Here's a sample software document for creating a portfolio website with two pages:

Project Name: Portfolio Website
Version: 1.0
Date: December 13, 2024
Author: Manoj Manjunatha

---

1. Introduction

This document outlines the specifications, structure, and functionality of a two-page portfolio website. The website will serve as a personal portfolio for showcasing the developer's skills, experience, and projects.

---

2. Project Overview

The portfolio website will consist of the following two pages:

- **Home Page**
  The home page will provide an introduction to the developer and highlight key features such as skills, experience, and recent projects.

- **Projects Page**
  This page will feature detailed descriptions of the developer's completed and ongoing projects, with links to relevant repositories or live demos.

---

3. Functional Requirements

Home Page
- **Header:**
  A navigation bar with links to the Home and Projects pages.

- **Main Section:**
  A brief introduction about the developer, including:
  - Name, title (e.g., Software Engineer)
  - A brief summary of skills and interests
  - A professional photo (optional)
  - Links to social media profiles (LinkedIn, GitHub, etc.)

- **Skills Section:**
  A list of the key technical skills, displayed in cards with icons for each technology used.

- **Footer:**
  A footer with copyright information and additional links.

Projects Page
- **Header:**
  The same navigation bar from the home page.

- **Project Listings:**
  A list of the developer's projects, each with:
  - Project title
  - A brief description
  - Technologies used
  - Links to the project repository (GitHub) or live demo

- **Footer:**
  The same footer as the home page.

---

4. Non-Functional Requirements

- **Responsive Design:**
  The website must be fully responsive and work across a variety of devices (mobile, tablet, desktop).

- **Performance:**
  The website should load quickly with minimal latency.

- **SEO Optimization:**
  The website should follow basic SEO best practices for better visibility in search engines.

- **Cross-Browser Compatibility:**
  The website should work seamlessly across major browsers (Chrome, Firefox, Safari, etc.).

---

5. Technology Stack

- **Frontend:**
  - HTML5, CSS3, and JavaScript
  - CSS Framework: Bootstrap or Tailwind CSS
  - JavaScript Framework (optional): React or Angular

- **Backend (optional):**
  - If implementing a backend, Node.js or Python with Django can be used to provide additional functionality such as a contact form or blog.

- **Hosting:**
  The website can be hosted on GitHub Pages, Netlify, or any similar platform for static sites.

---

6. Wireframe Design

Home Page Wireframe
```

--------------------------------------------------
|       Navbar (Home, Projects)          |
--------------------------------------------------
|          Developer's Photo and Bio        |
|   "Hello! I am Manoj Manjunatha, a Software    |
|    Engineer with a passion for full-stack     |
|    development."                       |
--------------------------------------------------
|        Skills Section (Icons for Technologies)|
--------------------------------------------------
|       Footer (Social Media Links)        |
--------------------------------------------------
```

Projects Page Wireframe
```

--------------------------------------------------
|       Navbar (Home, Projects)          |
--------------------------------------------------
|          Project 1 Title and Description     |
|   "This project is a full-stack web app using   |
|    Angular and Django."                |
--------------------------------------------------
|          Project 2 Title and Description     |
|   "This project is a mobile app built with     |
|    React Native."                     |
--------------------------------------------------
|       Footer (Social Media Links)        |
--------------------------------------------------
```

---

7. Implementation Plan

1. **Setup Development Environment:**
   - Initialize a Git repository.
   - Set up a local environment using VS Code.
   - Install any required dependencies (e.g., Bootstrap, Tailwind CSS).

2. **Build the Home Page:**
   - Create the HTML structure.
   - Style the page with CSS or a CSS framework.
   - Add JavaScript for interactivity if needed.

3. **Build the Projects Page:**
   - Create a new HTML page for the projects section.
   - Design the layout for displaying projects.
   - Include links to GitHub or live demos for each project.

4. **Testing and Debugging:**
   - Test the website for responsiveness and compatibility across browsers.
   - Optimize performance (e.g., minify CSS/JS files).

5. **Deploy the Website:**
   - Host the website on a platform like GitHub Pages or Netlify.

---

8. Conclusion

The portfolio website will be a professional and visually appealing showcase of the developer's skills and projects. By using modern web technologies and ensuring a responsive design, the website will serve as an effective tool for presenting one's capabilities to potential employers or clients.

This document serves as a foundational guide for the portfolio website development process. Further improvements and additions can be made over time as needed.