Code & Conquer: Coding Marathon

Day 1

Course: CSCI 165 / CSCI 101

Challenge Points: 40 Total Points

Project: "Sam's Clinic" Appointment page

This booklet includes the general information about the objectives, then the information about the project and lastly some tips to make things work for this project. You don't have to finish this project completely but you have to score the highest points. So be wise and choose the stuff with most points to do first! You only have 2 hours.

Coding Marathon Challenge

Course: CSCI 165 / CSCI 101
Challenge Points: 40 Total Points
Project: "Sam's Clinic" Appointment page

Objective:
Sam, a doctor, wants a custom webpage for his clinic, "Sam's Clinic," to streamline his online appointment booking process. Currently, he uses a Google Form linked to Google Sheets for appointment bookings, which he wants to keep as his main database. The new webpage should allow patients to book appointments directly through the site, while keeping all booking data in the existing Google Sheets.

Important Note:
Dr. Sam has an established system that automatically emails patients with the earliest available appointment slot based on his schedule. **Patients cannot select an appointment date**—the system will always book them for the next available slot. This feature is intentional, as Dr. Sam prefers a structured schedule that maximizes his availability.

Requirements:

1. **Appointment Form Integration**
    - Implement a custom appointment form on the website for patients to book appointments. This form should not redirect to the original Google Form.
    - Ensure the form data is sent to the same Google Sheets where the Google Form currently saves data.
    - **Hint:** Inspect the Google Form's network requests to find out how it sends data, and replicate this process on your custom form using JavaScript to send data directly to the Google Form's API endpoint.
2. **Frontend Implementation**
    - Use only HTML, CSS, and JavaScript for the website's structure and functionality. Frameworks like Bootstrap are allowed, but using them will not earn extra points.
3. **Design & Visual Appeal**
    - Make the site visually appealing by including animations, a cohesive color scheme, and a user-friendly layout.
4. **Responsiveness**
    - Ensure the webpage is fully responsive, providing a seamless experience across various devices (desktop, tablet, and mobile).

Evaluation Criteria:

Criteria	Points	Description
Functionality & Requirements	10	Website meets all requirements, with properly aligned and formatted content, and an effective understanding of the instructions.
Responsiveness	10	Webpage is responsive and functions well on different screen sizes (desktop, tablet, mobile).
Form Integration	15	Custom appointment form successfully sends data to the existing Google Sheets, maintaining current functionality without issues.
Visual Design	5	The webpage has a cohesive color scheme, suitable animations, and an overall appealing layout.

Submission Guidelines:

- Complete your project by the end of the coding marathon.
- Submit all files (HTML, CSS, JS) in a zipped folder.

Good luck!

Tips for Making Your Website More Appealing

Here are some useful techniques and code snippets to make your website more visually appealing, interactive, and efficient. This guide covers single-page website navigation, form data handling, button-click functionality, and making POST requests.

---

1. Capturing Form Data in JavaScript Variables

To capture form data in JavaScript, assign id attributes to your form inputs and then access the values using document.getElementById.

HTML Form:

```
<form id="appointmentForm">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name" />

  <label for="email">Email:</label>
  <input type="email" id="email" name="email" />

  <button type="button" onclick="submitForm()">Submit</button>
</form>
```

JavaScript Code:

```
function submitForm() {
  const name = document.getElementById('name').value;
  const email = document.getElementById('email').value;

  console.log("Name:", name);
  console.log("Email:", email);
  // Now, you can use `name` and `email` variables to send data or perform
other actions
}
```

---

2. Running a Function on Button Click

To trigger a function on a button click, use the onclick attribute on your button. This is useful for form submissions or any action you want to perform when a button is clicked.

Example:

```
<button onclick="showMessage()">Click Me</button>

<script>
  function showMessage() {
    alert("Button clicked!");
  }
</script>
```

In the example above, clicking the button will run the showMessage function, displaying an alert.

---

3. Replicating a POST Request from JavaScript

To send form data directly to Google Sheets, you can use JavaScript's fetch API to make a POST request. First, inspect the Google Form's network requests (using Developer Tools) to find the URL and parameters it uses, and replicate it in your code.

JavaScript Code:

```
function submitAppointment() {
  const name = document.getElementById('name').value;
  const email = document.getElementById('email').value;

  // Google Form POST URL (found by inspecting network requests)
  const googleFormURL =
"https://docs.google.com/forms/d/e/YOUR_GOOGLE_FORM_ID/formResponse";

  const formData = new URLSearchParams();
  formData.append("entry.YOUR_NAME_FIELD_ID", name);
  formData.append("entry.YOUR_EMAIL_FIELD_ID", email);

  fetch(googleFormURL, {
    method: 'POST',
    body: formData,
    headers: { 'Content-Type': 'application/x-www-form-urlencoded' },
  })
  .then(response => {
    if (response.ok) {
      alert("Appointment submitted successfully!");
    } else {
      alert("Error submitting appointment.");
    }
  })
  .catch(error => console.error("Error:", error));
}
```

Replace YOUR_GOOGLE_FORM_ID, YOUR_NAME_FIELD_ID, and YOUR_EMAIL_FIELD_ID with the actual values found by inspecting the Google Form's requests. This code sends the same request to Google Forms as a regular submission would.

With these techniques, your website will not only meet the functional requirements but will also provide a smooth, professional, and user-friendly experience for Dr. Sam's patients. Happy coding!