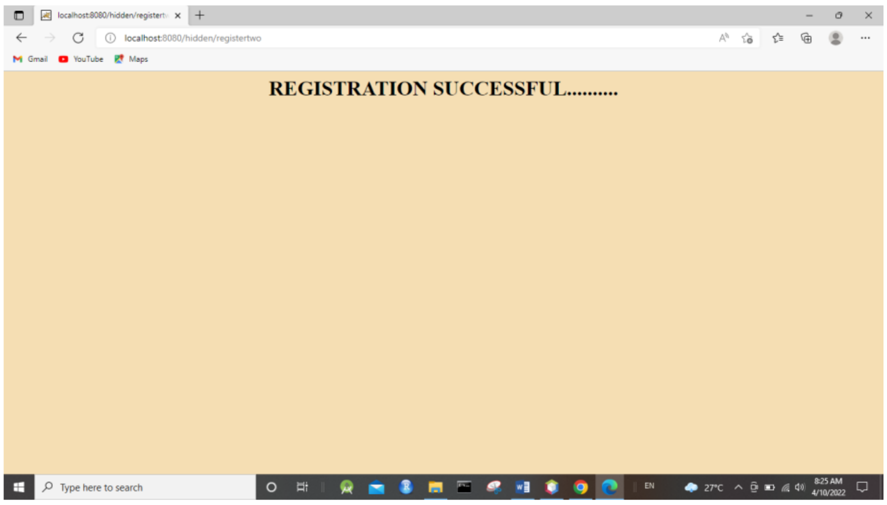


COMPLETE THE REGISTRATION

EMAIL ID: indhucsc20@gmail.com

PHONE NO: 23453657

registernow



REGISTRATION SUCCESSFUL..........

ALGORITHM:

1. Design the database schema for the information retrieval system. Determine thetables, fields, and relationships between the tables.
2. Create a database using MySQL and populate it with sample data.
3. Design the user interface for the search feature. This may involve creating anHTML form where users can enter search terms and submit the form to the server.
4. Write PHP code to handle the user input from the search form. The PHP code should connect to the MySQL database, retrieve the relevant data based on the userinput, and display the results to the user.
5. Implement a ranking algorithm to sort the search results by relevance. This may involve calculating a score based on factors like the frequency of the search terms in thedatabase and the proximity of the search terms to each other.
6. Implement pagination to display search results across multiple pages.
7. Add security measures to prevent SQL injection attacks and other security vulnerabilities.
8. Test the system thoroughly to ensure that it works as expected.
9. Deploy the system to a web server and make it accessible to users.
10. Monitor the system for performance and security issues, and make updates as needed to improve the system over time.

PROGRAM:

1. **Create a MySQL database and table to store the information we want to retrieve:**

```
CREATE DATABASE example_db;
USE example_db;
CREATE TABLE example_table (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(30) NOT NULL,
email VARCHAR(50) NOT NULL,
phone VARCHAR(20) NOT NULL
);
```

2. **Create a PHP script to connect to the database and retrieve the information:**

```
<?php
// Connect to database
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "example_db";
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
// Retrieve information from database
$sql = "SELECT * FROM example_table";
$result = $conn->query($sql);
// Close database connection
$conn->close();
?>
```

3. **Create a web interface using HTML and CSS to display the retrieved information:**

```
<!DOCTYPE html>
<html>
<head>
```

```
<title>Information Retrieval System</title>
<style>
table {
border-collapse: collapse;
width: 100%;
}
th, td {
text-align: left;
padding: 8px;
}
tr:nth-child(even) {
background-color: #f2f2f2;
}
</style>
</head>
<body>
<h1>Information Retrieval System</h1>
<table>
<tr>
<th>ID</th>
<th>Name</th>
<th>Email</th>
<th>Phone</th>
</tr>
<?php
// Output retrieved information in table rows
if ($result->num_rows> 0) {
while($row = $result->fetch_assoc()) {
echo "<tr>";
echo "<td>" . $row["id"] . "</td>";
echo "<td>" . $row["name"] . "</td>";
echo "<td>" . $row["email"] . "</td>";
echo "<td>" . $row["phone"] . "</td>";
echo "</tr>";
}
} else {
echo "0 results";
}
?></table>
</body>
</html>
```

OUTPUT:

```
ID | Name | Email | Phone
-------------------------------------------------------
1 | John Doe | johndoe@example.com | 555-1234
2 | Jane Doe | janedoe@example.com | 555-5678
3 | Bob Smith| bob.smith@example.com | 555-9876
```

ALGORITHM:

1. Start the program.
2. In the Netbeans Ide click File→ New→ WebProject.
3. Give the name of the project.
4. Add the Jsp file by right clicking the project name→ Webroot→ New→ Jsp.
5. Open the web.xml file and update the file by editing the name of the first page of the project.
6. Now create the database in Oracle to store the student details and configure the database connections in the control panel for database connectivity.
7. Save the document.
8. Run the jsp program by, right click the jsp file→ Run as→ Run Configurations…
9. Result will be displayed.

PROGRAM:

//jsps.html

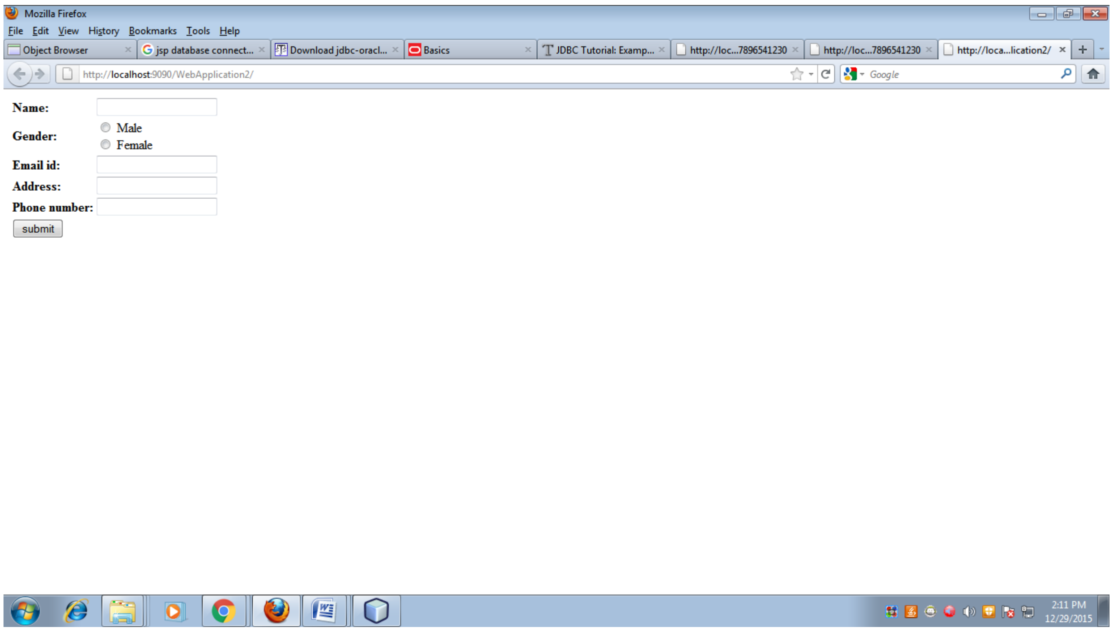
```
<html>
<head>
<title></title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
  <form name="form" action="jsps.jsp" method=”get”>
            <table>
                      <tr>
                                <td><b><font color="black">Name:</b></td>
                                <td><input type="text" name="fname" /></td
                      </tr>
                      <tr>
                                <td><b><font color="black">Gender:</b></td>
                                <td><input type="radio" name="gen" value="male" />
Male<br />
                                          <input type="radio" name="gen" value="female" />
Female</td>
                      </tr>
                      <tr>
                                <td><b><font color="black">Email id:</b></td>
                                <td><input type="text" name="id" /></td
                      </tr>
```

```
                <tr>
                        <td><b><font color="black">Address: </b></td>
                        <td><input type="text" name="ad"></td
                </tr>
                <tr>
                        <td><b><font color="black">Phone number:</b></td>
                        <td><input type="text" name="pn" /></td
                </tr>
                <tr>
                        <td><input type="submit" value="submit"></td>
                </tr>
                </form>
        </table>
        </center>
</body>
</html>
```

//jsps.jsp

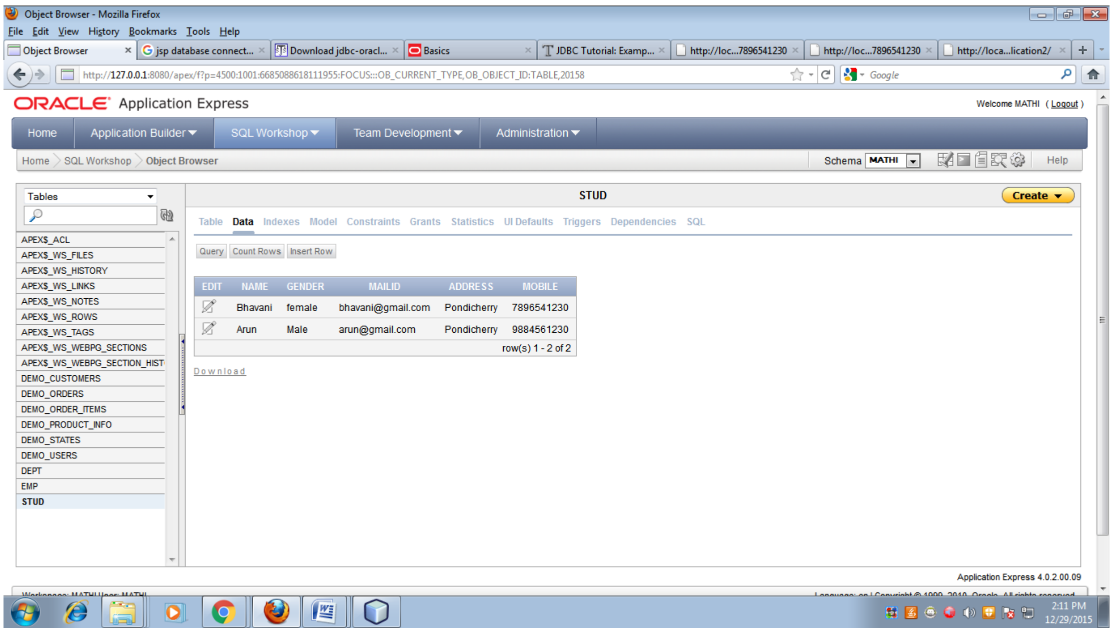
```
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.Connection"%>
<%@page import="java.sql.DriverManager"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<% String fname=request.getParameter("fname");
String gen=request.getParameter("gen");
String id=request.getParameter("id");
String ad=request.getParameter("ad");
String pn=request.getParameter("pn");
try {
Class.forName("oracle.jdbc.driver.OracleDriver");
Connection con=DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","Mathi","Mathi");
Statement stmt=con.createStatement();
String SQL = "INSERT INTO stud
VALUES('"+fname+"','"+gen+"','"+id+"','"+ad+"','"+pn+"')";
stmt.executeUpdate(SQL);
out.println("<br><b> successfully registered</b>");
con.close();
}
catch(Exception e)
{
out.println(e);
}
%>
```

OUTPUT:


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:9090/WebApplication2/

Name:	
Gender:	Male / Female
Email id:	
Address:	
Phone number:	

submit


Object Browser - Mozilla Firefox
ORACLE Application Express
Home Application Builder SQL Workshop Team Development Administration
Home SQL Workshop Object Browser
Schema MATHI
Tables
APEX$_ACL
APEX$_WS_FILES
APEX$_WS_HISTORY
APEX$_WS_LINKS
APEX$_WS_NOTES
APEX$_WS_ROWS
APEX$_WS_TAGS
APEX$_WS_WEBPG_SECTIONS
APEX$_WS_WEBPG_SECTION_HIST
DEMO_CUSTOMERS
DEMO_ORDERS
DEMO_ORDER_ITEMS
DEMO_PRODUCT_INFO
DEMO_STATES
DEMO_USERS
DEPT
EMP
STUD
Table Data Indexes Model Constraints Grants Statistics UI Defaults Triggers Dependencies SQL
Query Count Rows Insert Row
Create
Download

STUD

EDIT	NAME	GENDER	MAILID	ADDRESS	MOBILE
	Bhavani	female	bhavani@gmail.com	Pondicherry	7896541230
	Arun	Male	arun@gmail.com	Pondicherry	9884561230

row(s) 1 - 2 of 2