**Web Programming Lab (ADL47)**

**Lab Manual**

**Assignment 1: Write an HTML code to create a web page consisting of your resume information such as, photo, objective, technical skills, educational details, hobbies, contact information using header tags, tables, ordered and unordered list, image, table, link, paragraph tags**

**Program:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>My Resume</title>

</head>

<body>

<header>

<h1>Kapil Dongare</h1>

<img src="your-photo.jpg" alt="Profile Photo" width="150">

<p><a href="mailto:kapil@example.com">kapil@example.com</a> | +91-9876543210</p>

<hr>

</header>

<h2>Objective</h2>

<p>

To obtain a challenging position in a reputable organization to expand my learnings, knowledge, and skills.

</p>

<h2>Technical Skills</h2>

<ul>

<li>Programming Languages: HTML, CSS, JavaScript, Python</li>

<li>Web Technologies: React, Node.js, Express</li>

<li>Databases: MySQL, MongoDB</li>

<li>Tools: Git, VS Code, Postman</li>

</ul>

<h2>Educational Details</h2>

<table border="1">

<tr>

<th>Qualification</th>

<th>Institute</th>

<th>Year</th>

<th>Percentage/CGPA</th>

</tr>

<tr>

<td>B.E. in Computer Engineering</td>

<td>XYZ College of Engineering</td>

<td>2023</td>

<td>8.2 CGPA</td>

</tr>

<tr>

<td>HSC</td>

<td>ABC Junior College</td>

<td>2019</td>

<td>85%</td>

</tr>

<tr>

<td>SSC</td>

<td>DEF High School</td>

<td>2017</td>

<td>90%</td>

</tr>

</table>

<h2>Hobbies</h2>

<ol>

<li>Reading Tech Blogs</li>

<li>Photography</li>

<li>Playing Badminton</li>

<li>Traveling</li>

</ol>

<h2>Contact Information</h2>

<p>

<strong>Email:</strong> <a href="mailto:kapil@example.com">kapil@example.com</a><br>

<strong>Phone:</strong> +91-9876543210<br>

<strong>LinkedIn:</strong> <a href="https://www.linkedin.com/in/yourprofile" target="\_blank">linkedin.com/in/yourprofile</a><br>

<strong>Address:</strong> 123, Tech Street, Pune, Maharashtra, India

</p>

</body>

</html>

**Output**:

#### **ABC\_Name**

## **📷 *(Your photo shown here — 150px wide)***

## **📧 ABC@example.com | 📞 +91-9876543210**

### **Objective**

To obtain a challenging position in a reputable organization to expand my learnings, knowledge, and skills.

### **Technical Skills**

* Programming Languages: HTML, CSS, JavaScript, Python
* Web Technologies: React, Node.js, Express
* Databases: MySQL, MongoDB
* Tools: Git, VS Code, Postman

### **Educational Details**

A table with 4 columns:

| **Qualification** | **Institute** | **Year** | **Percentage/CGPA** |
| --- | --- | --- | --- |
| B.E. in Computer Engg. | XYZ College of Engg. | 2023 | 8.2 CGPA |
| HSC | ABC Junior College | 2019 | 85% |
| SSC | DEF High School | 2017 | 90% |

### **Hobbies**

1. Reading Tech Blogs
2. Photography
3. Playing Badminton
4. Traveling

### **Contact Information**

* **Email:** kapil@example.com
* **Phone:** +91-9876543210
* **LinkedIn:** [linkedin.com/in/yourprofile](https://www.linkedin.com/in/yourprofile)
* **Address:** 123, Tech Street, Pune, Maharashtra, India

**Assignment 2: Create a web page for college technical events. Generate student registration form for the same inclusive of student name, branch, USN, event they are participating in. Add hyperlinks for home page, event details page, registration page. Add event photos/ college photos, header and footer**

**Program:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>College Tech Fest 2025</title>

</head>

<body>

<!-- Header Section -->

<header>

<h1>Welcome to TechNova 2025 - College Technical Festival</h1>

<nav>

<a href="index.html">Home</a> |

<a href="event-details.html">Event Details</a> |

<a href="registration.html">Register Now</a>

</nav>

<hr>

</header>

<!-- Main Content -->

<h2>Student Registration Form</h2>

<form action="submit-form.html" method="post">

<label for="name">Student Name:</label><br>

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

<label for="branch">Branch:</label><br>

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

<label for="usn">USN:</label><br>

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

<label for="event">Select Event:</label><br>

<select id="event" name="event" required>

<option value="">--Select an Event--</option>

<option value="Coding Challenge">Coding Challenge</option>

<option value="Hackathon">Hackathon</option>

<option value="Robotics">Robotics</option>

<option value="Quiz">Tech Quiz</option>

</select><br><br>

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

</form>

<hr>

<!-- Event and College Photos -->

<h2>Event Highlights</h2>

<img src="event1.jpg" alt="Tech Event 1" width="300">

<img src="event2.jpg" alt="Tech Event 2" width="300">

<img src="college.jpg" alt="College Campus" width="300">

<!-- Footer Section -->

<footer>

<hr>

<p>&copy; 2025 TechNova, ABC College of Engineering</p>

<p>Contact us at <a href="mailto:technova@college.edu">technova@college.edu</a></p>

</footer>

</body>

</html>

**Output:**

## **Welcome to TechNova 2025 - College Technical Festival**

## **Home | Event Details | Register Now**

### **Student Registration Form**

**Student Name:** [\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_]

**Branch:** [\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_]

**USN:** [\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_]

**Select Event:** [ Coding Challenge ▼ ]

[Register] (Submit Button)

### **Event Highlights**

📷 event1.jpg (Tech Event 1 image - 300px wide)  
 📷 event2.jpg (Tech Event 2 image - 300px wide)  
 📷 college.jpg (College Campus photo - 300px wide)

© 2025 TechNova, ABC College of Engineering  
 📧 Contact us at technova@college.edu

**Assignment 3: Develop and demonstrate the usage of inline, internal and external style sheet using CSS**

**Program:**

Asgn3.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>CSS Style Demo</title>

<!-- Internal Style -->

<style>

body {

font-family: Arial, sans-serif;

background-color: #f3f3f3;

}

h1 {

color: navy;

text-align: center;

}

.internal-style {

color: green;

font-weight: bold;

}

</style>

<!-- External Style Sheet -->

<link rel="stylesheet" href="styles.css">

</head>

<body>

<h1>Student Profile</h1>

<!-- Inline style -->

<p style="color: red; font-size: 18px;">This paragraph uses an inline style (red text).</p>

<!-- Internal style -->

<p class="internal-style">This paragraph uses internal styling (green text).</p>

<!-- External style -->

<p class="external-style">This paragraph uses external styling (blue background, white text).</p>

</body>

</html>

Style1.css

.external-style {

background-color: blue;

color: white;

padding: 10px;

border-radius: 5px;

}

**Assignment 4:**

**Create a web page with the following characteristics using BOX Model**

* 1. **h1's have 1px red solid borders, background color #D18C1D, and 10px of space between the content and the border (padding)**
  2. **List items have 15px extra space around them (margin) and background color #C0A9DB**
  3. **Paragraphs are contained in 600px by 400px boxes with 2px black dotted borders and background color #D1D631**

Program:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Box Model Demo</title>

<style>

/\* h1 styles \*/

h1 {

border: 1px solid red;

background-color: #D18C1D;

padding: 10px;

}

/\* list item styles \*/

li {

margin: 15px;

background-color: #C0A9DB;

}

/\* paragraph box \*/

.box-paragraph {

width: 600px;

height: 400px;

border: 2px dotted black;

background-color: #D1D631;

padding: 10px;

overflow: auto; /\* ensures scroll if content overflows \*/

}

</style>

</head>

<body>

<h1>Welcome to Box Model Demo</h1>

<ul>

<li>HTML & CSS</li>

<li>JavaScript</li>

<li>Web Development</li>

</ul>

<div class="box-paragraph">

<p>

This is a paragraph inside a 600x400 pixel box. It demonstrates the use of the CSS box model, including width, height, borders, padding, and background color. The border is 2px, dotted, and black. The background is a light yellow-green. If this paragraph gets too long, the overflow will be scrollable due to the `overflow: auto` property.

</p>

<p>

You can place more text here to test how the content behaves inside the box. Adjust height and width to experiment further with the box model.

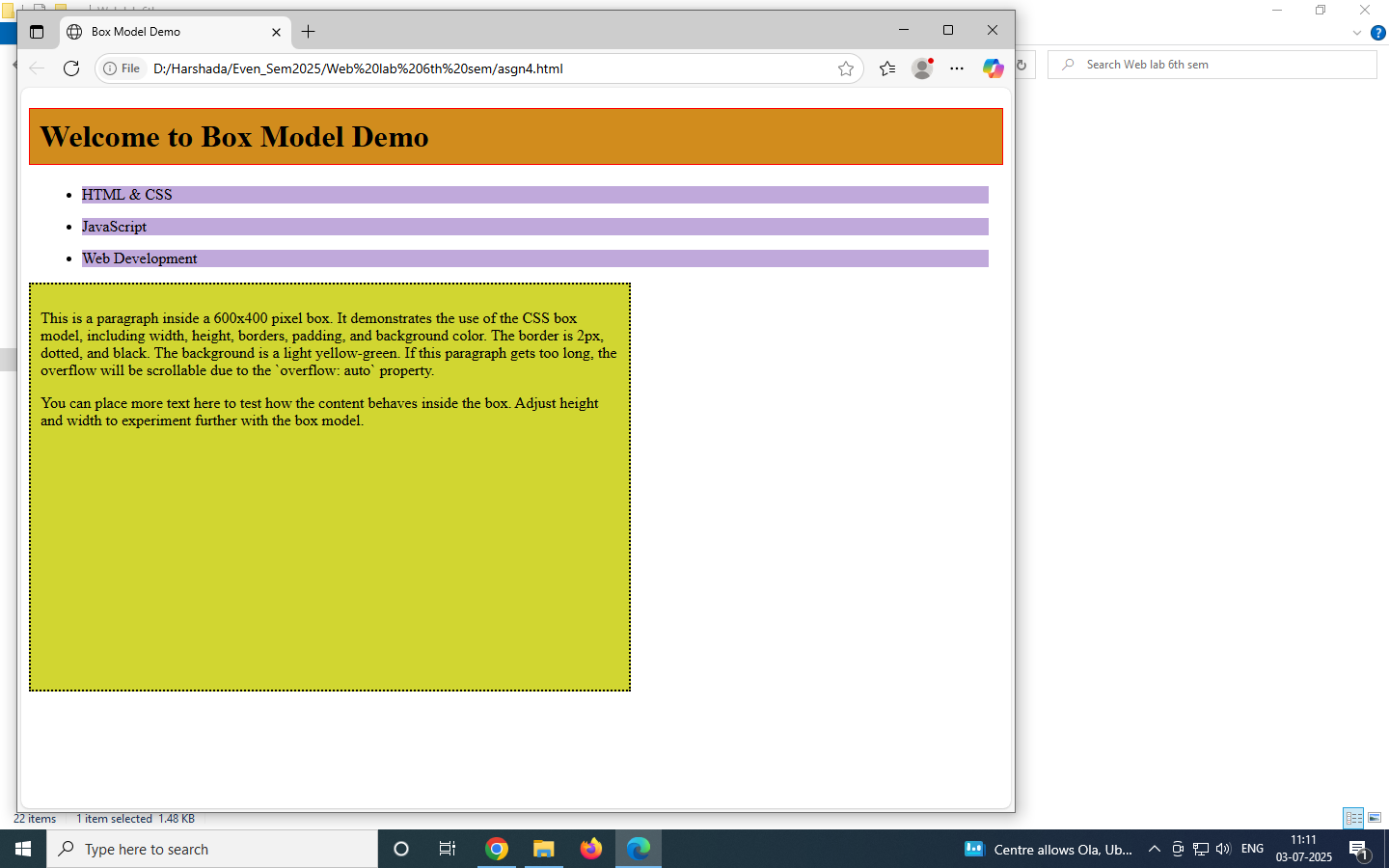
</p>

</div>

</body>

</html>

**Output:**



**Assignment 5: Write a code to build a simple calculator using HTML and CSS to handle basic arithmetic operations.**

**Program:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Simple Calculator</title>

<style>

body {

background: #f0f4f8;

font-family: Arial, sans-serif;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

.calculator {

background-color: #ffffff;

padding: 20px;

border-radius: 16px;

box-shadow: 0 4px 10px rgba(0,0,0,0.1);

width: 260px;

}

.display {

width: 100%;

height: 50px;

margin-bottom: 15px;

font-size: 22px;

text-align: right;

padding-right: 10px;

border: 1px solid #ccc;

border-radius: 8px;

}

.buttons {

display: grid;

grid-template-columns: repeat(4, 1fr);

gap: 10px;

}

button {

padding: 15px;

font-size: 18px;

border: none;

border-radius: 8px;

background-color: #e1e5ea;

cursor: pointer;

transition: background-color 0.3s;

}

button:hover {

background-color: #d0d5db;

}

.operator {

background-color: #ffd54f;

}

.equal {

background-color: #4caf50;

color: white;

}

.clear {

background-color: #f44336;

color: white;

}

</style>

</head>

<body>

<div class="calculator">

<input type="text" class="display" id="display" disabled>

<div class="buttons">

<button class="clear" onclick="clearDisplay()">C</button>

<button onclick="appendValue('/')">/</button>

<button onclick="appendValue('\*')">\*</button>

<button onclick="appendValue('-')">-</button>

<button onclick="appendValue('7')">7</button>

<button onclick="appendValue('8')">8</button>

<button onclick="appendValue('9')">9</button>

<button onclick="appendValue('+')">+</button>

<button onclick="appendValue('4')">4</button>

<button onclick="appendValue('5')">5</button>

<button onclick="appendValue('6')">6</button>

<button class="equal" onclick="calculate()">=</button>

<button onclick="appendValue('1')">1</button>

<button onclick="appendValue('2')">2</button>

<button onclick="appendValue('3')">3</button>

<button onclick="appendValue('0')">0</button>

</div>

</div>

<script>

function appendValue(value) {

document.getElementById("display").value += value;

}

function clearDisplay() {

document.getElementById("display").value = "";

}

function calculate() {

try {

let result = eval(document.getElementById("display").value);

document.getElementById("display").value = result;

} catch (e) {

alert("Invalid expression");

}

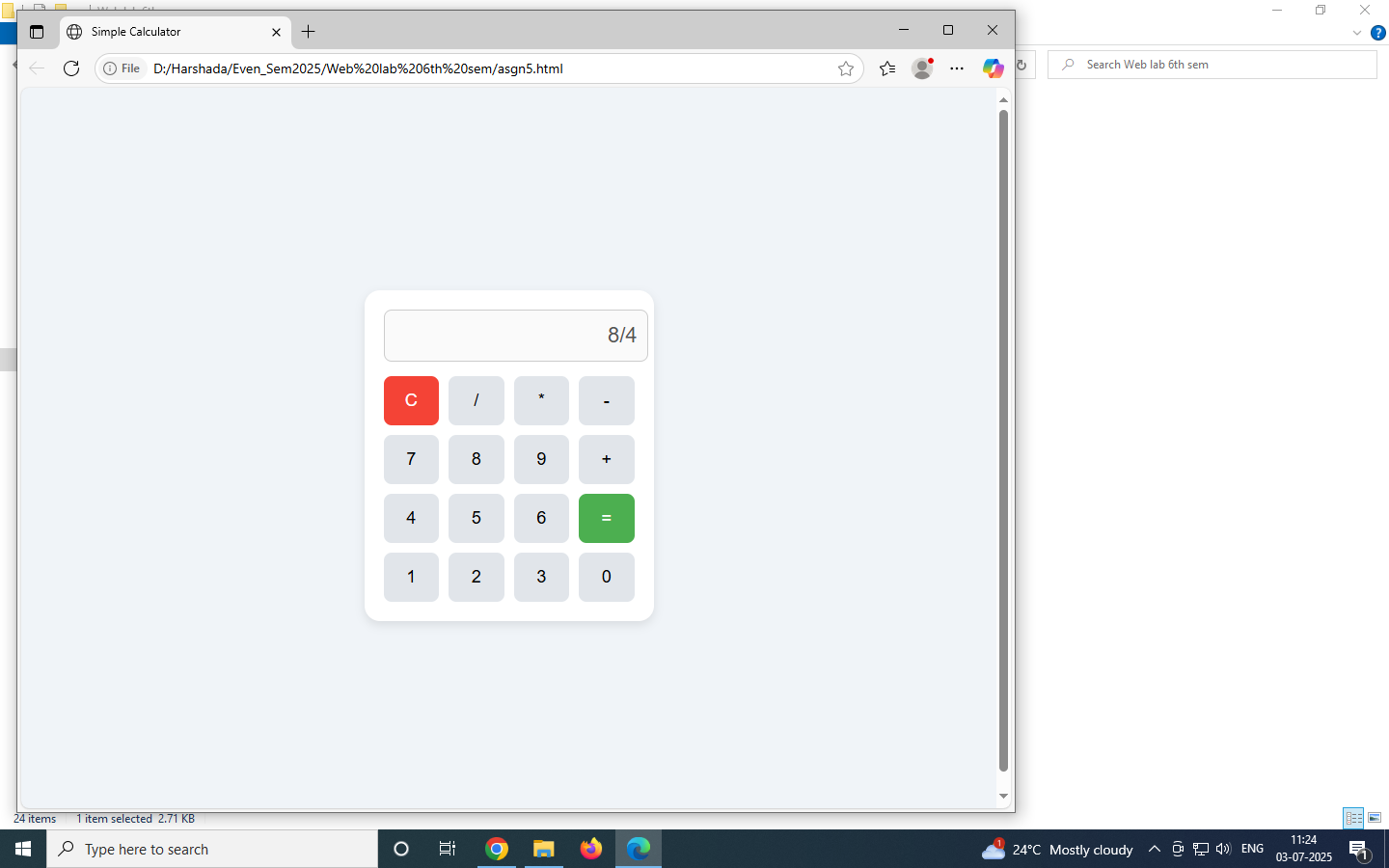
}

</script>

</body>

</html>

**Output:**



**Assignment 6: Design a Student Form using HTML5 which has following fields**

**a) Name : Required must be characters**

**b) Email : Validation placeholder: please enter valid email address**

**c) Phone : accept numbers in the following format (080-555-5555)**

**d) Semester : For the range 1 to 8**

**e) Branch :Data list**

**f) Website :Required pattern of the form-http://**

**Program:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Student Registration Form</title>

</head>

<body>

<h1>Student Registration Form</h1>

<form>

<!-- Name -->

<label for="name">Name: \*</label><br>

<input type="text" id="name" name="name" required pattern="[A-Za-z\s]+" placeholder="Enter your name"><br><br>

<!-- Email -->

<label for="email">Email: \*</label><br>

<input type="email" id="email" name="email" required placeholder="Please enter valid email address"><br><br>

<!-- Phone -->

<label for="phone">Phone (Format: 080-555-5555):</label><br>

<input type="tel" id="phone" name="phone" pattern="\d{3}-\d{3}-\d{4}" placeholder="080-555-5555"><br><br>

<!-- Semester -->

<label for="semester">Semester (1 to 8):</label><br>

<input type="number" id="semester" name="semester" min="1" max="8"><br><br>

<!-- Branch -->

<label for="branch">Branch:</label><br>

<input list="branches" id="branch" name="branch" placeholder="Select your branch">

<datalist id="branches">

<option value="Computer Science">

<option value="Electronics">

<option value="Mechanical">

<option value="Civil">

<option value="Information Technology">

</datalist><br><br>

<!-- Website -->

<label for="website">Website (must start with http://): \*</label><br>

<input type="url" id="website" name="website" required pattern="http://.+" placeholder="http://yourwebsite.com"><br><br>

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

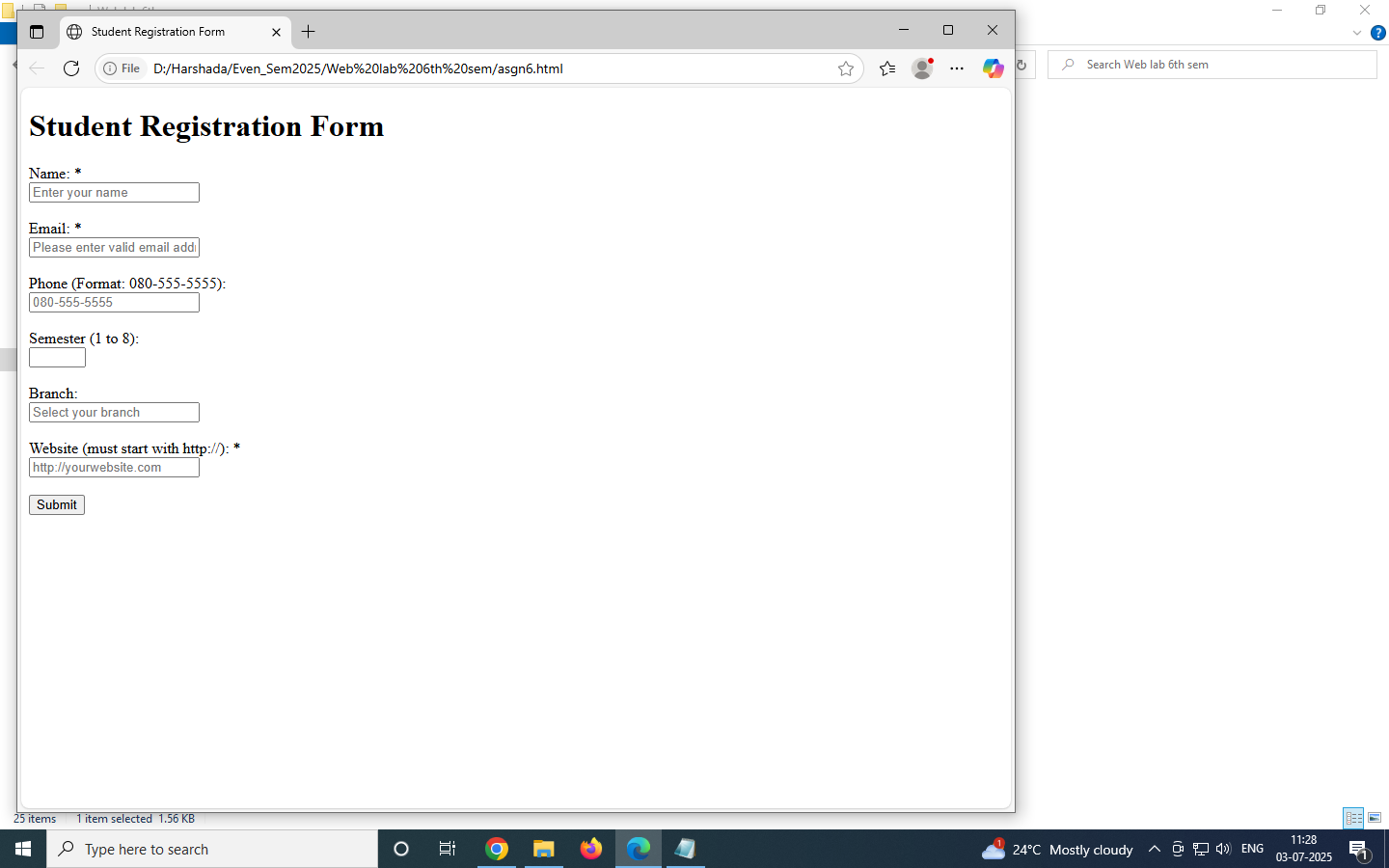
</form>

</body>

</html>

| **Field** | **Feature Used** | **Description** |
| --- | --- | --- |
| **Name** | required, pattern="[A-Za-z\s]+" | Ensures only letters and spaces are accepted. |
| **Email** | type="email", required, placeholder | Validates email format automatically. |
| **Phone** | pattern="\d{3}-\d{3}-\d{4}" | Matches specific phone format like 080-555-5555. |
| **Semester** | type="number", min="1", max="8" | Limits entry to numbers from 1 to 8 only. |
| **Branch** | datalist | Provides a drop-down with predefined options. |
| **Website** | type="url", pattern="http://.+" | Ensures input starts with http://. |

**Output:**



**Assignment 7: Write a java script program to implement Stack and Queue using modules**

**Program:**

**Stack-queue Program**

**Steps to execute this program:**

Open your project folder (stack-queue) in VS Code.

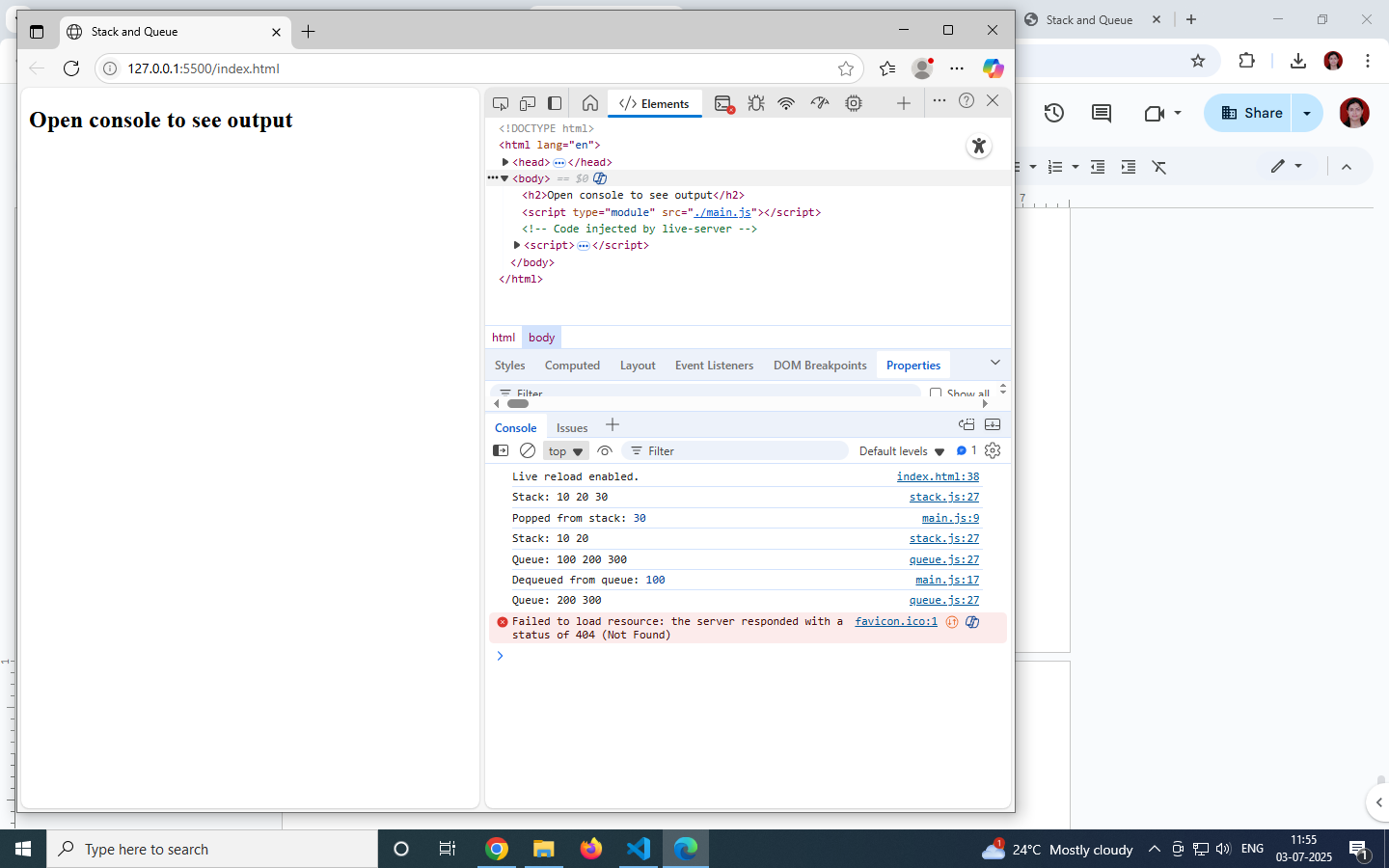
Go to **Extensions** → Search and install **Live Server**.

Right-click index.html → Click **“Open with Live Server”**.

A new tab will open in your browser:  
 http://127.0.0.1:5500/index.html

Press **F12 → Console tab** to view the output.

**Output:**



**Assignment 8: Write a java script program to convert month number to month name using closures.**

* 1. **If the user enters a number less than 1 or greater than 12 or a non-number, have the function write "Bad Number" in the monthName field.**
  2. **If the user enters a decimal between 1 and 12 (inclusive), strip the decimal portion of the number.**

**Program:**

<!DOCTYPE html>

<html>

<head>

<title>Month Converter</title>

</head>

<body>

<h2>Enter Month Number (1 to 12):</h2>

<input type="text" id="monthInput" />

<button onclick="showMonthName()">Get Month Name</button>

<p id="monthName"></p>

<script>

// Closure function to convert number to month name

function monthConverter() {

const months = [

"January", "February", "March", "April", "May", "June",

"July", "August", "September", "October", "November", "December"

];

return function(monthNumber) {

let num = Number(monthNumber);

if (isNaN(num) || num < 1 || num > 12) {

return "Bad Number";

}

num = Math.floor(num); // Strip decimal part

return months[num - 1];

};

}

// Create closure instance

const getMonthName = monthConverter();

// Function to be called when button is clicked

function showMonthName() {

const userInput = document.getElementById("monthInput").value;

const result = getMonthName(userInput);

document.getElementById("monthName").textContent = result;

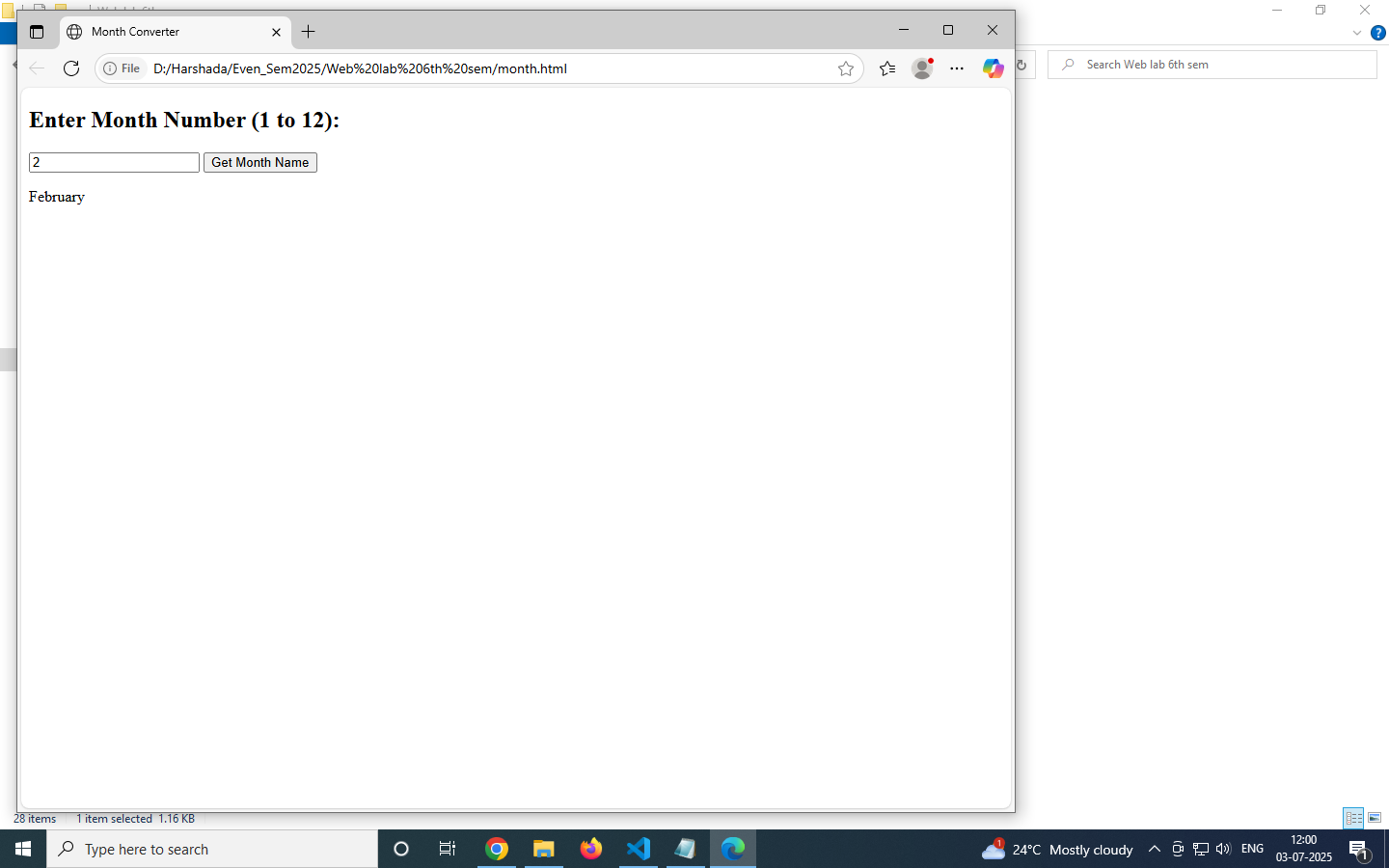
}

</script>

</body>

</html>

**Output:**



**Assignment 9: Write a java script function named pluralize that:**

* 1. **takes 2 arguments, a noun and a number.**
  2. **returns the number and pluralized form, like "5 cats" or "1 dog".**
  3. **Make it handle a few collective nouns like "sheep" and "geese".**

**Program:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Pluralize Nouns</title>

</head>

<body>

<h1>Pluralizer</h1>

<label for="noun">Enter a noun:</label>

<input type="text" id="noun" placeholder="e.g. cat">

<label for="number">Enter a number:</label>

<input type="number" id="number" placeholder="e.g. 2">

<button onclick="showPlural()">Pluralize</button>

<p id="result"></p>

<script>

function pluralize(noun, number) {

// List of irregular/plural-exception nouns

const irregularNouns = {

sheep: 'sheep',

goose: 'geese',

person: 'people',

mouse: 'mice',

child: 'children',

deer: 'deer',

fish: 'fish'

};

let pluralForm;

// If number is 1, use singular form

if (number === 1) {

pluralForm = noun;

}

// If it's in the irregular noun list, use the mapped plural

else if (irregularNouns[noun]) {

pluralForm = irregularNouns[noun];

}

// If the noun ends in 'y' and not a vowel before it (e.g., city -> cities)

else if (noun.endsWith('y') && !/[aeiou]y$/i.test(noun)) {

pluralForm = noun.slice(0, -1) + 'ies';

}

// For nouns ending in s, x, z, ch, sh (e.g., box -> boxes)

else if (noun.endsWith('s') || noun.endsWith('x') || noun.endsWith('z') ||

noun.endsWith('ch') || noun.endsWith('sh')) {

pluralForm = noun + 'es';

}

// Default rule: add 's'

else {

pluralForm = noun + 's';

}

return `${number} ${pluralForm}`;

}

function showPlural() {

const noun = document.getElementById('noun').value.trim().toLowerCase();

const number = parseInt(document.getElementById('number').value);

if (!noun || isNaN(number)) {

document.getElementById('result').textContent = 'Please enter both a noun and a valid number.';

return;

}

const result = pluralize(noun, number);

document.getElementById('result').textContent = result;

}

</script>

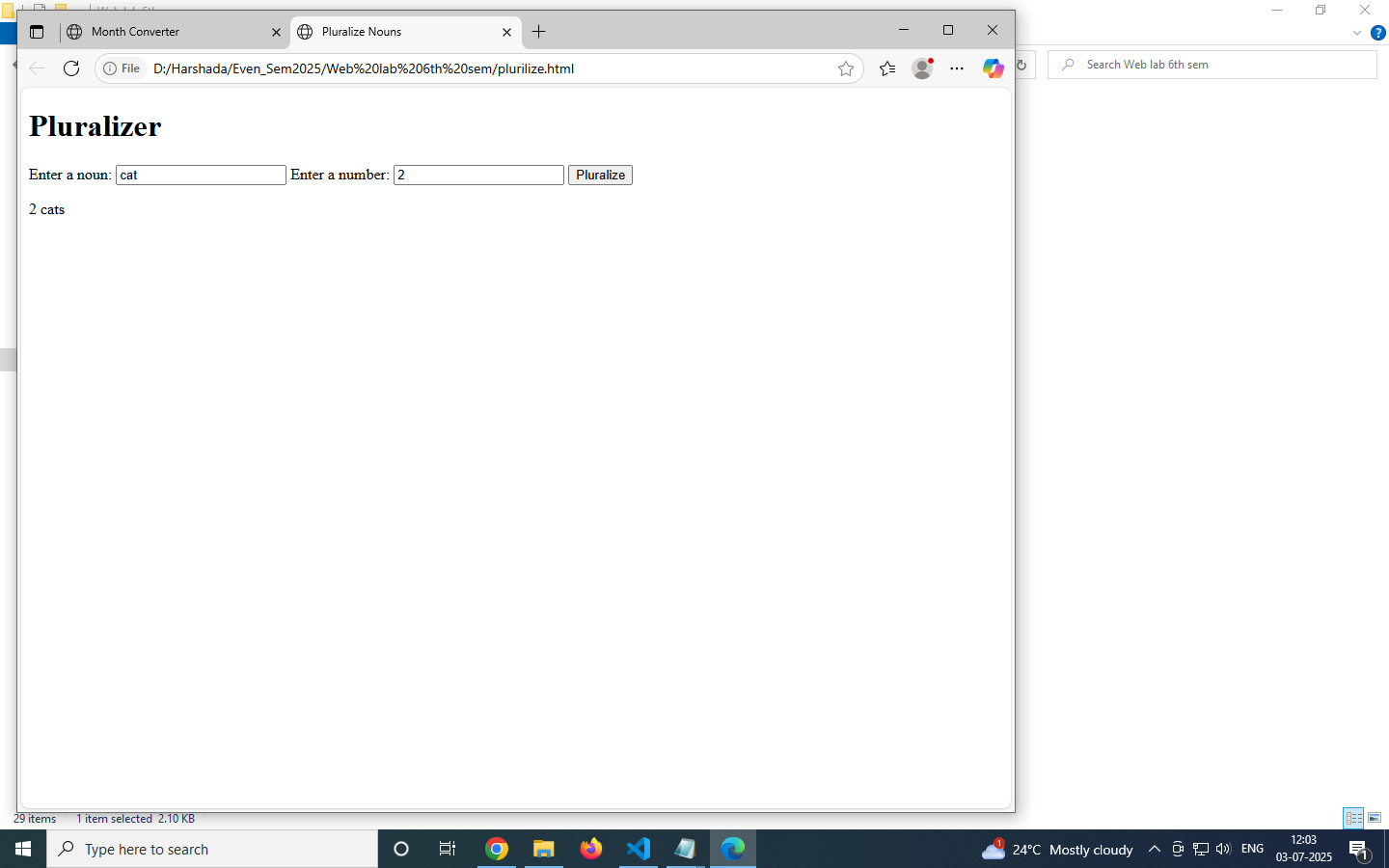
</body>

</html>

**JavaScript Explanation**

1. **Function pluralize(noun, number)**
   * This function handles the logic of converting a singular noun to its plural form based on the number.
   * It checks:
     + If the number is 1, the noun remains singular.
     + If the noun is in the irregularNouns list (like "sheep" or "goose"), it uses the specified plural form.
     + If the noun ends in 'y' and doesn't follow a vowel (e.g., "city" → "cities"), it replaces 'y' with 'ies'.
     + If the noun ends in common plural-suffix patterns (e.g., "box" → "boxes"), it adds 'es'.
     + Otherwise, it just adds 's' (e.g., "dog" → "dogs").
2. **Function showPlural()**
   * This function is called when the button is clicked.
   * It reads user input from the HTML fields, sanitizes it, and validates it.
   * It then calls pluralize() and displays the result inside the <p id="result">.
3. **DOM Manipulation**
   * document.getElementById('...') is used to access HTML elements and update content dynamically.
   * This makes it interactive: the result updates based on user input without needing to reload the page.

**Output:**



**Assignment 10: Using node.js Express and Mongo to implement ‘FinalYears’ database which accepts ‘USN’,’Name’ and ‘Company\_name’ (by campus selection) as fields and store it in a database. Display the list of students who are selected for 'Infosys'**

**Program:**

Finalyearproject program

### **1. Create a New Folder**

Open VS Code and create a new folder, e.g., FinalYearsApp.

### **2. Open Terminal in VS Code**

Use **Terminal → New Terminal** or press Ctrl + `.

### **3. Initialize Node.js**

npm init -y

4. Install Required Packages

npm install express mongoose body-parser dotenv

## **STEP 4: Run the Application**

### **📌 Start MongoDB**

● If installed locally, run:

Mongod

**Run the App from Terminal in VS Code**

node app.js

OUTPUT:

MongoDB connected

Server running at http://localhost:3000

## **STEP 5: Open Browser and type**

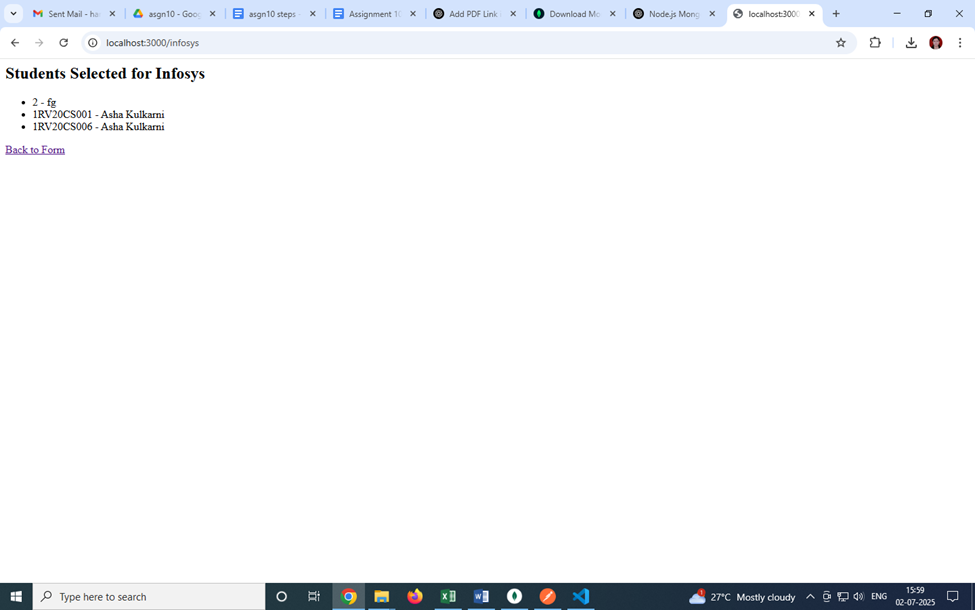
<http://localhost:3000/form.html>

Enter the details

Submit the details

Click “View infosys selection”

**Output:**



**Assignment 11: Using Node.js Express and Mongo, implement a program to accept USN, Name, Subject\_code, CIE marks and store the information in a database and display students whose CIE<20.**

**Program:**

Student cie app program

Steps to execute a program

#### **Install MongoDB**

Ensure MongoDB is installed and running locally on default port (mongodb://localhost:27017).

#### **Open Terminal and Navigate to the Project Folder**

cd path/to/student-cie-app

1. **Install Dependencies**

npm install

1. **Start server**

npm start

You will see

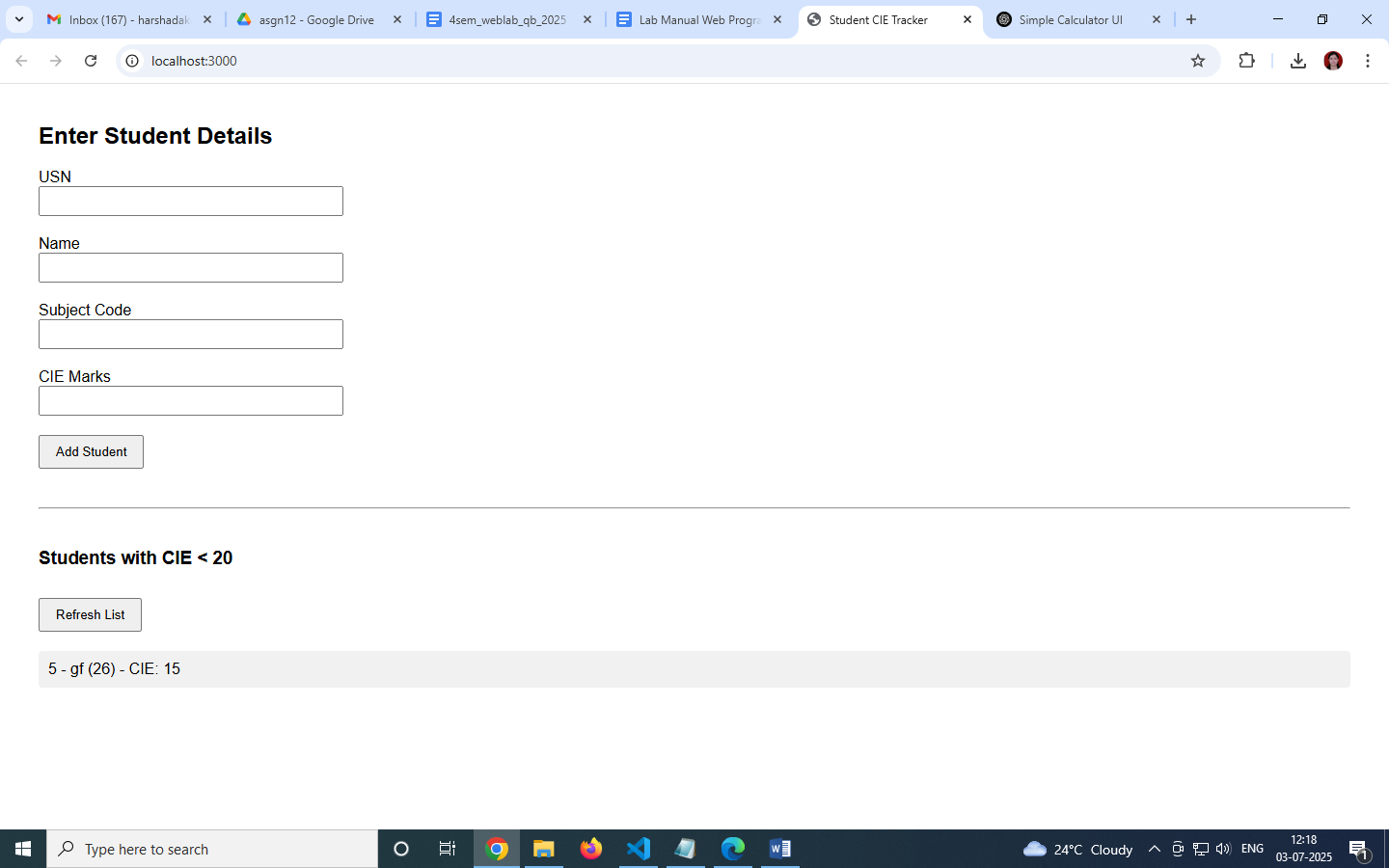
Server running at http://localhost:3000

Connected to MongoDB

#### **Open in Browser**

Go to:<http://localhost:3000>

**Output:**



**Assignment 12:Write a Node.js program using Express framework to display different branch information offered in an Engineering College with different background color and fonts (Note: Use Routing, Min: 3 branches)**

**Program:**

Engg branches app program

Steps to execute

#### **Open Terminal**

#### Navigate to the project folder:

cd path/to/engg-branches-app

1. **Install express**

npm init -y

npm install express

1. **Run the app**

node server.js

#### **View in Browser**

Go to:<http://localhost:3000>

Click on:

* **CSE** – Blue-themed
* **ECE** – Orange-themed
* **Mechanical** – Purple-themed

**Output:**

