Siyabonga Coka

ST10477943

Web Development

**Step 1: Plan the Structure of the HTML**

Before writing any CSS, I created the structure of my webpage using semantic HTML elements like:

* <h1> for the main title
* <div> containers for layout sections
* <a> for clickable links
* <img> for images
* <video> for multimedia
* <footer> for bottom information

This helped me separate content from style.

**Step 2: Apply a CSS Reset**

To remove default browser spacing and make styling consistent, I used a universal selector:

\* {  
 margin: 0;  
 padding: 0;  
 box-sizing: border-box;  
}

**Step 3: Style the Body**

I set the background color to black for a nightclub feel, and white text for readability:

body {  
 font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
 background-color: #000;  
 color: #fff;  
 padding: 30px 20px;  
}

**Step 4: Style the Headings**

To make the heading stand out, I used gold color, center alignment, and large font size:

h1 {  
 color: gold;  
 text-align: center;  
 font-size: 2.8em;  
 margin-bottom: 20px;  
}

**Step 5: Style Content Boxes (like .description)**

I styled .description using background color, padding, border, and center alignment:

.description {  
 max-width: 900px;  
 margin: 0 auto 30px auto;  
 font-size: 1.2em;  
 line-height: 1.8;  
 color: #ccc;  
 text-align: center;  
 background-color: #111;  
 padding: 20px;  
 border: 1px solid gold;  
 border-radius: 10px;  
}

**Step 6: Style the Links**

I made the links look like buttons and added hover effects for interactivity:

.back-link a {  
 color: gold;  
 text-decoration: none;  
 font-weight: bold;  
 border: 1px solid gold;  
 padding: 10px 20px;  
 border-radius: 6px;  
 transition: all 0.3s ease;  
}  
  
.back-link a:hover {  
 background-color: gold;  
 color: black;  
}

**Step 7: Create the Image Gallery**

I used Flexbox to create a responsive layout for the gallery:

.image-gallery {  
 display: flex;  
 justify-content: center;  
 flex-wrap: wrap;  
 gap: 20px;  
 margin: 30px 0;  
}  
  
.image-gallery img {  
 width: 180px;  
 height: 180px;  
 object-fit: cover;  
 border-radius: 10px;  
 border: 2px solid gold;  
 transition: transform 0.3s ease;  
}  
  
.image-gallery img:hover {  
 transform: scale(1.05);  
}

**Step 8: Style the Video**

I made the video responsive and added a border to match the theme:

video {  
 display: block;  
 max-width: 100%;  
 height: auto;  
 margin: 0 auto;  
 border: 2px solid gold;  
 border-radius: 10px;  
}

**Step 9: Style the Footer**

I added spacing and color to the footer section:

footer {  
 margin-top: 50px;  
 text-align: center;  
 font-size: 0.9em;  
 color: #aaa;  
 border-top: 1px solid #333;  
 padding-top: 15px;  
}  
  
footer span {  
 display: block;  
 margin-top: 5px;  
 color: gold;  
}

**Step 10: Make It Responsive**

To ensure the site works well on mobile screens, I used a media query:

@media (max-width: 600px) {  
 h1 {  
 font-size: 2em;  
 }  
  
 .description {  
 font-size: 1em;  
 }  
  
 .image-gallery {  
 flex-direction: column;  
 align-items: center;  
 }  
  
 .image-item {  
 width: 100%;  
 max-width: 300px;  
 }  
  
 .image-gallery img {  
 height: auto;  
 }  
}

**Step 11: Add Transitions for Better UX**

To make the site feel smooth, I used transition properties on hover effects:

transition: all 0.3s ease;

This was added to buttons and images for interactive feedback.