Student 213110003

DETAILED EXPLANATORY FILE

Responsive one-page website using HTM5L&CSS and a little JavaScript

One-Page Stockholm Attraction and Tourist Website

Overview:

The website is designed as a one-page portal that introduces users to the charm of Stockholm, Sweden, highlighting its rich history, vibrant culture, and various tourist attractions. This site serves as a comprehensive guide for tourists, providing sections for attractions, activities, events, dining, and accommodations. It emphasizes the city's unique character and offers practical information for travelers planning a visit.

Sections Breakdown:

Header and Navigation:

1. The website starts with a navigation bar that includes links to different sections such as **Attractions**, **Activities**, **Events**, **Eat & Drink**, and **Contact**
2. The header features the city's name (Stockholm) along with a tagline "The Nordic Venice" to emphasize the city's distinct and picturesque nature.
3. The responsive hamburger menu is included to ensure the site is easily navigable on mobile devices.

4. Second navigaion menu

Intro Section:

1. This section immediately introduces visitors to Stockholm with a welcoming headline and a brief description inviting users to explore the city's beauty, history, and culture.
2. The design is minimal, ensuring users are drawn to the rest of the content.

History Section:

1. This section provides a brief yet informative overview of Stockholm’s rich history. Key historical points include its origins in 1252, its significance during the Hanseatic League, and its transformation under King Gustav Vasa.
2. A video embedded from YouTube offers additional insights into Stockholm's historical journey, enhancing the visitor's understanding.

Climate Section:

1. The climate section introduces the four distinct seasons of Stockholm, showcasing how the city changes throughout the year. This is particularly useful for tourists trying to decide the best time to visit.
2. The content is complemented by a slideshow showing images of Stockholm across the different seasons, from vibrant spring to snowy winter, encouraging users to experience the city in various climates.

What to Do Section:

1. This part offers suggestions on must-see attractions in Stockholm, from historical sites like **Gamla Stan** (Old Town) to museums such as the **Vasa Ship Museum**.
2. Visitors can enjoy a video highlighting top attractions and activities around the city, from boat tours to cultural landmarks, reinforcing the city's appeal.

Where to Stay Section:

1. The accommodation section provides suggestions for places to stay, categorized by location and type of experience.

Responsive Design:

1. The website has been designed to be mobile-friendly, with sections laid out in a single-column format for easy reading on smaller screens.
2. Navigation is optimized with a hamburger menu on mobile and a fixed navigation bar on desktop.

External Resources:

1. Links to useful external resources like booking platforms for accommodations are provided to enhance the user's experience and facilitate easier trip planning.

Design & Aesthetics:

- The visual design follows a minimalist approach, with large images that showcase Stockholm’s beauty. The use of modern fonts like **Montserrat**, **Lora**, and **Poppins** enhances readability and overall aesthetic appeal.
- Icons are used creatively (such as phone and email icons) to provide easy access to contact details.
- The site's layout is responsive, with well-structured content blocks and images that adjust to different screen sizes, ensuring a smooth user experience across devices.

Technical Considerations:

- **HTML5** was used for semantic web markup, ensuring the content is properly structured and accessible.
- The **CSS** file controls the visual layout, with animations such as the slideshow and hover effects on the navigation links.
- Javascript
- **YouTube embedding** is used to present informative videos on the history of Stockholm and popular activities, keeping the page engaging and informative.

HTML5

1. DOCTYPE Declaration

<!DOCTYPE html>

- This tells the browser that the document type is HTML5. It's important for proper rendering and compatibility with modern browsers.

2. Head Section

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=yes">
  <title>Stockholm City</title>
  <!-- External links to stylesheets and fonts -->
</head>
```

- <meta charset="UTF-8">: Specifies the character encoding as UTF-8, which supports a wide range of characters and symbols.
- <meta name="viewport" content="...">: Ensures the page is responsive on mobile devices by controlling the viewport's size and scale.
- <title>: Sets the title of the webpage, which appears in the browser tab.
- External resources are linked here, such as external fonts (Google Fonts) and icons (Font Awesome), and the styles.css file for styling.

3. Body Section

```
<body id="home">
```

- The main content of the webpage resides here, with an ID home set for potential use in styling or JavaScript functions.

Header Section

```
<header>
  <nav>
    <!-- Left: Stockholm City | Right: Hamburger Menu -->
    <div class="header-content">
      <div class="city-name">
        <span class="city-main">Stockholm</span>
        <span class="city-sub">The Nordic Venecia</span>
      </div>
      <div id="hamburger-menu" class="hamburger">
        <span></span>
        <span></span>
        <span></span>
      </div>
    </div>
    <!-- Left Navigation -->
    <div class="left-nav">
      <ul>
        <!-- List items for navigation links -->
      </ul>
    </div>
    <!-- Right Navigation -->
    <div class="right-nav">
      <!-- Contact info with clickable phone and email -->
    </div>
  </nav>
</header>
```

- <header>: Contains the website’s top navigation bar. It includes the city name, hamburger menu for mobile view, and both left and right navigation areas.
- <nav>: Defines a block of navigation links. These are grouped into left-nav (main menu) and right-nav (contact info).

    - The hamburger menu (a common icon used in mobile layouts) is created with three <span> elements.
    - The navigation items in the <ul> are used to create links to different sections of the page.
    - The right navigation has contact links for phone and email.

Intro Section

```
<section class="intro">
  <h1>Stockholm</h1>
  <p>Explore the beauty...</p>
</section>
```

- <section>: A semantic HTML5 element that groups related content. Here, it introduces Stockholm with a heading and a paragraph.
- <h1>: Represents the main heading for the section.
- <p>: Represents a paragraph of introductory text.

Second Navigation Section

```
<nav class="nav-2">
  <ul>
    <!-- Links for secondary navigation -->
  </ul>
</nav>
```

- <nav class="nav-2">: A secondary navigation bar with links to other sections of the page (e.g., History, Climate, etc.).
- <ul>: An unordered list used for the navigation links.

History Section

```
<section id="History" class="history-section">
  <div class="container">
    <div class="text-content">
      <h2>Stockholm's History</h2>
      <p>Stockholm's History...</p>
    </div>
  </div>
  <!-- Embedded video -->
  <div class="video-container">
    <iframe src="..." allowfullscreen></iframe>
  </div>
```

```
</section>
```

- <section id="History">: Defines a section of the page dedicated to Stockholm's history. The id="History" allows for internal navigation links.
- <div class="container">: A container that can be used for layout purposes, typically centered or constrained in width.
- <h2>: A heading for the subsection (History).
- <p>: Describes the content, in this case, the history of Stockholm.
- <iframe>: Embeds a video from YouTube in the section for additional context.

Climate Section

```
<section id="Climate" class="climate-section">
  <div class="container">
    <div class="text-content">
      <h2>Stockholm's Climate...</h2>
    </div>
  </div>
  <!-- Slideshow Container -->
  <div class="slideshow-container">
    <div class="slides">
      <img src="images/spring.jpg" alt="Stockholm in Spring">
      <!-- Images for different seasons -->
    </div>
    <!-- Navigation Arrows -->
    <span class="prev" onclick="changeSlide(-
1)">&#10094;</span>
    <span class="next"
onclick="changeSlide(1)">&#10095;</span>
  </div>
  <!-- Thumbnails for slideshow -->
  <div class="thumbnails">
```

```
    <img src="images/spring.jpg" alt="Spring" onclick="showSlide(0)">
    <!-- Additional thumbnails -->
  </div>
</section>
```

- <section id="Climate">: Section with an id for navigation purposes.
- **Slideshow**:
    - <div class="slideshow-container">: Contains images for the seasonal slideshow.
    - <img src="...">: Displays images for different seasons in Stockholm.
    - **Arrows (**<span class="prev"> **and** <span class="next">**)**: Allow users to navigate between slides.
    - **Thumbnails**: Small clickable images below the slideshow to navigate directly to each image.

What-to-do Section

```
<section id="What-to-do" class="what-to-do-section">
  <div class="container">
    <div class="text-content">
      <h2>What to Do in Stockholm</h2>
      <p>What to do in Stockholm...</p>
    </div>
  </div>
  <div class="video-container">
    <iframe src="..." allowfullscreen></iframe>
  </div>
</section>
```

- Similar structure as the History section, with a description of activities in Stockholm and an embedded video.

Where-to-stay Section

```
<section id="Where-to-stay" class="where-to-stay-section">
  <div class="container">
    <div class="text-content">
      <h2>Where to Stay in Stockholm</h2>
      <p>Where to stay in Stockholm...</p>
    </div>
  </div>
  <div class="accommodation-list">
    <!-- Accommodation options -->
  </div>
</section>
```

- **Accommodation List**: Lists different areas in Stockholm with hotels and other lodging recommendations, organized in individual <div> elements for each area. Each accommodation includes a link to a booking website.

4. Interactive Elements: JavaScript Functionality

Some sections, such as the slideshow in the **Climate Section**, require JavaScript for interaction:

```
<span class="prev" onclick="changeSlide(-1)">&#10094;</span>
<span class="next" onclick="changeSlide(1)">&#10095;</span>
```

- The onclick event calls the JavaScript function changeSlide(), which will change the slide based on the user's interaction.
- Thumbnails allow for direct navigation to a specific image via the showSlide() function.

5. Attraction Section:

```
<section id="Attractions">
  <h2>Attractions</h2>
  ...
</section>
```

- **Purpose:** This section is dedicated to showcasing different attractions in Stockholm. The title Attractions introduces the section.
- The <section> tag groups the content, making it easier to style and manage. The id="Attractions" is used for linking or styling purposes.

6. Slider Controls:

```
<div class="slider-controls">
  <button class="prev-slide">&lt;</button>
  <button class="next-slide">&gt;</button>
</div>
```

- **Purpose:** These buttons (prev-slide and next-slide) are used to navigate between slides in a carousel or image slider. The &lt; and &gt; are HTML entities for the left and right arrow symbols, respectively.
- **How it works:** When clicked, JavaScript (not included here) will trigger the carousel to show the previous or next slide, respectively.

7. Slider Structure:

```
<div class="attractions-slider">
  <div class="slider-images">
    <!-- Slide 1 -->
    <div class="slider-slide">
      ...
    </div>
    <!-- Slide 2 -->
    <div class="slider-slide">
      ...
    </div>
  </div>
</div>
```

- **Purpose:** This is the container for the image carousel. It has a class attractions-slider, which will be styled and potentially manipulated by JavaScript.
- The slider-images contains all the individual slides (slider-slide), which are the images of each attraction.
- **How it works:** Only one slide will be visible at a time, and JavaScript controls which one is shown based on the user interactions (e.g., clicking the arrows).

8. Attraction Cards:

```
<div class="attraction-card" data-modal-target="#storkyrka-modal">
  <img src="images/storkyrkan.jpeg" alt="Storkyrkan">
  <h4>Storkyrkan</h4>
  <p>SEK 100.00 / person normally</p>
</div>
```

- **Purpose:** Each attraction is presented as a card, with an image, title (h4), and price information.
- The data-modal-target="#storkyrka-modal" is an attribute that indicates the target modal for this card. When clicked, this card will trigger the modal with the specified ID (e.g., #storkyrka-modal).
- **How it works:** Clicking a card will open the corresponding modal with more detailed information about the attraction.

9. Bullet Controls:

```
<div class="bullet-controls">
  <span class="bullet active"></span>
  <span class="bullet"></span>
</div>
```

- **Purpose:** These are the bullet indicators under the slider, representing each slide.
- The class active is used to highlight the bullet for the currently displayed slide.
- **How it works:** JavaScript will change the active class on the bullet as the user navigates through the slides.

10. Modals:

```
<div id="storkyrka-modal" class="attraction-modal">
  <div class="modal-content">
    <button class="modal-close">&times;</button>
    <h3>Storkyrkan - Stockholm Cathedral</h3>
    <img src="images/storkyrkan-interior.jpeg" alt="Storkyrkan Interior">
    <p>...</p>
  </div>
</div>
```

- **Purpose:** Each attraction card opens a modal (popup) window when clicked. These modals provide more detailed information about the attraction, including images, descriptions, and external links for further exploration or ticket purchases.
- **How it works:** The data-modal-target in each attraction card matches the id of a modal (e.g., #storkyrka-modal). When a card is clicked, JavaScript displays the relevant modal.
- The modal also includes a close button (&times;), which will hide the modal when clicked.

- ntent is well-organized and accessible.

11. Activities, Events, Eat & Drink sections

Each of these sections (Attractions, Activities, Events, Eat & Drink) has the same layout and functionality, providing a consistent experience across the website. They use sliders to display multiple items, allow users to click through to view more details, and provide modal popups with further information when a card is clicked. This uniform layout enhances the usability and flow of the website, ensuring that users know how to interact with the content regardless of the section they're in.

12.Contact Section:

The **Contact** section provides an easy way for users to get in touch with the site owner. It is designed to engage users by offering them a clear call to action. Here's how it's structured:

```
<section id="Contact" class="contact-section">
  <div class="contact-overlay">
    <h2>Get in Touch with Us</h2>
    <p>Your next adventure starts with a single message. We're
here to make it happen.</p>
    <button id="contact-button"
onclick="location.href='mailto:elli.corleone@yahoo.com?subje
ct=Contact Us Inquiry'">
      Contact Us
    </button>
  </div>
</section>
```

<section id="Contact" class="contact-section">:

This defines the section element with an id="Contact" to link to this part of the page from elsewhere (like a navigation bar). The class contact-section allows the section to be styled in a specific way.

Contact Overlay (<div class="contact-overlay">**)**:

A div wrapper, used to contain the content inside the section, and applies an overlay effect for visual enhancement.

Contact Button (<button id="contact-button">**)**:

This button allows users to quickly open their default email client with a pre-filled subject ("Contact Us Inquiry") and email address (elli.corleone@yahoo.com) when clicked. This is achieved using the mailto: link inside the onclick event.

13.Footer Section:

The **Footer Section** typically provides site-wide information, links, and contact details. It's located at the bottom of the webpage for easy access. Let's break down each component:

About Us Modal:

This modal appears when users click the "About Us" link. It contains information about the website, explaining it was created as a university project and is not for commercial use.

Privacy Policy Modal:

A modal that describes the website's privacy policy, ensuring transparency about data collection practices. This is important for legal and trust-building purposes.

Contact Information (<div class="footer-contact">**)**:

Provides contact details like email, phone number, and address. These are essential for user inquiries or follow-up.

The email is a clickable link, so users can directly send an email by clicking on it.

Social Media Links (<div class="footer-social">**)**:

1.

Links to social media accounts (Facebook, Twitter, Instagram, LinkedIn) represented with social media icons, which allow users to connect with the website on different platforms.

Footer Bottom (<div class="footer-bottom">**)**:

A copyright notice that states the ownership and rights of the content on the website. Additionally, there is a link to the privacy policy for further user information.

CSS

- **Global Reset:**

```
*,
*::before,
*::after {
  margin: 0;
  padding: 0;
  box-sizing: border-box; /* Ensures padding doesn't affect element width */
}
```

- **Resets margin and padding** across all elements to ensure consistent spacing across browsers.
- box-sizing: border-box prevents padding and border from altering the element's total width.

- **General Setup:**

```
html, body {
  font-family: 'Roboto', sans-serif;
  margin: 0;
  padding: 0;
  overflow-x: hidden; /* Prevents horizontal scrolling */
}

html {
  scroll-behavior: smooth; /* Smooth scroll behavior */
}

body {
  background: linear-gradient(to bottom, #f5f5f5, #e0e0e0); /* Light gray gradient */
}
```

- font-family: 'Roboto', sans-serif: Applies a modern sans-serif font.
- overflow-x: hidden: Disables horizontal scrolling to prevent layout issues.
- scroll-behavior: smooth: Smoothens scrolling when navigating within the page.
- **Gradient background** adds a light gray-to-dark gray effect.

- **Modal Body Lock:**

```
body.modal-open {
  overflow: hidden; /* Prevent scrolling */
  position: fixed;  /* Fix position */
  width: 100%;      /* Full width */
  top: 0;           /* Fix to top */
}
```

- **Modal open behavior**: Locks the page to prevent scrolling when a modal is active.

- **Typography:**

```
h2 {
  font-family: 'Montserrat', sans-serif;
  color: #1A2A45; /* Navy */
  font-size: 1.5rem;
  text-align: justify;
  text-transform: uppercase;
  border-left: 5px solid #1A2A45;
}

p {
  color: #2F4F4F; /* Dark Slate Gray */
  font-size: 1rem;
  line-height: 1.5;
  text-align: justify;
}

h3, h4 {
  font-family: 'Montserrat', sans-serif;
  text-transform: capitalize;
  text-align: justify;
}
```

- **Headings** (h2, h3, h4): Use **Montserrat** for a modern, consistent style.

  - **Text color and size** are chosen for readability, with **navy**, **slate gray**, and **steel blue** tones.
  - **Justified text** for a clean, professional look.

- **Header Styling:**

```
header {
  background-color: #1A2A45; /* Navy Blue */
  color: #FFFFFF; /* White text */
  padding: 20px 70px;
  position: fixed; /* Fixed header */
  z-index: 1000;
}
```

  - **Fixed header**: Stays at the top with a **navy background** and **white text**.

- **Active Links in Navbar:**

```
header nav .left-nav ul li a.active {
  color: #F8B800; /* Golden Yellow */
  font-weight: 700;
  transition: color 0.2s ease;
}
```

  - **Active links**: Highlighted with **golden yellow** and **bold text**, with a smooth color transition.

- **Navbar Layout:**

```
header nav {
  display: flex;
  justify-content: space-between;
}

header nav .left-nav ul {
  list-style-type: none;
  display: flex;
}

header nav .right-nav {
  text-align: right;
}
```

- **Flexbox** layout: Aligns items in the navbar efficiently, with **space-between** for left and right sections.

- **Icons in Navbar:**

```
header nav .right-nav .contact-icon img,
header nav .right-nav .email-icon img {
  width: 25px;
  height: 25px;
  transition: all 0.3s ease;
}
```

- **Icons**: Adjusted to **25px** for consistency and have a **hover effect** for interactivity.

- **City Name & Hamburger:**

```
.city-name {
  display: none; /* Hidden by default */
  font-size: 2rem;
  color: #F8B800; /* Yellow */
}

.hamburger {
  display: none; /* Hidden by default */
  width: 25px;
  height: 20px;
}
```

- **City name**: Hidden on larger screens, shown on smaller ones (responsive design).
- **Hamburger menu**: Hidden by default, used for mobile navigation.

Here are the key highlights, with a bit of code included:

- **Navigation Bar (.nav-2) Styling:**

  - **Default**: A mustard yellow background with navy text for contrast. It has padding, a subtle shadow, and is centered.
  - **Fixed State**: When the navbar is fixed at the top, it has a white text color, remains at the top, and has a slight shadow.
  - **Active Link**: The active link color is dark blue, bold, and highlighted on hover with a blue shade.

```
.nav-2 {
  background-color: #D9A400;
  color: #1A2A45;
  padding: 20px 0;
}

.nav-2.fixed {
  position: fixed;
  top: 0;
  width: 100%;
  color: #FFFFFF;
}
```

- **Intro Section:**

  - **Background**: A gradient overlay with an image (Stockholm), with text aligned at the bottom.

- **Text Styling**: The main title and paragraph are styled with uppercase, bold text and a blue shadow for emphasis.

```
.intro {
  background: linear-gradient(to bottom, rgba(0, 0, 0, 0.3),
rgba(0, 0, 0, 0.3)), url('images/stockholm.jpg');
  padding-bottom: 5%;
}

.intro h1 {
  font-family: 'Lora', serif;
  color: #fff;
  text-shadow: 1px 0 #003f8e, -1px 0 #003f8e, 0 1px #003f8e, 0
-1px #003f8e;
}
```

- **Video Container:**

  - **Aspect Ratio**: Ensures the video maintains a 16:9 aspect ratio with padding-top: 56.25% for responsiveness.
  - **Styling**: A border-radius and shadow give it a polished look.

```
.video-container {
  position: relative;
  padding-top: 56.25%;
  overflow: hidden;
  box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.2);
}
```

- **Slideshow Section:**

  - **Container**: Ensures images are shown as a carousel with flexbox layout.
  - **Arrows**: Custom navigation arrows for transitioning between slides, with hover effects.

```
.slideshow-container {
  position: relative;
  max-width: 800px;
  overflow: hidden;
}

.slides img {
  display: none;
}

.prev, .next {
  position: absolute;
  background-color: white;
  color: black;
  padding: 5px 15px;
}
```

- **Accommodation Section:**
  - **Styling**: Displays accommodation items as columns with images that are cropped to fit.
  - **Links**: Accommodation links are bold and underlined on hover.

```
.accommodation-item img {
  width: 100%;
  max-height: 400px;
  object-fit: cover;
}

.accommodation-item a:hover {
  text-decoration: underline;
}
```

- **General Section Layout:**

- **Containers**: Sections like history, climate, what-to-do, and where-to-stay use flexbox for layout.
- **Padding**: Consistent padding and border-radius for rounded corners.

```
.history-section, .climate-section {
  padding: 40px 20px;
  max-width: 1200px;
  margin: 0px 250px;
  border-radius: 10px;
}
```

The **Attraction Section** (and similarly structured sections like **Activities**, **Events**, **Eat & Drink**, etc.) in the CSS is styled with flexibility and responsiveness. Each section is designed to highlight content in an organized manner using modern design principles such as **flexbox**, **hover effects**, and **smooth transitions**.

- **Attraction Section Styling**

The **Attraction** section is positioned in a container with a max width for large screens, and is vertically centered using **flexbox**:

```
#Attractions {
  margin: 0 auto;
  max-width: 1200px;
  text-align: center;
  position: relative;
  height: calc(100vh - 190px); /* Adjust height to fit below
navigation bars */
  display: flex;
  flex-direction: column;
  justify-content: center; /* Center content vertically */
  padding: 0px 20px;
  padding-bottom: 20px;
  margin-bottom: 15px; /* Space between sections */
  scroll-margin-top: 100px; /* For smooth scrolling to work
nicely */
}
```

This creates a full-screen section with appropriate spacing and ensures it scrolls smoothly.

- **Attraction Heading Styling**

The heading (h2) within the **Attraction** section has a **left-aligned** text with a subtle margin for spacing:

```
#Attractions h2 {
  text-transform: capitalize;
  margin-left: 15px;
  margin-top: 20px;
  text-align: left;
  font-family: 'Roboto', sans-serif;
  font-size: 1.5rem;
  color: #333;
  font-weight: 500;
  position: relative;
  border: none;
  padding-left: 0px;
}
```

This ensures the heading appears clean, left-aligned, and uses a modern font style for a professional look.

- **Attraction Slider (Image Carousel)**

For displaying multiple **attractions** in a slideshow format, a **slider** is used. The images are placed side by side in a **flexbox container**, and the slider is styled with smooth transitions and controls (previous and next buttons):

```
.attractions-slider {
  position: relative;
  width: 100%;
  max-width: 1200px;
  overflow: hidden;
  margin: 30px auto;
  height: 360px;
}

.slider-images {
  display: flex;
  align-items: center;
  transition: transform 0.6s ease-in-out;
  height: 100%;
}

.slider-slide {
  display: flex;
  justify-content: center;
  align-items: center;
  min-width: 100%;
  height: 360px;
}
```

This section uses a flexbox layout to align slides horizontally and allows smooth transitions between them.

- **Slider Controls (Navigation Buttons)**

The previous and next navigation buttons are placed **absolutely** at the top-right of the slider and are styled to be circular with smooth hover effects:

```
.slider-controls {
  position: absolute;
  top: 0;
  right: 10px;
  transform: translate(-20%, 90%);
  display: flex;
  gap: 5px;
  z-index: 10;
}

.slider-controls button {
  background-color: #ffffff;
  border: none;
  color: #333;
  cursor: pointer;
  font-size: 20px;
  width: 40px;
  height: 40px;
  border-radius: 50%;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.2);
  transition: transform 0.3s ease, background-color 0.3s ease;
}

.slider-controls button:hover {
  background-color: #f0f0f0;
  transform: scale(1.1);
}
```

These buttons have a hover effect to increase interactivity and make the UI more engaging.

- **Attraction Cards (Individual Attractions)**

Each individual attraction is placed inside a **card-style** element with a shadow effect and hover animations:

```
.attraction-card {
  margin: 15px;
  width: 260px;
  height: auto;
  text-align: center;
  background: transparent;
  border-radius: 15px;
  overflow: hidden;
  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);
  transition: transform 0.3s ease, box-shadow 0.3s ease;
}

.attraction-card img {
  width: 100%;
  height: 200px;
  object-fit: cover;
  transition: transform 0.3s ease, box-shadow 0.3s ease;
}

.attraction-card:hover {
  transform: translateY(-5px);
  box-shadow: 0 8px 16px rgba(0, 0, 0, 0.15);
}

.attraction-card img:hover {
  transform: scale(1.05);
  box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
}
```

Each **card** includes an image that scales slightly on hover, enhancing the interaction feel.

- **Bullet Navigation for Slider**

To navigate between the images in the slider, **bullet points** are used. These are small, clickable circles:

```
.bullet-controls {
  margin-top: 15px;
  display: flex;
  justify-content: center;
}

.bullet {
  width: 12px;
  height: 12px;
  margin: 0 5px;
  background-color: #d1d1d1;
  border-radius: 50%;
  cursor: pointer;
  transition: background-color 0.3s ease;
}

.bullet.active {
  background-color: #555;
}
```

These bullets act as additional navigation controls, providing users with a visual indication of the current slide.

- **Other Sections (Activities, Events, Eat & Drink, etc.)**

Other sections such as **Activities**, **Events**, and **Eat & Drink** are styled similarly to the **Attraction** section, with slight variations in content and layout. They follow the same principles for **flexbox layout**, **text styling**, and **smooth scrolling**.

For example, the **Activities** section might have content specific to different activity types, but the structure will remain largely unchanged:

```
#Activities {
  margin: 0 auto;
  max-width: 1200px;
  text-align: center;
  position: relative;
  height: calc(100vh - 180px);
  display: flex;
  flex-direction: column;
  justify-content: center;
}
```

- **Footer Container**

The footer section uses a gradient background and centers its content with flexible boxes:

```
.site-footer {
  background: linear-gradient(135deg, #1A1A1A, #2D2D2D); /* Dark gradient */
  color: #EDEDED; /* Light text */
  padding: 60px 20px;
  line-height: 1.8;
}

.footer-container {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-between;
  gap: 40px;
}
```

- **Footer Branding**

The branding part of the footer includes a gold-colored heading and some text:

```
.footer-branding h1 {
  font-size: 1.3rem;
  color: #FFD700; /* Gold */
}
```

```
.footer-branding p {
  font-size: 0.8rem;
}
```

- **Footer Navigation**

Links in the footer navigation are underlined in gold with a hover effect:

```
.footer-nav ul li a {
  color: #EDEDED;
  text-decoration: none;
}

.footer-nav ul li a:hover {
  color: #FFD700;
  transform: scale(1.05);
}
```

- **Footer Contact**

Contact details and links in the footer are styled with subtle hover effects:

```
.footer-contact a {
  color: #EDEDED;
  text-decoration: none;
}

.footer-contact a:hover {
  color: #FFD700;
  text-decoration: underline;
}
```

14. Footer Social

Social media icons are styled as circles with hover effects that change their background color and add a shadow:

```
.footer-social .social-icons a {
  background: #EDEDED;
  color: #1A1A1A;
  width: 40px;
  height: 40px;
  border-radius: 50%;
}

.footer-social .social-icons a:hover {
  background: #FFD700;
  transform: scale(1.2);
  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
}
```

- **Footer Bottom**

The bottom section of the footer centers the copyright and other information with subtle text styling:

```
.footer-bottom {
  text-align: center;
  font-size: 0.8rem;
  border-top: 1px solid rgba(237, 237, 237, 0.3);
}
```

JavaScript

1. Ensure the page scrolls to the top after refresh

```
if (window.scrollY !== 0) {
  window.scrollTo(0, 0);
}
```

This checks if the page is already at the top, and if not, it scrolls to the top on page load.

2. Smooth Scroll for Navigation Links

```
const scrollToSection = (event, targetSelector) => {
  event.preventDefault();
  const targetSection = document.querySelector(targetSelector);
  const headerHeight = document.querySelector("header").offsetHeight || 0;
  const nav2Height = document.querySelector(".nav-2").offsetHeight || 0;
  const totalNavbarHeight = headerHeight + nav2Height;
  const targetPosition = targetSection.offsetTop - totalNavbarHeight;

  window.scrollTo({
    top: targetPosition,
    behavior: 'smooth'
  });
};
```

Smooth scroll to sections, adjusting for navbar heights.

3. Desktop-Specific Navbar Behavior

```
const handleScroll = () => {
  if (window.innerWidth > 768) {
    nav2.classList.add("fixed");
    nav2.style.top = `${headerHeight}px`;
  } else {
    nav2.classList.remove("fixed");
  }
};
window.addEventListener('scroll', handleScroll);
```

Makes the .nav-2 navbar sticky when scrolling, but only on desktop screens.

4. Active State for Navigation Links

```
headerLinks.forEach(link => {
  link.addEventListener('click', function (e) {
    removeActiveClasses(headerLinks);
    addActiveClass(this);
    scrollToSection(e, this.getAttribute('href'));
  });
});
```

Adds the "active" class to the clicked navbar link and scrolls to the respective section.

5. Video Modal Script

```
videoTrigger.addEventListener('click', () => {
  modal.style.display = 'flex';
  modalVideo.src =
'https://www.youtube.com/embed/5fcEntH5a5I?autoplay=1';
});
modalOverlay.addEventListener('click', closeModal);
```

Opens a video modal and autoplay the YouTube video when a trigger is clicked.

6. Attractions Slider

```
function initializeSlider(container) {
  let currentIndex = 0;
  function updateSlider() {
    sliderImages.style.transform = `translateX(-${currentIndex *
100}%)`;
```

```
  }

  nextButton.addEventListener('click', () => { currentIndex++;
updateSlider(); });
  prevButton.addEventListener('click', () => { currentIndex--;
updateSlider(); });
}
```

Initializes the slider, enabling next/previous navigation between slides.

7. Modals Script

```
cards.forEach(card => {
  card.addEventListener('click', () => {
    const targetModal =
document.querySelector(card.getAttribute('data-modal-
target'));
    targetModal.classList.add('show');
  });
});
```

Opens a modal when an attraction card is clicked.

8. Hamburger Menu (Mobile)

```
hamburgerMenu.addEventListener('click', () => {
  leftNav.classList.toggle('active');
});
```

Toggles the visibility of the mobile menu when the hamburger icon is clicked.

9. Adjust Video Aspect Ratio

```
function adjustVideos() {
  videoContainers.forEach((videoContainer) => {
    const iframe = videoContainer.querySelector('iframe');
    const aspectRatio = 9 / 16; // 16:9 ratio
    const containerWidth = videoContainer.offsetWidth;
    videoContainer.style.height = `${containerWidth *
aspectRatio}px`;
  });
}
window.addEventListener('resize', adjustVideos);
```

Adjusts the size of the video to maintain a 16:9 aspect ratio when the window is resized.

10. Reset Slider on Load/Resize

```
function resetSlider() {
  slider.scrollLeft = 0; // Reset the slider to the first slide
}
window.addEventListener('load', resetSlider);
window.addEventListener('resize', resetSlider);
```

Resets the slider position to the first image when the page loads or is resized.

These snippets cover the main functionality of your code. Let me know if you'd like further details!