1. **<u>Why CSS? Inline vs. Internal vs. External Scenario:</u>**

   The designer wants you to experiment with different ways to apply styles.

   Objective:

   Understand various CSS inclusion methods and their impact.

   Task:

   • Apply an inline style to make one heading red.

   • Use an embedded to define body background.

   • Link an external stylesheet styles.css and move all reusable styles there.

   • Add comments in your CSS to label each section (/* Header styles */)

   Inline Style

   ```
   <h1 style="color: red;">Local Events</h1>
   ```

   Internal Embedded

   ```
   <style>
     body {
       background-color: #f4f4f4;
     }
   </style>
   ```

   External

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

   Header to label each section

   ```
   /* Header styles */
   h1, h2 {
     font-family: Arial, sans-serif;
   }
   ```

2. **<u>CSS Syntax and Comments Scenario:</u>**

   You've joined a team and need to understand and maintain a large stylesheet.

   Objective: Write clean, readable CSS with proper structure and comments.

   Task:

   • Create a section in styles.css with formatted rules and consistent indentation.

   • Add descriptive comments above selectors.

   • Example:

   ```
   /* Style for main CTA button */
   .cta-button {
   background-color: #007BFF;
   color: white;
   }
   ```

```
/* Style for main CTA button */
.cta-button {
  background-color: #007BFF;
  color: white;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  font-size: 16px;
}
```

3. <u>**Selectors Playground Scenario:**</u>

   You need to style various elements based on IDs, classes, and element types.
   Objective: Master different selector types.
   Task:
   - **Use:**
     - **o Universal selector * to reset margin/padding**
     - **o Element selector to style all**
     - **o ID selector #mainHeader for the banner**
     - **o Class selector .eventCard for event containers**
     - **o Grouping selector for h3, p to style together**

```
/* Reset all margins and padding */
* {
  margin: 0;
  padding: 0;
}

/* Style all <h2> elements */
h2 {
  color: navy;
}

/* ID selector */
#mainHeader {
  background-color: lightblue;
}

/* Class selector */
.eventCard {
  border: 1px solid #ccc;
  padding: 10px;
  margin: 10px;
}

/* Group selector */
h3, p {
  font-family: 'Open Sans', sans-serif;
  line-height: 1.5;
}
```

4. **<u>Color & Background Styling Scenario:</u>**

   You’re theming the portal based on a city council’s branding.
   Objective: Apply consistent colors and background visuals.
   Task:
   • Use HEX and RGBA for setting text and background colors
   • Add a background image to the body with fallback color
   • Apply gradients to section headers using background: linear-gradient(...)

```
body {
  background: url('background.jpg') no-repeat center center fixed;
  background-color: #e0e0e0; /* fallback */
}

/* HEX and RGBA */
h1 {
  color: #333333;
}

.eventCard {
  background-color: rgba(255, 255, 255, 0.9);
}

/* Gradient */
section.header {
  background: linear-gradient(to right, #0066cc, #66ccff);
  color: white;
}
```

5. **<u>Typography: Fonts and Text Scenario:</u>**

   The marketing team wants more appealing fonts and better readability.

   Objective: Enhance textual appearance using CSS properties.

   Task:

   • Use @import or to include a Google Font

   • Set font-family, font-size, font-style, font-weight in different sections

   • Use text-align, text-transform, letter-spacing, line-height on descriptions

   In HTML Head

```
<link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap"
rel="stylesheet">
```

   In CSS

```
body {
  font-family: 'Roboto', sans-serif;
  font-size: 16px;
}

.description {
  text-align: justify;
  text-transform: capitalize;
  letter-spacing: 0.5px;
  line-height: 1.6;
}
```

6. **Link and List Styling Scenario:**

   The default blue links and bullet lists don’t match the design.

   Objective: Customize links and lists. Task:

   • Style links with :link, :hover, :active, and :visited pseudo-classes

   • Use list-style-type, list-style-position, and remove bullets from nav menus

   • Add padding and margin to list items for spacing

```
/* Link styles */
a:link {
  color: #007BFF;
}
a:visited {
  color: purple;
}
a:hover {
  color: #0056b3;
}
```

```
a:active {
  color: red;
}

/* List styles */
nav ul {
  list-style-type: none;
  padding: 0;
}

nav li {
  margin: 10px;
  list-style-position: inside;
}
```

7. **Table Styling Scenario: The events admin table needs a cleaner look.**
   Objective: Format tables using CSS.
   Task:
   • Style table, th, and td with borders, padding, and background color
   • Add zebra striping to rows using nth-child(even)
   • Use border-collapse: collapse and text-align: center

```
table, th, td {
  border: 1px solid #ccc;
  border-collapse: collapse;
}

th, td {
  padding: 10px;
  text-align: center;
}

/* Zebra striping */
tr:nth-child(even) {
  background-color: #f9f9f9;
}
```

8. **<u>Box Model & Layout Control Scenario:</u>**
   Sections are cramped and need spacing.
   Objective: Control element spacing with margin, padding, border, and outline.

Task:

- **Use developer tools to inspect and tweak box model properties**
- **Add border, padding, and margin to .eventCard**
- **Add outline to highlight selected fields in a form**
- **Compare visibility: hidden vs. display: none**

```
.eventCard {
  border: 1px solid #ddd;
  margin: 15px;
  padding: 20px;
}

input:focus {
  outline: 2px solid #007BFF;
}

/* Visibility comparison */
.hidden-element {
  visibility: hidden; /* occupies space */
}

.removed-element {
  display: none; /* doesn't occupy space */
}
```

9. **<u>Multiple Columns in Text Scenario:</u>**

The community bulletin needs to be displayed like a newspaper.

Objective: Use CSS3 multi-column layout.

Task:

- **Create a news article section and apply:**

  column-count: 2;
  column-gap: 30px;
  column-rule: 1px solid gray;

```
.news-article {
  column-count: 2;
  column-gap: 30px;
  column-rule: 1px solid gray;
}
```

10. **<u>Responsive Web Design with Media Queries Scenario:</u>**

Users will access the portal on phones, tablets, and desktops.
Objective: Apply media queries for responsiveness.
Task:

- **Add a media query for screens smaller than 768px**
- **Stack navigation links vertically instead of horizontally**
- **Reduce image sizes and font sizes**
- **Use %, vw, vh for flexible layouts**
- **Bonus: Try Flexbox or Grid for responsive layouts**

```
@media screen and (max-width: 768px) {
  nav ul {
    display: block;
  }

  nav li {
    display: block;
    margin: 10px 0;
  }

  img {
    width: 100%;
    height: auto;
  }

  body {
    font-size: 14px;
  }
}

/* Bonus: Flexbox example */
.container {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-between;
}
```