**MICRO – PROJECT ANNEXURE 2**

Course Title:

Advanced Java Programming

Course Code : 22517

Under Guidance of:

Mrs. Suhasini Shukla

Topic: Design a servlet to accept name from the user and

count the number of characters in the name. If the

number of characters is greater than 7, display the square

of number of characters.



**Group Members**:

|  |  |  |
| --- | --- | --- |
| **Roll Number** | **Enrollment Number** | **Name** |
| 19 | 2200070332 | Manish Naresh Padyar |
| 55 | 2200070380 | Karishma Sanjay Kale |

1. **Rationale**

This project involves creating a servlet that accepts a user's name and counts the number of characters in it, displaying the square of the count if it exceeds seven. It serves as an educational tool to understand Java web development fundamentals, including handling HTTP requests and responses, manipulating strings, and creating user interfaces through HTML forms. By engaging with this project, learners gain practical skills in servlet programming and user interaction, providing a strong foundation for more advanced web applications.

1. **AIMS AND BENEFITS OF MICRO-PROJECT:**

• demonstrate a simple servlet handling user input.

• practice web application development using Java Servlets

• Enhances understanding of server-side processing with Java.

• Provides experience in handling form data and processing user input.

• Introduces simple conditional logic for dynamic web content generation.

1. **COURSE OUTCOMES ADDRESSED (COs):**

• Develop programs using GUI framework (AWT and Swing).

• Handle Events by AWT and Swing components.

• Develop programs to handle event in java programming.

• Develop java programs using network Concept.

• Develop programs using Database.

• Develop programs using Servlet.

1. **PROPOSED METHODOLOGY:**

• Identify the main functionalities: accepting a user’s name, counting the characters, and performing conditional logic if the count exceeds 7.

• Create a simple web form (HTML) for user input.

• Design the servlet to handle HTTP requests (GET/POST).

• Develop the servlet to process the input.

• Use HttpServletRequest to capture the user’s name and HttpServletResponse to send the result back.

• Deploy the servlet on a local or cloud-based server like Apache Tomcat.

1. **Output of the Project:**

**2.1 Code:**

Servlet Code

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet("/CharacterCount")

public class CharacterCountServlet extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

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

int length = name.length();

String result;

if (length > 7) {

result = "The square of the number of characters is: " + (length \* length);

} else {

result = "The number of characters is: " + length;

}

response.setContentType("text/html");

response.getWriter().println("<html><body>");

response.getWriter().println("<h2>" + result + "</h2>");

response.getWriter().println("</body></html>");

}

}

Html Code

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Character Count</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

form {

background: white;

padding: 20px;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0,0,0,0.1);

}

input[type="text"] {

width: 100%;

padding: 10px;

margin-bottom: 10px;

border-radius: 4px;

border: 1px solid #ccc;

}

input[type="submit"] {

background-color: #5cb85c;

color: white;

border: none;

padding: 10px;

border-radius: 4px;

cursor: pointer;

}

</style>

</head>

<body>

<form action="CharacterCount" method="post">

<label for="name">Enter your name:</label>

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

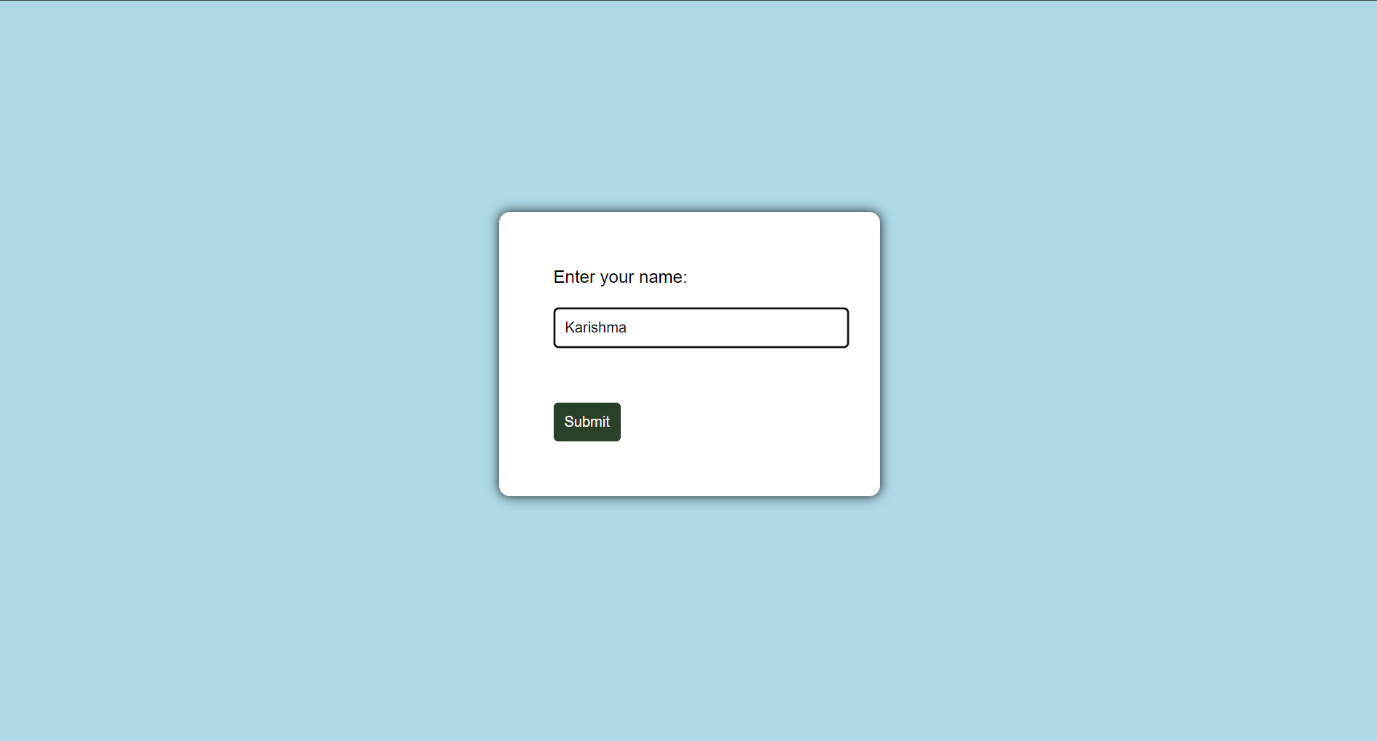
<input type="submit" value="Submit">

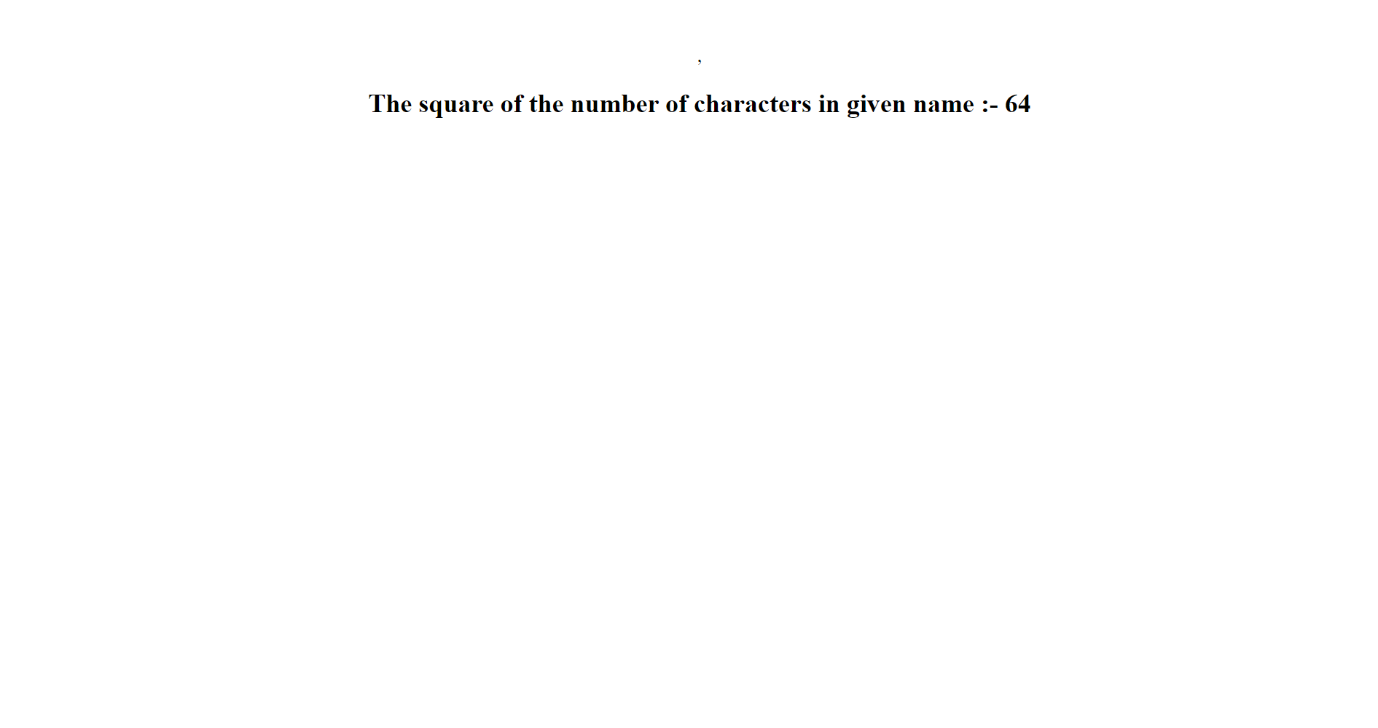
</form>

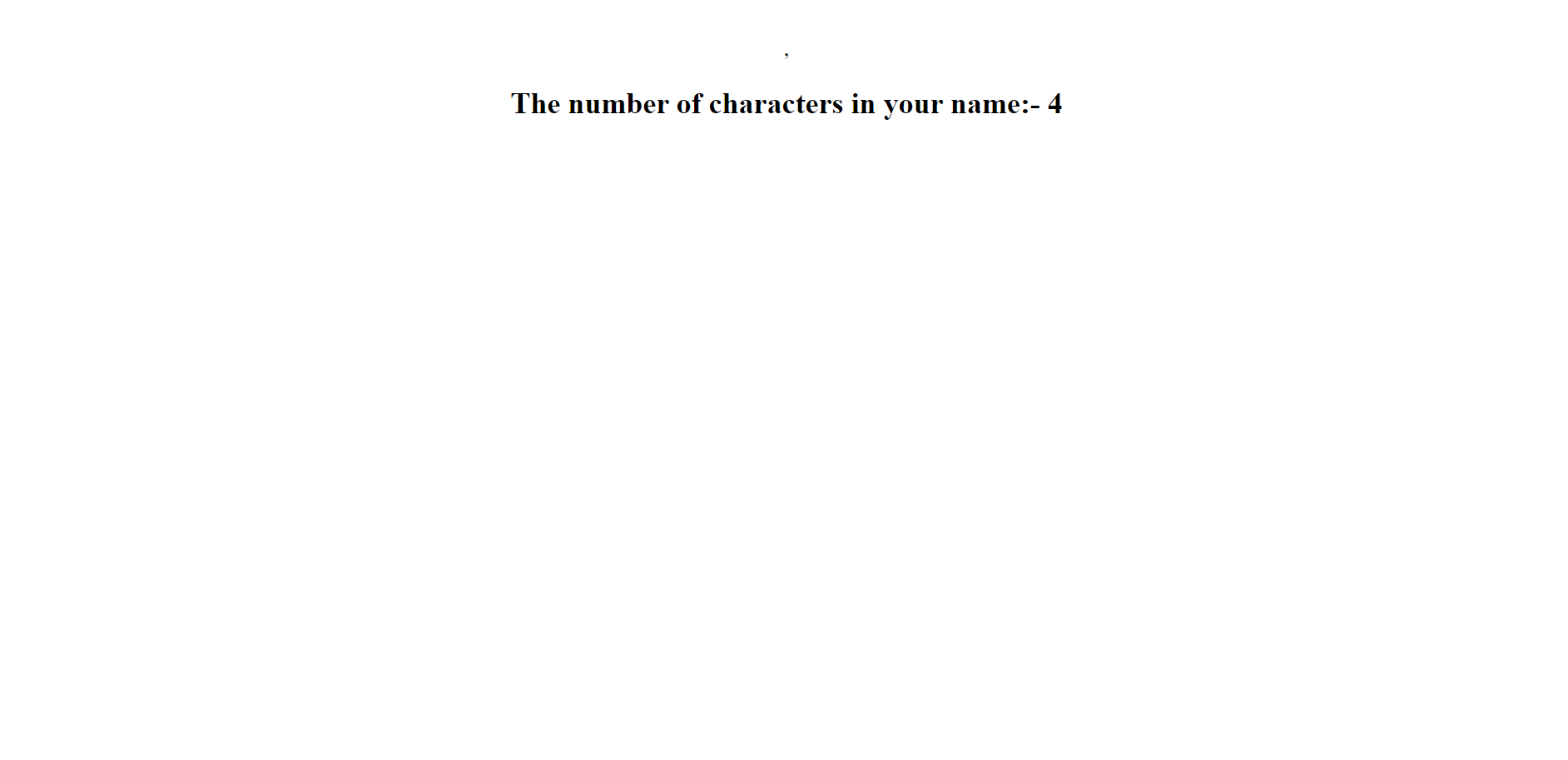
</body>

</html>

**2.2 Output of Project:**

****

****

****

1. **Skill Developed / Learning Outcome of the Microproject:**

* **Java Programming Skills**: Improve proficiency in Java, particularly in the context of web applications.
* **Understanding of Servlets**: Learn the basics of Java Servlets, including lifecycle methods (init, service, and destroy) and how they handle HTTP requests and responses.
* **Web Application Development**: Gain experience in developing dynamic web applications, understanding the interaction between client-side and server-side components.
* **HTML and Form Handling**: Familiarize yourself with HTML forms, including how to collect user input and send it to a servlet.
* **String Manipulation**: Enhance skills in string handling and manipulation, specifically counting characters and performing arithmetic operations.
* **Conditional Logic**: Develop your ability to implement conditional statements to control the flow of the application based on user input.
* **Error Handling**: Learn how to manage exceptions and validate user input, ensuring the application behaves as expected under various conditions.
* **Session Management**: If applicable, understand how to manage user sessions and data across multiple requests.
* **Deployment Skills**: Gain experience in deploying web applications on a servlet container (like Apache Tomcat), understanding configuration and setup.
* **Basic Knowledge of MVC Architecture**: Get introduced to the Model-View-Controller (MVC) design pattern, which is often used in web applications.
* **User Experience Consideration**: Consider how to display results effectively to users, enhancing the overall user experience.

1. **Application of Micro-Project**

* **User Input Validation**: This application can serve as a simple validation tool for user inputs, ensuring that names or other text fields meet specific criteria (e.g., length restrictions).
* **Interactive Web Applications**: It can be integrated into web applications that require user engagement, such as registration forms or feedback systems, enhancing user experience by providing immediate feedback.
* **Data Analytics:** The servlet can be extended to gather statistics on user inputs, such as average name length, which could be useful for demographic analysis in applications.
* **Educational Tools**: This project serves as an educational resource in programming courses, helping students learn about servlets, string manipulation, and web technologies.
* **Gamification Features**: The application could be expanded to include gamified elements, rewarding users for longer names or unique character combinations, encouraging user participation**.**