# **Project Documentation: Digital Agency Website (HTML & CSS Only)**

**Project Title:** Digital Agency Website

**Description:** A responsive multi-page website built using only HTML and CSS, designed for a fictional digital agency that offers services like web design, digital marketing, and app development.

### Pages:

#### 1. Home (index.html)

- o Header with logo and navigation menu
- o Hero section with background image and promotional text
- Services overview section (3-4 cards with icons/text)
- Testimonials section
- o Call to Action (CTA)
- Footer with contact info and social links

### 2. About Us (about.html)

- Company mission and vision
- Team members with photos and roles
- o Company history or timeline

## 3. Services (services.html)

- o List of services offered, each in a card format
- Use of Flexbox for layout
- o Simple hover animations for interactivity

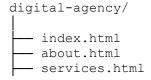
### 4. Blog (blog.html)

- o Grid layout showing latest articles or posts
- o Each post preview includes title, image, snippet
- Responsive design with hover effects

## 5. Contact (contact.html)

- o Contact form: name, email, subject, message
- o Embedded map image (static placeholder)
- Social media icons

### **Folder Structure:**



```
blog.html
contact.html
css/
style.css
images/
(images used in all pages)
```

## Features and Techniques Used:

- HTML5 Semantic Tags: <header>, <nav>, <section>, <article>, <footer>
- CSS Layouts: Flexbox and Grid
- **Responsive Design**: Media queries for mobile/tablet support
- **CSS Styling**: Custom fonts, colors, buttons, form styling
- Transitions & Animations: Hover effects and smooth transitions
- Pseudo-classes/elements: :hover, :focus, ::before, ::after
- Positioning: relative, absolute, fixed for nav and hero sections

### **Learning Outcomes:**

- Understand how to structure a multi-page website using HTML only
- Apply responsive CSS design techniques
- Build UI components like navigation bars, cards, footers, and forms
- Improve understanding of layout systems like Flexbox and Grid
- Practice clean code structure and reusable CSS classes

### **Future Enhancements:**

- Add JavaScript for form validation and interactivity
- Implement light/dark theme toggle
- Add animations using libraries (e.g., AOS.js)
- Connect contact form to backend (PHP or Node.js)
- Add CMS for blog (WordPress or custom)

**Author:** Kiroloes Reda Wassef **Tools Used:** VS Code, Google Fonts, Font Awesome (for icons) **Date Created:** July 2025