Proyecto I - [Desarrollo de Landing Page](https://www.workana.com/job/desarrollo-de-landing-page-en-wix?ref=projects_9)

Publicado: Ayer

Propuestas: 7

Plazo de Entrega: 16/12/2024

Descripción del Proyecto:

Se requiere un desarrollador con experiencia para la creación de una landing page en un plazo máximo de 15 días. El proyecto consiste en el diseño y desarrollo de una página web con un enfoque minimalista, vanguardista o futurista, dirigida a una empresa que busca transmitir innovación, tecnología y sostenibilidad.

El diseño debe ser atractivo y funcional, optimizado para la visualización en diferentes dispositivos. Se requiere la integración de elementos como imágenes de alta calidad, tipografía moderna, paleta de colores coherente con la identidad de la marca y animaciones sutiles.

Categoría: Diseño y Multimedia  
Subcategoría: Diseño Web  
¿Qué necesitas?: Crear un nuevo diseño personalizado  
¿Es un proyecto o una posición?: Un proyecto  
Disponibilidad requerida: Según se necesite

**Búsqueda de referencias**

Para inspirar el diseño de la landing page, he revisado varios ejemplos de páginas relacionadas con estética, tecnología, y diseño minimalista que destacan por su estructura y enfoque moderno. Aquí están las principales referencias y los puntos clave que pueden ser útiles:

1. **Fig.1 Beauty** y **TWYG Skincare**:
   * Sitios orientados al cuidado personal que utilizan un diseño limpio, con tipografías modernas y paletas de colores neutros combinadas con toques de verde y azul para evocar sostenibilidad y calma.
   * Incorporan imágenes de alta calidad en hero sections y secciones de productos para destacar servicios de lujo.
   * Usan animaciones suaves al desplazar la página para mantener la experiencia fluida y atractiva​

[Justinmind](https://www.justinmind.com/ui-design/landing-page-examples)

[Lapa Ninja](https://www.lapa.ninja/category/beauty/)

1. **Flora**:
   * Un diseño simple que destaca con el uso efectivo del espacio en blanco y un enfoque en el mensaje clave. Incluye llamadas a la acción claras y destacadas con botones prominentes.
   * Ideal para comunicar profesionalismo e innovación​

[Justinmind](https://www.justinmind.com/ui-design/landing-page-examples)

1. **Warby Parker**:
   * Aunque es para un sector distinto (lentes), el enfoque en imágenes de productos y una experiencia interactiva es una inspiración útil para galerías de imágenes y posibles animaciones de transición​

[Justinmind](https://www.justinmind.com/ui-design/landing-page-examples)

Estas referencias son ideales para una página del sector de estética que busca transmitir innovación y sostenibilidad. Se enfocan en la claridad visual, una navegación intuitiva y el uso de elementos modernos.

**Siguientes pasos**

1. **Planificación**: Con base en estas ideas, comenzaremos a estructurar la landing page con las secciones clave mencionadas antes.
2. **Desarrollo inicial**: Elaboraré el contenido básico de la página con una estructura simple en HTML, CSS y JavaScript.
3. **GitHub**: Durante el desarrollo, te guiaré para subir los avances al repositorio de GitHub y asegurarte de que esté listo para revisiones y ajustes.

A continuación, desarrollaré el prototipo inicial de la landing page para el centro de estética con el enfoque planteado. Vamos a comenzar con la estructura en HTML, los estilos en CSS y un script básico en JavaScript para elementos interactivos.

**Estructura del prototipo inicial**

**1. HTML (Estructura)**

Crearé un archivo index.html con la siguiente estructura:

* Uso de etiquetas semánticas (header, main, footer).
* Inclusión del logo, navegación, hero section, galería, y formulario.

**2. CSS (Diseño)**

Crearé un archivo style.css que:

* Incluya un diseño responsivo usando flexbox y/o grid.
* Use colores basados en el logotipo adjunto.
* Añada estilos modernos para botones, formularios y la galería.

**3. JavaScript (Interactividad)**

Crearé un archivo scripts.js que:

* Maneje interacciones básicas, como la validación del formulario.
* Incluya animaciones suaves para el desplazamiento o elementos dinámicos como un carrusel en la galería.

**2. CSS - Galería**

**Planificación de los estilos para la galería**

1. **Diseño tipo "grid"**:
   * Usaremos **CSS Grid** para organizar las imágenes en filas y columnas con márgenes adecuados.
   * Las imágenes serán responsivas (se ajustarán al tamaño de la pantalla).
2. **Estilos para la galería**:
   * Añadiremos un efecto de **hover** (por ejemplo, zoom suave) para hacerla más interactiva.
   * Usaremos un margen interno entre las imágenes para evitar que se toquen.
3. **Código base para la galería**: Aquí tienes un ejemplo del diseño básico que implementaremos en CSS:

.image-grid {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));

gap: 15px;

}

.image-grid img {

width: 100%;

height: auto;

border-radius: 8px; /\* Bordes redondeados \*/

transition: transform 0.3s ease; /\* Animación suave \*/

}

.image-grid img:hover {

transform: scale(1.05); /\* Zoom suave al pasar el cursor \*/

}

**Paso 1: Añadir las imágenes al index.html**

En la sección de la galería, crearé una estructura HTML utilizando una clase contenedora .image-grid y colocaré tus imágenes dentro de ella.

**Código actualizado para la galería en index.html:**

<section id="galeria">

<h2>Galería</h2>

<div class="image-grid">

<img src="example1.jpg" alt="Tratamiento facial 1">

<img src="example2.jpg" alt="Tratamiento facial 2">

</div>

</section>

**Paso 2: Estilos para la galería en styles.css**

Usaremos CSS Grid para organizar las imágenes en un diseño fluido y adaptable a diferentes tamaños de pantalla.

**Estilos CSS para la galería:**

/\* Estilo para el contenedor de la galería \*/

.image-grid {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));

gap: 20px;

padding: 20px;

}

/\* Estilo para las imágenes \*/

.image-grid img {

width: 100%;

height: auto;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /\* Sombra suave \*/

transition: transform 0.3s ease, box-shadow 0.3s ease;

}

/\* Efecto hover en las imágenes \*/

.image-grid img:hover {

transform: scale(1.05); /\* Zoom suave \*/

box-shadow: 0 6px 12px rgba(0, 0, 0, 0.2); /\* Más sombra al pasar el cursor \*/

}

**Paso 3: Probar el diseño**

1. Asegúrate de que los archivos example1.jpg y example2.jpg estén correctamente referenciados y que las rutas sean correctas.
2. Abre el archivo index.html en un navegador para ver cómo se visualiza la galería.
3. La galería debe adaptarse automáticamente al ancho de la pantalla, con las imágenes organizadas en filas y columnas dependiendo del espacio disponible.

**CSS – Formulario**

Para mejorar el estilo del formulario y hacerlo más atractivo y funcional, aplicaremos un diseño moderno con bordes redondeados, una paleta de colores coherente con la identidad del cliente, y mayor tamaño en los campos de entrada para mejorar la experiencia del usuario.

**Estilos para el formulario en styles.css**

Aquí tienes un diseño atractivo y limpio para el formulario:

/\* Contenedor del formulario \*/

form {

max-width: 600px;

margin: 0 auto;

padding: 20px;

background-color: #f9f9f9; /\* Fondo claro \*/

border-radius: 10px;

box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1); /\* Sombra \*/

font-family: 'Arial', sans-serif;

}

/\* Etiquetas de los campos \*/

form label {

display: block;

margin-bottom: 8px;

font-weight: bold;

color: #333;

font-size: 16px;

}

/\* Campos de entrada \*/

form input[type="text"],

form input[type="email"],

form textarea {

width: 100%;

padding: 15px;

margin-bottom: 20px;

border: 1px solid #ccc;

border-radius: 8px;

font-size: 16px;

transition: border-color 0.3s ease, box-shadow 0.3s ease;

}

/\* Estilo para el campo de texto en foco \*/

form input[type="text"]:focus,

form input[type="email"]:focus,

form textarea:focus {

border-color: #57c4d5; /\* Color del cliente \*/

box-shadow: 0 0 5px rgba(87, 196, 213, 0.5); /\* Sombra en foco \*/

outline: none;

}

/\* Botón de envío \*/

form button[type="submit"] {

width: 100%;

padding: 15px;

background-color: #57c4d5; /\* Color principal del cliente \*/

color: #fff;

font-size: 18px;

font-weight: bold;

border: none;

border-radius: 8px;

cursor: pointer;

transition: background-color 0.3s ease, transform 0.2s ease;

}

form button[type="submit"]:hover {

background-color: #45a3b3; /\* Color más oscuro al pasar el cursor \*/

transform: scale(1.05); /\* Efecto de agrandar suavemente \*/

}

**Resultado esperado**

1. **Formulario más grande**: Los campos de texto, área de texto y botón ocuparán más espacio, lo que facilita su uso.
2. **Bordes redondeados**: Para un diseño más moderno y amigable. 3

El formulario ahora tendrá un diseño más atractivo, con un tamaño más cómodo y funcional para los usuarios.

1. **Etiquetas claras**: Los campos tendrán etiquetas bien visibles y organizadas.
2. **Interacción mejorada**: Al hacer clic o escribir en los campos, el borde cambia a un color representativo de la marca, añadiendo un toque profesional.
3. **Botón llamativo**: El botón de envío será grande y con colores que contrasten, acompañado de un efecto al pasar el cursor.

3. **Script aplicado**

document.querySelector('form').addEventListener('submit', function(e) {

alert('Gracias por contactarnos. Pronto estaremos en comunicación.');

e.preventDefault();

});

**Cómo funciona este script**

1. **document.querySelector('form'):**
   * Selecciona el primer elemento <form> en el archivo HTML.
   * Es importante que el formulario esté correctamente definido en el HTML.
2. **.addEventListener('submit', function(e) { ... }):**
   * Escucha el evento submit del formulario.
   * Cuando el usuario envía el formulario (al hacer clic en el botón "Enviar"), se ejecuta la función proporcionada.
3. **alert(...):**
   * Muestra un cuadro de diálogo con el mensaje "Gracias por contactarnos. Pronto estaremos en comunicación."
4. **e.preventDefault():**
   * Cancela el comportamiento predeterminado del formulario (que normalmente enviaría los datos a un servidor).
   * Esto es útil en este caso porque no estamos manejando un servidor, y queremos mostrar el mensaje al usuario sin recargar la página.

**Solución recomendada**

Actualiza el archivo scripts.js de esta manera para asegurarte de que todo funciona correctamente:

// Asegurarnos de que el DOM esté completamente cargado antes de ejecutar el script

document.addEventListener('DOMContentLoaded', function() {

const form = document.querySelector('form');

if (form) { // Verificar que el formulario exista

form.addEventListener('submit', function(e) {

alert('Gracias por contactarnos. Pronto estaremos en comunicación.');

e.preventDefault();

});

} else {

console.error('El formulario no se encontró en el DOM.');

}

});

**Pasos para verificar el funcionamiento**

1. Asegúrate de que los archivos index.html, style.css, y scripts.js estén bien enlazados y ubicados en las carpetas correctas.
2. Abre la consola del navegador (Ctrl+Shift+I en Chrome) y revisa si hay errores. Si los hay, revisa el mensaje para identificar la causa (puedes compartirlo conmigo si necesitas ayuda).
3. Asegúrate de que el archivo scripts.js esté cargado correctamente en la sección de la consola "Sources".

**¿Cómo funciona el método form.reset()?**

1. **form.reset()**:
   * Este método pertenece al objeto de formulario en JavaScript.
   * Al llamarlo, todos los campos del formulario (inputs, textarea, etc.) vuelven a sus valores iniciales, es decir, se limpian si no tienen valores predeterminados.
2. **Orden en el script**:
   * Primero se detiene el comportamiento predeterminado con e.preventDefault().
   * Luego se muestra el alert.
   * Finalmente, se ejecuta form.reset() para vaciar los campos.

**Resultado esperado**

Cuando el usuario envíe el formulario:

1. Se mostrará un cuadro de diálogo con el mensaje de agradecimiento.
2. Todos los campos del formulario se limpiarán automáticamente.