Input Form (`index.html`)

Create an HTML form with the following fields:

- **Student Name** (Text input)
- **Subject 1 Score** (Number input, range 0–100)
- **Subject 2 Score** (Number input, range 0–100)
- **Subject 3 Score** (Number input, range 0–100)
- Submit button sending a `POST` request to `/calculateGrade`.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Student Grade Portal</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 40px; line-height: 1.6; }
        .form-card { max-width: 400px; padding: 20px; border: 1px solid #ccc;
border-radius: 8px; }
        .form-group { margin-bottom: 15px; }
        label { display: block; margin-bottom: 5px; font-weight: bold; }
        input[type="text"], input[type="number"] { width: 100%; padding: 8px;
box-sizing: border-box; }
        button { background-color: #007bff; color: white; padding: 10px 15px;
border: none; border-radius: 4px; cursor: pointer; }
        button:hover { background-color: #0056b3; }
    </style>
</head>
<body>
    <div class="form-card">
        <h2>Student Grade Calculator</h2>
        <form action="calculateGrade" method="POST">
            <div class="form-group">
                <label for="studentName">Student Name:</label>
                <input type="text" id="studentName" name="studentName" required>
            </div>
            <div class="form-group">
                <label for="score1">Subject 1 Score (0-100):</label>
                <input type="number" id="score1" name="score1" min="0" max="100"
required>
            </div>
            <div class="form-group">
                <label for="score2">Subject 2 Score (0-100):</label>
                <input type="number" id="score2" name="score2" min="0" max="100"
required>
            </div>
            <div class="form-group">
                <label for="score3">Subject 3 Score (0-100):</label>
                <input type="number" id="score3" name="score3" min="0" max="100"
required>
            </div>
            <button type="submit">Calculate Grade</button>
        </form>
    </div>
</body>
</html>
```

Calculation Servlet (`GradeCalculatorServlet.java`)

Create a Servlet mapped to `/calculateGrade` that handles `POST` requests:

1. **Read Request Parameters:** Retrieve student name and score values.
2. **Calculate Results:**
   - **Average Score:** Average = {Score1 + Score2 + Score3}/3
   - **Letter Grade:**
     - Average >= 90: **A**
     - Average >= 80: **B**
     - Average >= 70: **C**
     - Average >= 60: **D**
     - Average >= 60: **F**
3. **Session Storage:** Store the student's name, calculated average, and letter grade inside the current `HttpSession`.
4. **Display Response:** Render an HTML page displaying the student's name, calculated average score (formatted to 2 decimal places), letter grade, and a link/button to view the **Summary Page**

```
package com.university.portal;

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 jakarta.servlet.http.HttpSession;

import java.io.IOException;
import java.io.PrintWriter;

@WebServlet("/calculateGrade")
public class GradeCalculatorServlet extends HttpServlet {

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        try {
            // 1. Retrieve Form Data
            String studentName = request.getParameter("studentName");
            double score1 =
Double.parseDouble(request.getParameter("score1"));
            double score2 =
Double.parseDouble(request.getParameter("score2"));
            double score3 =
Double.parseDouble(request.getParameter("score3"));

            // Server-side Input Validation
            if (score1 < 0 || score1 > 100 || score2 < 0 || score2 > 100 ||
score3 < 0 || score3 > 100) {
                out.println("<html><body><h3 style='color:red;'>Error: All
scores must be between 0 and 100.</h3>");
                out.println("<a href='index.html'>Try
Again</a></body></html>");
                return;
            }
```

```
            // 2. Calculate Average Score & Letter Grade
            double average = (score1 + score2 + score3) / 3.0;
            String grade;
            if (average >= 90) grade = "A";
            else if (average >= 80) grade = "B";
            else if (average >= 70) grade = "C";
            else if (average >= 60) grade = "D";
            else grade = "F";

            // 3. Store Results in HttpSession
            HttpSession session = request.getSession();
            session.setAttribute("studentName", studentName);
            session.setAttribute("averageScore", String.format("%.2f",
average));
            session.setAttribute("grade", grade);

            // 4. Render Response
            out.println("<html><head><title>Calculation
Result</title></head><body>");
            out.println("<h2>Calculation Successful</h2>");
            out.println("<p><strong>Student Name:</strong> " + studentName +
"</p>");
            out.println("<p><strong>Average Score:</strong> " +
String.format("%.2f", average) + "</p>");
            out.println("<p><strong>Letter Grade:</strong> " + grade +
"</p>");
            out.println("<br>");
            out.println("<a href='summary'>View Stored Summary</a> | ");
            out.println("<a href='index.html'>Calculate Another Score</a>");
            out.println("</body></html>");

        } catch (NumberFormatException e) {
            out.println("<html><body><h3 style='color:red;'>Invalid
numerical input.</h3>");
            out.println("<a href='index.html'>Try Again</a></body></html>");
        }
    }
}
```

3. Session Summary Servlet (`SummaryServlet.java`)

Create a second Servlet mapped to `/summary` that handles `GET` requests:

1. **Retrieve Session:** Check if a valid session exists containing student data.
2. **Authorized Access:**
    - If session data **exists**: Render an HTML summary card showing the stored name, average score, and letter grade, along with a "Logout / Start Over" link.
    - If session data **does not exist**: Render an error message ("No active student record found. Please submit scores first.") with a link back to `index.html`.

```
package com.university.portal;

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 jakarta.servlet.http.HttpSession;
```

```
import java.io.IOException;
import java.io.PrintWriter;

@WebServlet("/summary")
public class SummaryServlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        // Retrieve existing session (do not create a new one if it doesn't
exist)
        HttpSession session = request.getSession(false);

        out.println("<html><head><title>Student
Summary</title></head><body>");

        if (session != null && session.getAttribute("studentName") != null)
{
            String studentName = (String)
session.getAttribute("studentName");
            String averageScore = (String)
session.getAttribute("averageScore");
            String grade = (String) session.getAttribute("grade");

            out.println("<div style='border:1px solid #007bff; padding:15px;
width:300px; border-radius:5px;'>");
            out.println("<h3>Student Session Summary</h3>");
            out.println("<p><strong>Name:</strong> " + studentName +
"</p>");
            out.println("<p><strong>Average:</strong> " + averageScore +
"</p>");
            out.println("<p><strong>Grade:</strong> " + grade + "</p>");
            out.println("</div><br>");
            out.println("<a href='logout'>Logout / Clear Session</a>");
        } else {
            out.println("<h3 style='color:red;'>No active student record
found.</h3>");
            out.println("<p>Please submit your scores first.</p>");
            out.println("<a href='index.html'>Go to Registration Form</a>");
        }

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

4. Session Clear Servlet (`LogoutServlet.java`)

Create a Servlet mapped to `/logout` that invalidates the session (`session.invalidate()`) and redirects the user back to `index.html`.

```
package com.university.portal;

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 jakarta.servlet.http.HttpSession;

import java.io.IOException;

@WebServlet("/logout")
public class LogoutServlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {

        HttpSession session = request.getSession(false);
        if (session != null) {
            session.invalidate(); // Destroy session and remove all
attributes
        }

        // Redirect user back to form page
        response.sendRedirect("index.html");
    }
}
```

Program Execution Flow

1. Open `http://localhost:8080/your-app-context/index.html` in your browser.
2. Submit student name and scores. `GradeCalculatorServlet` processes the inputs, computes the result, stores data in the `HttpSession`, and returns the calculation result page.
3. Click **View Stored Summary** to call `SummaryServlet`, which reads stored attributes directly from `HttpSession`.
4. Click **Logout / Clear Session** to trigger `LogoutServlet`, which invalidates the session and redirects back to `index.html`.