Documentation:
Agency One-Page Website

Overview

This document outlines the development of a one-page website for an agency, highlighting the implementation details, technologies used, and project structure. The website consists of five main sections: About Us, Services, Partners, Projects, and Contact Us. The site was developed using HTML, CSS, and JavaScript.

Project Structure

root/
|-- index.html
|-- css/
  |-- style.css
|-- js/
  |-- script.js
|-- images/
|-- README.md

1. `index.html`

The main HTML file that structures the content of the one-page website. It includes semantic HTML5 tags for accessibility and better SEO.

Key Sections:

- **Header**: Contains the navigation bar with links to each section.
- **About Us**: Provides an introduction to the agency.
- **Services**: Lists the services offered by the agency.
- **Partners**: Displays the agency's partners.
- **Projects**: Highlights featured projects.
- **Contact Us**: Provides contact information and a form for user inquiries.

2. `css/style.css`

The CSS file that styles the website, ensuring a responsive and visually appealing design. Key features include:

- **Responsive Design**: Utilizes media queries to ensure the website is mobile-friendly.
- **Styling Consistency**: Uses variables for consistent color schemes and fonts.
- **Hover Effects**: Adds interactivity to buttons and links.

3. `js/script.js`

The JavaScript file that adds interactivity and dynamic functionality to the website.

Key Functionalities:

- **Form Validation**: Validates the contact form to ensure all fields are correctly filled.
- **Modal Windows**: Expands project details in a modal view when a project is clicked.

Technologies Used

HTML

- Semantic HTML5 tags to structure the content.
- `<header>`, `<section>`, and `<footer>` tags for layout.

CSS

- **Flexbox**: For layout alignment and responsiveness.
- **Grid**: For creating a dynamic layout in the projects section.
- **Media Queries**: For ensuring the site is responsive across different devices.

JavaScript

- **DOM Manipulation**: To dynamically update the content and handle user interactions.
- **Event Listeners**: For handling user actions like clicks and form submissions.
- **Fetch API**: For future enhancement, enabling dynamic content loading.

Installation and Setup

Prerequisites

- A modern web browser (e.g., Chrome, Firefox).
- A code editor (e.g., VS Code, Sublime Text).

Steps to Run the Project

1. Clone the repository:
   git clone https://github.com/syntax-sensei/fitex_intern.git
2. Open `index.html` in your preferred web browser.

Future Enhancements

- **Backend Integration**: Connect a backend for form submissions and dynamic content management.
- **Performance Optimization**: Minify CSS/TailwindCSS and JavaScript files for faster load times.