**Code:**

const express = require("express");

const app = express();

const PORT = 3000;

function calculateGrade(avg) {

  if (avg >= 90) return "A+";

  if (avg >= 80) return "A";

  if (avg >= 70) return "B";

  if (avg >= 60) return "C";

  return "D";

}

app.use(express.urlencoded({ extended: true }));

app.get("/", (req, res) => {

  res.send(`

        <style>

            body {

                font-family: Arial, sans-serif;

                background-color: #f4f4f9;

                display: flex;

                justify-content: center;

                align-items: center;

                height: 100vh;

                margin: 0;

            }

            form {

                background-color: white;

                padding: 20px;

                border-radius: 5px;

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

            }

            input, button {

                display: block;

                margin: 10px 0;

                padding: 8px;

                width: 100%;

                max-width: 300px;

            }

            button {

                background-color: #5cb85c;

                color: white;

                border: none;

                cursor: pointer;

            }

            button:hover {

                background-color: #4cae4c;

            }

        </style>

        <form action="/student" method="post">

            <h2>Enter Student Marks</h2>

            Name: <input type="text" name="name" required><br>

            Marks 1: <input type="number" name="mark1" required><br>

            Marks 2: <input type="number" name="mark2" required><br>

            Marks 3: <input type="number" name="mark3" required><br>

            Marks 4: <input type="number" name="mark4" required><br>

            Marks 5: <input type="number" name="mark5" required><br>

            <button type="submit">Submit</button>

        </form>

    `);

});

app.post("/student", (req, res) => {

  const { name, mark1, mark2, mark3, mark4, mark5 } = req.body;

  const marksArray = [

    parseFloat(mark1),

    parseFloat(mark2),

    parseFloat(mark3),

    parseFloat(mark4),

    parseFloat(mark5),

  ];

  const totalMarks = marksArray.reduce((a, b) => a + b, 0);

  const avg = totalMarks / marksArray.length;

  const grade = calculateGrade(avg);

  const highest = Math.max(...marksArray);

  const lowest = Math.min(...marksArray);

  res.send(`

        <style>

            body {

                font-family: Arial, sans-serif;

                background-color: #f4f4f9;

                padding: 20px;

                text-align: center;

            }

            h1 {

                color: #5cb85c;

            }

            p {

                font-size: 1.2em;

            }

            a {

                text-decoration: none;

                color: #5cb85c;

                font-weight: bold;

            }

            a:hover {

                text-decoration: underline;

            }

        </style>

        <h1>Results for ${name}</h1>

        <p>Total Marks: ${totalMarks}</p>

        <p>Average: ${avg.toFixed(2)}</p>

        <p>Grade: ${grade}</p>

        <p>Highest Mark: ${highest}</p>

        <p>Lowest Mark: ${lowest}</p>

        <a href="/">Go Back</a>

    `);

});

app.listen(PORT, () => {

  console.log(`Server running on port ${PORT}`);

});

**Explanation**

**Initial Setup:**

* express is required to create the server.
* The app runs on port 3000.
* app.use(express.urlencoded({ extended: true })) allows the app to parse incoming form data (the student name and marks) from a POST request.

**Grading Function:**

* calculateGrade(avg) is a helper function that accepts the average mark and returns the grade based on specific thresholds:
  + A+ for averages ≥ 90
  + A for averages between 80 and 89
  + B for averages between 70 and 79
  + C for averages between 60 and 69
  + D for anything below 60

**HTML Form (GET Request to /):**

* The form is rendered when the user accesses the root URL ("/").
* It contains five input fields to collect marks and one for the student's name. The form submits its data using the POST method to the /student endpoint.
* CSS styles are embedded within the HTML to style the form and page layout.

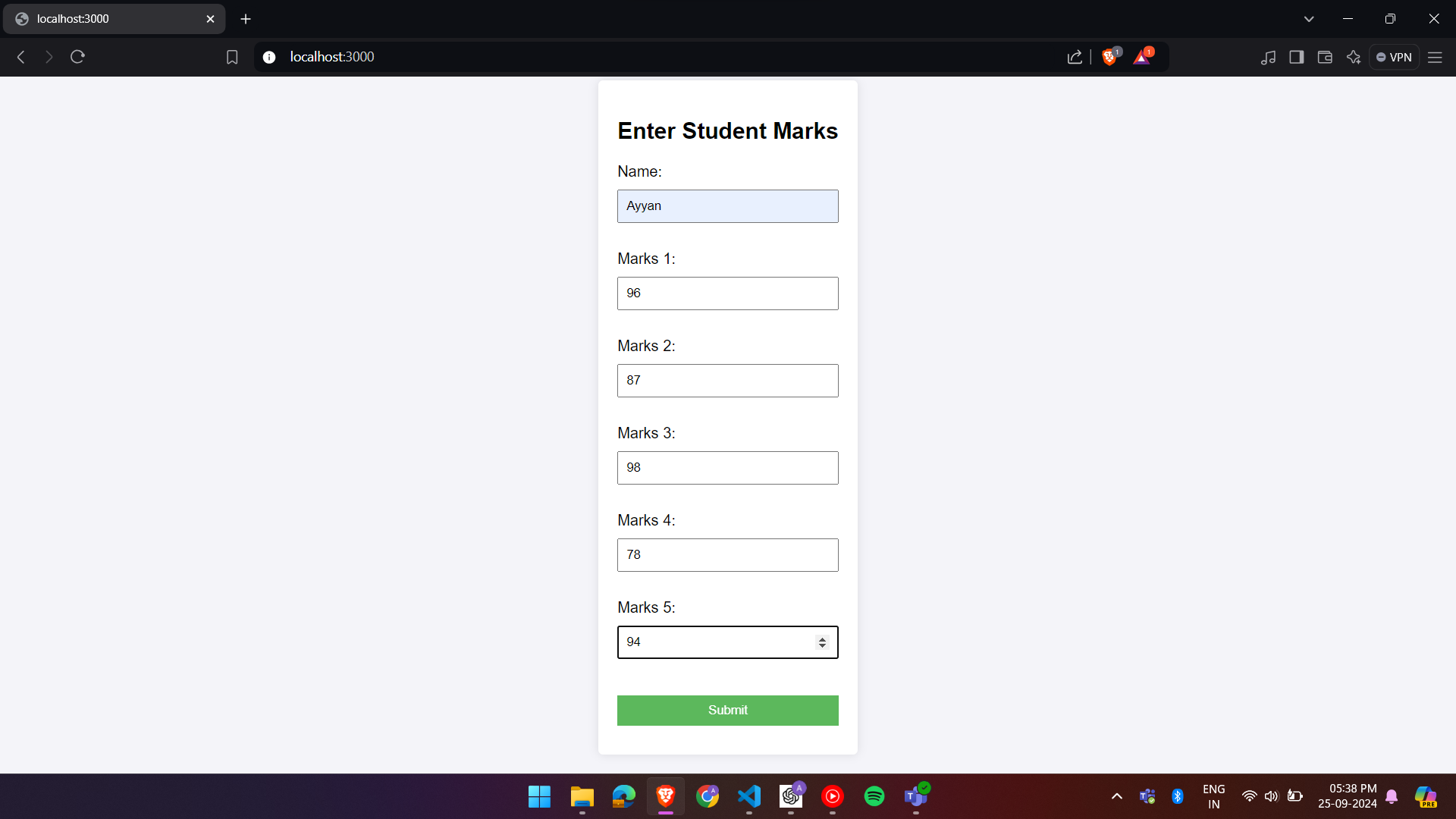
**Form Processing and Results Display (POST Request to /student):**

* The student's name and marks are received from the form data (req.body).
* The marks are converted to numbers and stored in an array (marksArray).
* Calculations:
  + totalMarks: Sum of the five marks.
  + avg: Average of the marks.
  + grade: Grade based on the average, using the calculateGrade function.
  + highest: The highest mark from the array.
  + lowest: The lowest mark from the array.
* The results, including total marks, average, grade, highest mark, and lowest mark, are displayed on a results page.
* This response page includes some simple inline CSS to make it look clean and user-friendly.

**Server Listener:**

* app.listen(PORT) starts the server on the defined port (3000) and logs a message in the console.

**Output:**

A screenshot of a computer

Description automatically generated