

# Landing Page Using HTML and CSS

## Introduction

In this guide, you will learn how to create a simple landing page using **HTML** and **CSS**. We will use **semantic HTML elements** to give the webpage a clear structure and **CSS** to style the table, making it easy to read and visually appealing.

## Why Use Semantic Tags?

Semantic tags like `<header>`, `<section>`, and `<footer>` make webpages:

- More **accessible** for screen readers.
  - Easier to **read and maintain**.
  - Better structured for **search engines**.
- 

## Step 1: Setting Up the Project

First, create a new project folder and add the following files:

```
/landing-page-project
|— index.html (Structure of the webpage)
|— styles.css (Styles for the webpage)
```

These two files will contain the HTML structure and CSS styling for the landing page.

---

## Step 2: Writing the HTML Code

Open `index.html` and add the following code:

```
<header class="header-content">
  <a href="#logo" class="logo">
    
```

```
    <span class="logo-text">StudySync</span>
  </a>

  <nav class="nav">
    <a href="#home" class="nav-link">home</a>
    <a href="#features" class="nav-link">features</a>
    <a href="#pricing" class="nav-link">pricing</a>
    <a href="#blog" class="nav-link">blog</a>
    <a href="#about" class="nav-link">about</a>
  </nav>

  <a href="#contact" class="contact-btn"> contact us </a>

  <button type="button" class="menu-btn">
    

  </button>
</header>

<section class="main-sect">
  <div class="content-left">
    <p class="section-label">Very proud to introduce</p>

    <h1 class="section-title">Seamless Learning for Brighter Futures</h1>
    <p class="section-desc">
      Our innovative platform offers an effortless and seamless approach
      to learning, empowering students of all ages to achieve brighter
      futures. Join us on a transformative journey to simplify education
      and unlock your full potential.
    </p>
    <div class="button-grp">
      <a href="#start" class="start-btn">start now</a>
      <a href="#tour" class="tour-btn">take tour</a>
    </div>
  </div>
</section>

<footer class="footer">
```

```
<div class="footer-top">
  <div class="comp-logo">
    <a href="#" class="logo-link">
      
      StudySync
    </a>
  </div>
</footer>

</body>
</html>
```

### Explanation:

- **<header>**: Contains the logo and navbar.
- **<nav>**: Contains links to navigate.
- **<section>**: Contains different sections of the main part of the page, you can use the **<main>** tag too.
- **<footer>**: Contains the bottom most section and various links.
- **<meta name="viewport">**: Ensures proper display on different screen sizes.

---

## Step 3: Writing the CSS Code

Open **styles.css** and add the following styles:

```
:root{
  --primary-color: #6366f1;
  --accent-color: #3f83f8;
  --text-color: #333333;
  --link-color: #2563eb;
  --background-color: #ffffff;
  --light-gray: #f0f0f0;
  --gray: #808080;
  --dark-gray: #555;
```

```
}

*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  text-decoration: none;
  font-family: "Inter", sans-serif;
}

body{
  height: 100vh;
  text-align: center;
  background-color: var(--background-color);
  padding: 1rem;
}

@media screen and (min-width: 768px){
  .contact-btn{
    display: block;
    margin-right: .6rem;
  }
}

.contact-btn:hover{
  background-color: #5254f8;
}

.content-left{
  text-align: center;
  animation: slideFromLeft 1s ease forwards;
  opacity: 0;
}

@keyframes slideFromLeft{
  0%{
```

```
        opacity: 0;
        transform: translateX(-100%);
    }

    100%{
        opacity: 1;
        transform: translateX(0);
    }
}

.start-btn:hover{
    background-color: #5254f8;
}

.tour-btn:hover{
    background-color: var(--gray);
}
```

### Explanation:

- Uses **CSS Variables** for easy theme management.
  - Ensures consistent styling across the webpage.
  - Animations & Keyframes: Main section fades in when they appear..
  - Media queries for responsiveness- Mobile First Approach
  - Pseudo classes - Hover effects for interactivity.
  - Hides navigation menu on small screens.
  - Uses a mobile-friendly layout.
- 

## Step 4: Viewing Your Landing Page

1. Save both files in the same folder.
2. Open `index.html` in your browser.

You should now see a well-structured, styled Landing Page with clear headings and readable content.

---

## What You Learned:

- How to use **semantic HTML** for a well-structured webpage.
- How to add **animations** and use **keyframes**
- How to add **responsiveness**.
- How to use **pseudo classes**.
- How to organize files for a small project efficiently.

Now you have a beautiful Landing Page ready! Keep experimenting and improving your project. Add some more styles to give this project your personal touch.

## Still confused?

If you are facing difficulties building this landing page feel free to check out the completed project on [github](#).

Download the project into your system and get your hands dirty.