Final Project Development

Web Development Essentials - Session 14

Session Overview

Learning Goals for Today:

- Allocate in-class time for final project development.
- Provide hands-on support and guidance for troubleshooting.
- Work on building key features of your project (e.g., forms, interactivity, responsiveness).

Sample Project Overview – Event Website

Sample Project: Event Website

- **Objective**: Build a website for an event that includes:
    - Event details (e.g., date, time, location).
    - Registration forms for attendees.
    - A gallery showcasing past events or highlights.

Features:

- **Homepage**: Displays the event name, date, location, and a call to action (CTA) to register.
- **Registration Form**: Allows users to sign up for the event.
- **Gallery**: Showcases event images or media using a dynamic JavaScript gallery.
- **Responsive Design**: Ensures the website works on mobile, tablet, and desktop.

Project Breakdown – Key Sections

Main Sections for the Event Website:

1. **Homepage**:
   - Hero banner with event details and a call-to-action button (e.g., "Register Now").
2. **Event Details**:
   - Description of the event, schedule, speakers, etc.
3. **Registration Form**:
   - Form fields for user input (name, email, number of attendees, etc.).
   - JavaScript validation for user input.
4. **Gallery**:
   - Image carousel or grid layout for event photos.

Technologies Used:

- **HTML** for structure.
- **CSS** for layout and design.
- **JavaScript** for dynamic interactivity (gallery, form handling).

Step 1 – Building the Homepage

Key Elements of the Homepage:

- **Header**: Event name and logo.
- **Hero Section**: Event date, location, and registration CTA button.
- **Navigation**: Links to other sections like "Event Details" "Gallery", and "Contact".

```html
<header>
  <h1>Annual Tech Conference 2024</h1>
  <p>Date: March 15, 2024 | Location: New York City</p>
  <a href="#register" class="cta-button">Register Now</a>
</header>
```

```css
header {
  text-align: center;
  background: url('event-banner.jpg') no-repeat center;
  background-size: cover;
  padding: 50px 0;
  color: white;
}

.cta-button {
  padding: 10px 20px;
  background-color: #ff5733;
  color: white;
  text-decoration: none;
  border-radius: 5px;
}
```

Step 2 – Creating the Registration Form

Form Structure:

- Include fields for the attendee's name, email, and number of attendees.
- Use JavaScript to validate the form (e.g., required fields, email format).

```html
<form id="register" action="submit-form.php" method="post">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name" required>

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

  <label for="attendees">Number of Attendees:</label>
  <input type="number" id="attendees" name="attendees" min="1" required>

  <button type="submit">Register</button>
</form>
```

```javascript
document.querySelector("form").addEventListener("submit", function(event) {
  let nameInput = document.getElementById("name");
  let emailInput = document.getElementById("email");

  if (nameInput.value === "" || emailInput.value === "") {
    alert("Please fill out all required fields.");
    event.preventDefault();
  }
});
```

Step 3 – Building the Gallery

Dynamic Image Gallery:

- Create a grid or carousel for event photos using JavaScript.

```html
<section id="gallery">
  <h2>Event Highlights</h2>
  <div class="gallery-grid">
    <img src="event1.jpg" alt="Event Photo 1">
    <img src="event2.jpg" alt="Event Photo 2">
    <img src="event3.jpg" alt="Event Photo 3">
  </div>
</section>
```

```css
.gallery-grid {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  gap: 10px;
}

.gallery-grid img {
  width: 100%;
  height: auto;
  border-radius: 8px;
}
```

```javascript
let currentImageIndex = 0;
let images = document.querySelectorAll('.gallery-grid img');

setInterval(() => {
  images[currentImageIndex].style.display = 'none';
  currentImageIndex = (currentImageIndex + 1) % images.length;
  images[currentImageIndex].style.display = 'block';
}, 3000);
```

Responsive Design & Media Queries

Making Your Website Responsive:

- Use **media queries** to adjust the layout for different screen sizes (desktop, tablet, mobile).

```
@media (max-width: 768px) {
  .gallery-grid {
    grid-template-columns: 1fr;
  }
}
```

Tips for Mobile-Friendly Design:

- Keep navigation simple.
- Ensure buttons and form fields are large enough to tap easily.
- Avoid large images that slow down mobile performance.

Troubleshooting Tips

Common Issues:

- **Form not submitting**: Ensure you're using the `event.preventDefault()` method if the form submission is being handled by JavaScript.
- **CSS not applying**: Check that the correct file paths are used in your `link` tags and ensure there are no typos in class or ID selectors.
- **JavaScript not working**: Use `console.log()` to debug and check the browser's developer tools for errors.

Summary

What We Worked On Today:

- Began hands-on development of the final project.
- Worked through challenges and troubleshooting with instructor guidance.

Next Session:

- Continue project development.
- Prepare for the final project presentation.

Questions?

Q&A Session

- Any questions before we wrap up?

Thank You & See You in the Next Class!