

# Developer Document for Portfolio Website

## 1. Project Overview

The portfolio website is a single-page web application designed to showcase professional skills and projects. It is structured to offer a seamless user experience through an interactive and visually appealing interface. This document outlines the key components, animations, and interactivity used to ensure a polished and professional portfolio presentation.

## 2. Page Structure

The portfolio will be a **single-page layout** consisting of the following sections:

- **Navigation Bar:** Provides links to different sections of the page for easy navigation.
- **Hero Banner:** A visually impactful banner at the top, showcasing key information.
- **About Us Section:** Introduction to the developer and the services provided.
- **Skills Section:** A visual breakdown of professional skills and expertise.
- **Contact Us Section:** Provides contact information and a feedback form.
- **Testimonial Section:** A carousel displaying testimonials from previous clients or projects.
- **Feedback Form:** Allows users to submit inquiries or feedback.
- **Footer:** Contains social media links, copyright information, and any additional contact details.

## 3. GreenSock and Animation Enhancements

For enhanced interactivity and visual appeal, the following animations and enhancements will be incorporated:

- **Smooth Scrolling:** Implemented using GreenSock (GSAP) for seamless navigation between sections of the page.
- **Scrolling Animations:** As the user scrolls, sections such as the skills section, testimonials, and feedback form will feature fade-in effects and slide animations using CSS and GSAP.

- **Animated Carousel:** An automatically moving carousel will be implemented for the testimonial section. It will rotate through feedback in a loop, with smooth transitions between slides.
- **Hero Image Animation:** The hero banner will include subtle animations such as parallax scrolling effects and motion on hover to engage the user.
- **CSS Animations on Scroll:** Various elements (e.g., skill icons, testimonials) will animate into view as the user scrolls down, using @keyframes and scroll-triggered CSS animations to capture attention.

#### 4. Interactivity

In addition to animations, the portfolio will include:

- **Interactive Navbar:** Clicking on the navbar links will smoothly scroll to the respective sections.
- **Carousel:** A fully interactive carousel in the testimonial section will automatically rotate but can be manually navigated by the user.
- **Feedback Form:** JavaScript will handle form validation and submission, ensuring proper data entry before submission.
- **Interactive Hover Effects:** Various elements such as buttons and skill icons will have hover effects, changing color, size, or applying slight animations for visual feedback.

Any advanced interactivity not covered in class will be researched and implemented using online resources, including forums and development platforms.

#### 5. Technology Stack

The following technologies will be used to build the portfolio:

- **Frontend:** HTML5, CSS3, JavaScript (for interactivity and animations), PHP (for handling backend processes such as form submissions).
- **Design Tools:** Wireframes were created using **Figma** for both desktop and mobile layouts. This ensures responsiveness and a user-friendly design.
- **Online Resources:** Inspiration for layout, design, and images has been sourced from various online repositories and design portfolios.

#### 6. Responsiveness

The portfolio is designed to be fully responsive, ensuring optimal viewing across devices:

- **Wireframes for Desktop and Mobile:** Wireframes were created for both desktop and mobile views to plan the layout and structure. Flexbox and media queries will be used in CSS to adapt the layout based on screen size.
- **Mobile Menu:** The navigation bar will collapse into a mobile-friendly menu on smaller screens, with an animation applied to transition the menu in and out of view.
- **Responsive Images and Videos:** All media will scale dynamically, ensuring proper display without compromising quality on different screen sizes.

## 7. Version Control, Repo Structure, and Sass Methodology

The project will be managed through **GitHub**, with a well-organized repository that follows best practices for version control and includes the **Sass folder structure** to enhance the maintainability of stylesheets.

### *Repository Structure*

- **Repository Name:** LastName\_FirstName\_Portfolio
- **Branches:** The repository will include multiple branches to organize development tasks. For example:
  - layout: Contains all HTML and basic layout work.
  - styles: Focuses on Sass and CSS styling.
  - interactivity: JavaScript and interactivity-related code.
  - animations: Includes all animation-related development (GreenSock/CSS animations).
  - **Master Branch:** All finalized work will be merged into the master branch before the project deadline. Regular merges from feature branches will be done to ensure the project remains synchronized.

### *Commit Strategy*

- **Frequent Commits:** Each feature, bug fix, or development stage will have a corresponding commit with detailed messages, making it easier to track changes and collaborate if necessary.
- **Commit Messages:** Clear, descriptive messages will be used to document each change (e.g., "Added smooth scrolling functionality" or "Refactored hero banner animation").

## *Sass Folder Structure*

To manage CSS styles efficiently, the **Sass 7-1 folder structure** methodology will be adopted. This system divides Sass files into different categories to ensure scalability and maintainability of the codebase. The **7-1 folder structure** means the styles are split into seven folders and one main `main.scss` file.

### Sass Folder Structure Breakdown:

1. **Base:** Contains the foundational elements (e.g., reset files, typography settings, and global styles).
  - `_reset.scss`: Reset file to remove default browser styling.
  - `_typography.scss`: Sets global font settings, headings, and paragraph styles.
2. **Components:** Stores reusable UI components (e.g., buttons, cards, forms).
  - `_buttons.scss`: Styles for button elements.
  - `_forms.scss`: Styles for form elements (contact form, feedback form).
3. **Layout:** Manages the layout structure of the page (e.g., navigation bar, grid systems).
  - `_header.scss`: Contains styles for the navigation bar.
  - `_footer.scss`: Contains styles for the footer.
  - `_grid.scss`: Grid systems for page layout, ensuring responsive design.
4. **Pages:** Contains specific styles for each section of the portfolio page.
  - `_home.scss`: Styles specific to the hero banner and introductory sections.
  - `_about.scss`: Styles for the about us section.
  - `_skills.scss`: Styles related to the skills section.
5. **Themes:** Allows for theme customization (light/dark modes).
  - `_theme.scss`: Manage color schemes and transitions for different themes if applicable.
6. **Utilities:** Contains helper classes or functions (e.g., margin, padding, hide/show classes).
  - `_utilities.scss`: Helper classes for spacing, visibility, alignment, etc.
7. **Vendors:** For third-party libraries or frameworks (e.g., Normalize.css, Bootstrap overrides).
  - `_vendors.scss`: Third-party library modifications and extensions.