Mini Project Report

On

Creation of Single Page Portfolio Website

Submitted in partial fulfillment of the requirements of the internship

Submitted by

Sahil Deepak Pardhi

Domain

Web Development Front-end – Self Paced Intern

Associated With

1Stop

July, 2024

ABSTRACT

A portfolio website is a comprehensive platform where individuals may showcase their work, share their professional journey, and provide contact information for potential clients or employers. In the digital age, having an online presence is essential for professionals across various fields. Single-page websites are very popular for their simplicity, ease of navigation, and seamless user experience. This project focuses on using modern web technologies to create a one-page portfolio website that successfully shows a person's professional background. By incorporating crucial features like responsive design, interactive elements, and a clear layout, the website will provide a consistent and engaging user experience across all devices and screen sizes. The end output will be an effective tool for personal identification and career growth in today's competitive digital world.

OBJECTIVE

The goal of this project is to create a single-page portfolio website that successfully highlights a person's abilities, projects, and professional accomplishments. The goal of this project is to design a visually user-friendly, and adaptable web application that can be used as both a digital CV and portfolio. In today's competitive employment market, a strong online presence is essential for professionals who want to stand out. A well-designed portfolio website provides a full platform for professionals to showcase their work, explain their professional path, and provide contact information for potential clients or employers. Unlike standard resumes, a portfolio website provides a dynamic and interactive way to highlight one's skills, leaving an impression on users.

CONTENT

INTRODUCTION

1.1 Introduction

A personal **Portfolio Website**, is an essential tool for professionals looking to showcase their skills, projects, and achievements. Unlike traditional resumes, which offer a static overview of an individual's qualifications, a personal portfolio website provides a dynamic and interactive platform to highlight one's professional background. This type of website allows individuals to present their work in a visually appealing and accessible manner, making it easier for potential clients, employers, and collaborators to understand their capabilities and accomplishments. A personal portfolio website is a digital representation of an individual's professional journey. It includes various sections that detail the person's skills, projects, work experiences, and contact information. This comprehensive approach not only helps in presenting a well-rounded profile but also in building a personal brand that stands out in a competitive job market. With the increasing reliance on digital mediums for recruitment and networking, a personal portfolio website has become a valuable asset for anyone looking to advance their career. This project focuses on developing a single-page personal portfolio website using modern web technologies. The website will consist of three main files: an HTML file, a CSS file, and a JavaScript (JS) file. Each of these files plays a crucial role in building a fully functional and aesthetically pleasing web application.

1.1.1 HTML File - The HTML (Hypertext Markup Language) file serves as the backbone of the website. It is responsible for creating the structure and layout of the web application. The HTML file includes various elements such as headers, paragraphs, images, and links that form the content of the website. By using HTML, we can organize the information in a logical and accessible way, ensuring that users can easily navigate through the different sections of the portfolio.

1.1.2 CSS File - The CSS (Cascading Style Sheets) file is used to enhance the visual appearance of the website. CSS allows us to apply styles to the HTML elements, such as fonts, colors, spacing, and layout. By using CSS, we can create a consistent and attractive design that aligns with the individual's personal brand. Additionally, CSS enables us to implement responsive design techniques, ensuring that the website looks and functions well on various devices, including desktops, tablets, and smartphones.

1.1.2 JavaScript File - The JavaScript (JS) file adds interactivity and functionality to the web application. JavaScript is a versatile programming language that allows us to create dynamic features, such as form validations, image sliders, and interactive menus. By incorporating JavaScript into the portfolio website, we can enhance the user experience and make the application more engaging and user-friendly. JavaScript also enables us to implement advanced features, such as animations and API integrations, which can further enrich the website's functionality.

1.1.2 Responsive Web Designing - A key aspect of this project is ensuring that the portfolio website is fully responsive. Responsive web design is an important method that ensures a website adapts easily to different screen sizes and devices, giving users the best viewing experience possible. This entails using flexible grid layouts, scalable graphics, and CSS media queries to alter the design and layout to the device's specs. We focus on responsive design to guarantee that the website is accessible and user-friendly regardless of whether it is accessed on a desktop, tablet, or smartphone.

This project aims to combine these three core technologies to develop a single-page personal portfolio website that effectively showcases an individual's professional background. By focusing on simplicity, usability, and visual appeal, the website will serve as a powerful tool for personal branding and career advancement. The final product will not only provide a comprehensive overview of the individual's skills and achievements but also offer an engaging and interactive experience for users.

METHODOLOGY

2.1 Introduction

This methodology outlines the comprehensive approach to designing and developing a single-page portfolio website. The website aims to showcase the professional profile of an individual, including their skills, projects, and contact information, in a concise yet effective manner. The methodology covers every phase of the development process, from planning and design to construction and deployment, ensuring a well-structured and user-friendly portfolio.

2.2 Overall Description

The single-page portfolio website is a web application designed to present the user's professional identity in a visually appealing and easily navigable format. The website includes distinct sections—Home, Skills, Projects, and Contact—each serving a specific purpose. The design is responsive, meaning it adapts seamlessly to different screen sizes and devices, ensuring accessibility for all users. The website is built using modern web technologies such as HTML, CSS, and JavaScript, with a focus on performance, security, and usability.

2.3 Software Development Model

The development of the portfolio website follows the Incremental Process Model. This model allows the website to be built and delivered in stages, or increments, with each increment adding specific features or improvements. This iterative approach ensures that the website can be reviewed and refined throughout the development process, allowing for early detection and resolution of issues, as well as the incorporation of user feedback.

2.3.1 Increment 1: Home Page and Navigation

2.3.1.1 Communication

Home Page:

- The home page serves as the introduction to the website. It includes a professional headshot, a brief bio, and a headline that summarizes the user's career or role.
- The navigation bar is designed to be simple and intuitive, allowing users to easily move between sections: Home, Skills, Projects, and Contact.

2.3.1.2 Planning

- The home page should have a clean, modern design that immediately captures the user's attention. The layout should be structured to highlight key information, with the headline and bio being prominently displayed.
- The navigation bar should be fixed at the top of the page, allowing users to access it at any time as they scroll. It should be designed to be responsive, collapsing into a menu icon on smaller screens.
- Potential risks include cross-browser compatibility issues, inconsistent design elements, and performance bottlenecks on mobile devices. These risks will be mitigated through rigorous testing and the use of modern web standards.

2.3.1.3 Modeling

This is one the important phases as the architecture of the system is designed in this phase. Analysis is carried out and depending on the analysis a software model is designed. Different models for developing software are created depending on the requirements gathered in the first phase and the planning done in the second phase.

Analysis: If we follow the Incremental model, one step has to be completed before moving forward to the next phase, so the plan would be to have a different team for each phase and the next team will work on it when the previous team finishes their work.

Design:

- Create a wireframe for the home page that outlines the placement of key elements such as the headline, bio, and navigation bar. The wireframe serves as a blueprint for the page's structure.
- Develop the HTML for the home page, focusing on semantic tags that enhance both accessibility and SEO. Apply CSS to style the elements, ensuring a consistent look and feel across all devices.
- Utilize CSS media queries to adjust the layout and styling for different screen sizes. The navigation bar should transition smoothly into a mobile-friendly format.

2.3.1.4 Construction

The actual coding of the software is done in this phase. This coding is done on the basis of the model designed in the modeling phase. So in this phase software is actually developed and tested.

Code:

- Write the HTML code for the home page, ensuring that it is well-structured and semantic. Use appropriate tags to define the page's sections and elements.
- Implement CSS to style the home page, focusing on typography, colors, and spacing that align with the overall design aesthetic. Ensure that the navigation bar is styled to be both functional and visually appealing.
- If any interactive elements are included on the home page (such as animations or hover effects), use JavaScript to implement these features.There should not be a loading time of more than 2 seconds.

Test:

- Test the home page on different browsers (e.g., Chrome, Firefox, Safari) to ensure consistent rendering and functionality.
- Test the home page on various devices (e.g., smartphones, tablets, desktops) to ensure that it is fully responsive and user-friendly on all screen sizes.
- Use tools like Google Lighthouse to analyze the page's load time, ensuring that it meets performance standards.

Deployment:

- Deploy the initial version of the home page to a staging environment for review and testing. Ensure that the deployment process is smooth and error-free.
- Collect feedback from users and stakeholders, focusing on the usability and design of the home page. Make adjustments based on the feedback received.
- Once all feedback has been addressed and the home page is finalized, deploy it to the production environment.

2.3.2 Increment 2: Skills Section

2.3.2.1 Communication

Skills:

- The Skills section is designed to showcase the user's technical and professional skills in a visually engaging manner. Each skill is represented by an icon and accompanied by a brief description that highlights the user's proficiency in that area.
- This section helps potential employers or clients quickly assess the user's capabilities.

2.3.2.2 Planning

- Plan the content for the Skills section, including the specific skills to be highlighted, the icons to represent each skill, and the descriptions. The design should be consistent with the overall theme of the website.
- The Skills section should be organized into a grid or card layout, making it easy to scan and visually appealing. The use of icons will add a graphical element that enhances the section's readability.
- Ensure that the Skills section is accessible to all users, including those with disabilities. Use appropriate ARIA labels and ensure that the icons and descriptions are easily readable.

2.3.2.3 Modeling

This is one of the important phases as the architecture of the system is designed in this phase. Analysis is carried out and depending on the analysis a software model is designed. Different models for developing software are created depending on the requirements gathered in the first phase and the planning done in the second phase.

Analysis: If we follow the incremental model, one increment has to be completed before moving forward to the next increment.

Design:

- Create a wireframe for the Skills section that outlines the placement of each skill card. The wireframe should include placeholders for icons and descriptions.
- Develop the HTML structure for the Skills section, ensuring that it is organized and easy to modify. Use CSS to style the skill cards, ensuring a consistent look with the rest of the website.
- Ensure that the Skills section is responsive, with the layout adjusting appropriately for different screen sizes.

2.3.2.4 Construction

The actual coding of the skill section is done in this phase. This coding is done on the basis of the model designed in the modeling phase. So in this phase skill section is actually developed and tested.

Code:

- Write the HTML code for the Skills section, creating a container for each skill card. Ensure that the HTML is semantically correct and easy to maintain.
- Implement CSS to style the skill cards, focusing on typography, colors, and spacing. Ensure that the icons are displayed correctly and that the section is visually consistent with the rest of the website.
- If any interactive elements are included in the Skills section (such as hover effects or animations), use JavaScript to implement these features.

Test:

- Test the Skills section on different browsers to ensure consistent rendering and functionality.
- Test the Skills section on various devices to ensure that it is fully responsive and user-friendly on all screen sizes.
- Use accessibility testing tools to ensure that the Skills section meets accessibility standards, including keyboard navigation and screen reader compatibility.

2.3.2.5 Deployment

- Deploy the Skills section to a staging environment for review and testing. Ensure that the deployment process is smooth and error-free.

- Collect feedback from users and stakeholders, focusing on the usability and design of the Skills section. Make adjustments based on the feedback received.
- Once all feedback has been addressed and the Skills section is finalized, deploy it to the production environment.

2.3.3 Increment 3: Project Section

2.3.3.1 Communication

Projects:

- The Projects section is designed to showcase the user's past work, including personal projects, professional work, and contributions to open-source projects. Each project is represented by an image, a title, a brief description, and a link to a live demo or source code repository.
- This section allows potential employers or clients to assess the user's work and gain insights into their skills and experience. This section helps potential employers or clients quickly assess the user's capabilities

2.3.3.2 Planning

- Plan the content for the Projects section, including the specific projects to be highlighted, the images to represent each project, and the descriptions. The design should be consistent with the overall theme of the website.
- The Projects section should be organized into a grid or card layout, making it easy to scan and visually appealing. The use of images will add a graphical element that enhances the section's readability.
- Ensure that the Projects section is accessible to all users, including those with disabilities. Use appropriate labels and ensure that the and descriptions are easily readable.

2.3.3.3 Modeling

This is one of the important phases as the architecture of the system is designed in this phase. Analysis is carried out and depending on the analysis a software model is designed. Different models for developing software are created depending on the requirements gathered in the second phase and the planning done in the third phase.

Analysis: If we follow the incremental model, one increment has to be completed before moving forward to the next increment.

Design:

- Create a wireframe for the Projects section that outlines the placement of each project card. The wireframe should include placeholders for images, titles, descriptions, and links.

- Develop the HTML structure for the Projects section, ensuring that it is organized and easy to modify. Use CSS to style the project cards, ensuring a consistent look with the rest of the website.
- Ensure that the Projects section is responsive, with the layout adjusting appropriately for different screen sizes.

2.3.3.4 Construction

The real coding of the Project Section is going to be done in this phase. This coding is done on the basis of the model designed in the modeling phase. So in this phase the project section is actually developed and tested.

Code:

- Write the HTML code for the Projects section, creating a container for each project card. Ensure that the HTML is semantically correct and easy to maintain.
- Implement CSS to style the project cards, focusing on typography, colors, and spacing. Ensure that the images are displayed correctly and that the section is visually consistent with the rest of the website.
- If any interactive elements are included in the Projects section (such as hover effects or animations), use JavaScript to implement these features.

Test:

- Test the Project section on different browsers to ensure consistent rendering and functionality.
- Test the Project section on various devices to ensure that it is fully responsive and user-friendly on all screen sizes.
- Use accessibility testing tools to ensure that the Skills section meets accessibility standards, including keyboard navigation and screen reader compatibility.

2.3.3.5 Deployment

- Deploy the Project section to a staging environment for review and testing. Ensure that the deployment process is smooth and error-free.
- Collect feedback from users and stakeholders, focusing on the usability and design of the Project section. Make adjustments based on the feedback received.
- Once all feedback has been addressed and the Project section is finalized, deploy it to

the production environment.

2.3.4 Increment 4: Contact Section

2.3.4.1 Communication

Contact:

- The Contact section is a critical component of the portfolio website, providing visitors with a direct way to get in touch with the user. This section includes a contact form for visitors to send messages, along with the user's contact information and links to social media profiles. The goal is to make it easy for potential clients, employers, or collaborators to initiate communication.
- The contact form should be simple yet effective, including fields for the visitor's name, email address, subject, and message. The form must be user-friendly and intuitive to encourage engagement.
- In addition to the form, the Contact section will display the user's direct contact information, such as email address and phone number, along with icons linked to social media profiles (e.g., LinkedIn, GitHub, Twitter), enhancing the user's online presence.

2.3.4.2 Planning

- The Contact section should have a clean and professional design that aligns with the overall aesthetic of the website. The layout should be straightforward, with the form fields arranged in a logical order that guides the user through the process of submitting a message.
- The form should be easy to fill out, with clear labels and placeholders in each field. Error messages should be implemented to guide users in case of incorrect or incomplete entries. The submit button should be prominently displayed, with a confirmation message or page upon successful submission.
- The user's contact details should be displayed in a way that is easy to find but not intrusive. Social media icons should be uniformly styled and aligned, with links opening in a new tab to maintain the user's presence on the portfolio site.
- Plan the visuals for the Contact section, including the background, borders, and spacing. The form should be visually distinct but harmonious with the rest of the page, using complementary colors and fonts.
- Determine the necessary fields for the contact form, focusing on essential information such as the visitor's name, email address, and message. Consider adding an optional

subject field for better categorization of inquiries.

- Choose the contact information to be displayed, such as a professional email address and phone number. Ensure that this information is up-to-date and relevant.
- Decide which social media profiles to include, focusing on platforms that best represent the user's profession.

2.3.4.3 Modeling

This is one of the important phases as the architecture of the system is designed in this phase. Analysis is carried out and depending on the analysis a software model is designed. Different models for developing software are created depending on the requirements gathered in the second phase and the planning done in the third phase.

Analysis: If we follow the incremental model, one increment has to be completed before moving forward to the next increment.

Design:

- Create a detailed wireframe for the Contact section, outlining the placement of the contact form, contact details, and social media links. The wireframe should depict the layout for both desktop and mobile views, ensuring a consistent user experience across devices.
- The wireframe should specify the positioning of each form field, the submit button, and any error or confirmation messages. The form should be centrally aligned on the page, with ample spacing between elements to prevent a cluttered appearance.
- Position the contact details and social media links in a way that complements the form without overwhelming the user. Typically, these elements are placed below the form or alongside it, depending on the overall design.
- Begin by creating the basic HTML structure for the Contact section, using semantic tags to enhance accessibility. This includes setting up the form fields (e.g., <input>, <textarea>, <button>) and the containers for contact information and social media links.
- Apply CSS to style the Contact section, focusing on typography, colors, and layout. The form should be styled to be visually appealing and consistent with the rest of the website. Use CSS media queries to ensure the section is responsive, adjusting the layout and font sizes as needed for different screen sizes.
- Prioritize a mobile-first approach, designing the Contact section to work seamlessly on smaller screens before scaling up to desktop views. Ensure that the form fields and

contact details are easy to interact with on mobile devices.

2.3.4.4 Construction

The real coding of the Contact Section is going to be done in this phase. This coding is done on the basis of the model designed in the modeling phase. So in this phase the contact section is actually developed and tested.

Code:

- Write the HTML code for the Contact section, setting up the form fields with proper labels and placeholders. Ensure that the form includes appropriate attributes (e.g., required, type="email") to enhance usability and prevent errors during submission.

- Implement the CSS for the Contact section, focusing on creating a polished and professional appearance. Use styles to define the layout, spacing, and typography, and ensure that the section is fully responsive. Pay special attention to the design of the submit button and form feedback messages.

- If any interactive features are needed (e.g., form validation, dynamic error messages, or a smooth scroll to the Contact section from other parts of the page), use JavaScript to implement them. Ensure that the form data is validated before submission to improve the user experience and reduce errors.

Test:

- Test the Contact section to ensure that the form works as intended. This includes checking that all form fields accept the correct data types, the submit button triggers the appropriate action, and the form submission is processed correctly (e.g., sending an email or storing data in a database).

- Test the Contact section across different web browsers (e.g., Chrome, Firefox, Safari, Edge) to ensure consistent functionality and appearance.

- Test the Contact section on various devices (e.g., smartphones, tablets, desktops) to ensure it is fully responsive. Ensure that the form is easy to use and the contact information is clearly visible on all screen sizes.

- Conduct accessibility testing to ensure the form and contact details are fully accessible to users with disabilities. This includes testing keyboard navigation, screen reader compatibility, and ensuring the form fields are properly labeled.

- Analyze the performance of the Contact section, ensuring that it loads quickly and does not negatively impact the overall page speed.

2.3.4.5 Deployment

- Deploy the Contact section to a staging environment for initial testing and review. This allows for feedback and testing in a live environment without affecting the production site.
- Collect feedback from users and stakeholders regarding the usability and design of the Contact section. Pay attention to any issues with form submission, layout, or responsiveness. Adjust the section based on this feedback to improve user experience.
- After addressing any issues and refining the section based on feedback, deploy the Contact section to the production environment. Ensure that the deployment is seamless and that all features are functioning as intended.

With each increment building upon the previous one, the single-page portfolio website gradually transforms into a polished and compelling digital showcase. The development journey is just the beginning; ongoing feedback from users will fuel continuous refinement and expansion, ensuring the website remains dynamic and relevant. As personal and professional needs evolve, this portfolio will adapt, incorporating new features and design elements to stay ahead in the digital landscape. By staying attuned to emerging trends, this portfolio will continue to empower the user, reflecting their growth and achievements while inspiring connections and opportunities. Together, we will build a digital presence that not only represents who you are today but also paves the way for future success.

2.4 Requirement

2.4.1 Functional Requirements

2.4.1.1 User Interface:

- The website should have a clean and modern design with easy navigation.

2.4.1.2 Responsiveness:

- The website should be responsive, ensuring a seamless experience across devices.

2.4.1.3 Skills Section:

- Display skills with icons and descriptions.

2.4.1.4 Projects Section:

- Showcase projects with images, descriptions, and links.

2.4.1.5 Contact Section:

- Provide a contact form and display contact details.

2.4.2 Non-functional Requirements

2.4.2.1 Performance:

- The website should load quickly and provide a smooth user experience.

2.4.2.2 Security:

- Ensure the website is secure and free from vulnerabilities.

2.4.2.3 Accessibility:

- The website should adhere to accessibility standards, ensuring it is usable by people with disabilities.

2.4.2.4 Scalability:

- The website should be able to handle an increasing number of visitors and content.

Code and Output

3.1 Code

- **Files:**

3.1.1. index.html:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Sahil - Portfolio</title>
  <link rel="stylesheet" href="styles.css" />
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.1/css/all.min.css"
    integrity="sha512-
DTOQO9RWCH3ppGqcWaEA1BIZOC6xxalwEsw9c2QQeAIftl+Vegovlnee1c9QX4Tctn
WMn13TZye+giMm8e2LwA=="
    crossorigin="anonymous" referrerpolicy="no-referrer" />
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.2/css/all.min.css"
    integrity="sha512-
SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6Ta86w+fsb2TkcmfRyVX3pBnMFc
V7oQPJkl9QevSCWr3W6A=="
    crossorigin="anonymous" referrerpolicy="no-referrer" />
</head>
```

```
<body>
  <header>
    <nav>
      <h1>Sahil</h1>
      <ul>
        <li><a href="#s-container">Home</a></li>
        <li><a href="#skills">Skills</a></li>
        <li><a href="#projects">Projects</a></li>
        <li><a href="#contact">Contact</a></li>
      </ul>
      <button id="btn">
        <i class="fa-solid fa-bars"></i>
      </button>
    </nav>
  </header>

  <section>
    <div class="s-container" id="s-container">
      <div class="content">
        <span class="subtitle"> Hello </span>
        <h1 class="title">
          I'm <span>Sahil Pardhi <br /></span>a Frontend Developer.
        </h1>
        <p class="description">
          I'm Sahil Pardhi, an 22 year-old front-end developer who loves spending the entire
day   learning and
          expermenting new stuffs and technologies.
        </p>
        <div class="buttons">
          <a href="#">About Me</a>
          <a href="#">Portfolio</a>
        </div>
      </div>
      <div class="image">
        <img src="./images/pffp.jpg" alt="pfp" />
      </div>
    </div>

 <div class="skills-section" id="skills">
      <div class="header">
        <h3>Skills</h3>
        <h1>Some Of My Skills:</h1>
      </div>
      <div class="cards-container container ">
        <div class="card">
          <div class="card-header">
            <img src="images/html.png" alt="card-image" id="html-img">
          </div>
          <div class="card-content">
            <h3 id="title-html">HTML</h3>
          </div>
        </div>
```

```
<div class="card">
     <div class="card-header">
      <img src="images/css.png" alt="card-image" id="css-img">
     </div>
     <div class="card-content">
      <h3 id="title-css">CSS</h3>
     </div>
    </div>

    <div class="card">
     <div class="card-header">
      <img src="images/js_5968292.png" alt="card-image" id="js-img">
     </div>
     <div class="card-content">
      <h3 id="title-js">Javascript</h3>
     </div>
    </div>

    <div class="card">
     <div class="card-header">
      <img src="images/bootstrap.png" alt="card-image" id="bs-img">
     </div>
     <div class="card-content">
      <h3 id="title-bs">Bootstrap</h3>
     </div>
    </div>

    <div class="card">
     <div class="card-header">
      <img src="images/c.png" alt="card-image" id="c-img">
     </div>
     <div class="card-content">
      <h3 id="title-c">C</h3>
     </div>
    </div>

    <div class="card">
     <div class="card-header">
      <img src="images/java.png" alt="card-image" id="java-img">
     </div>
     <div class="card-content">
      <h3 id="title-java">Java</h3>
     </div>
    </div>
   </div>
  </div>
```

```
<div class="projects-section" id="projects">
      <div class="header">
        <h3>Projects</h3>
        <h1>Some Of My Work:</h1>
      </div>
      <div class="wrapper">
        <div class="project-card">
          <div class="project-image-container">
            <img class="project-image" src="./images/p1.jpg" alt="" />
          </div>
          <div class="project-content">
            <h3>News Website </h3>
            <div class="check-project">
              <a href="#" target="_blank"><i class="fab fa-github"></i></a>
              <a href="#" target="_blank"><i class="fas fa-external-link-alt"></i></a>
            </div>
          </div>
        </div>
        <div class="project-card">
          <div class="project-image-container">
            <img class="project-image" src="./images/p2.jpg" alt="" />
            <div class="overlay">
              <span>Click Here</span>
            </div>
          </div>
          <div class="project-content">
            <h3>Hostel Website</h3>
            <div class="check-project">
              <a href="#" target="_blank"><i class="fab fa-github"></i></a>
              <a href="#" target="_blank"><i class="fas fa-external-link-alt"></i></a>
            </div>
          </div>
        </div>
      </div>
    </div>
    <div class="contact" id="contact">
      <div class="container">
        <div class="header">
          <h3>Contact</h3>
          <h1>Reach me out via:</h1>
        </div>
        <div class="wrapper">
          <div class="grp">
            <i class="fa-solid fa-envelope"></i>
            <span>sahilpardhi1820@gmail.com</span>
          </div>
          <div class="grp">
            <i class="fa-solid fa-phone"></i>
            <span>+91 8983527780</span>
          </div>
          <div class="footer">
            <p>
              <a href="mailto:sahilpardhi1820@gmail.com">
                Let's Talk <i class="fa-solid fa-envelope"></i>
              </a>
            </p>

</div>
</div>
<div class="contact-form" id="contact">
          <h3>Write to Us</h3>
          <form>
            <label for="name">Enter Name:</label>
            <input type="text" id="name" name="name"><br><br>
            <label for="email">Enter Email:</label>
            <input type="email" id="email" name="email"><br><br>
            <label for="message">Enter Message:</label>
            <textarea id="message" name="message"></textarea><br><br>
            <div class="buton">
              <input class="btn" type="button" value="Send Message">
            </div>
          </form>
        </div>
      </div>

      <!-- back top button  -->
      <a href="#" class="back-top">
        <i class="fa-solid fa-arrow-up"></i>
      </a>
  </section>
</body>
<script src="script.js"></script>

</html>
```

3.1.2. style.css:

```
@import
url("https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap");

* {
  padding: 0;
  margin: 0;
  box-sizing: border-box;
  scroll-behavior: smooth;
  font-family: "Poppins", sans-serif;
}

body {
  background: #111;
  color: white;
}

header nav {
  height: 15vh;
  width: 70%;
  margin: 0 auto;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

header h1 {
  font-size: 45px;
  color: crimson;
  font-weight: 800;
  text-transform: uppercase;
  letter-spacing: 5px;
}
header ul {
  display: flex;
  gap: 5rem;
  list-style: none;
}
header a {
  display: block;
  color: white;
  text-decoration: none;
  border-radius: 0.2rem;
  padding: 0.5rem 1rem;
  border-bottom: 2px solid crimson;
  overflow: hidden;
  position: relative;
}
```

```
header a:hover {
  color: white;
  background: crimson;
}

header ul li:nth-child(1) a {
  background: crimson;
  color: white;
}

.s-container {
  min-height: 80vh;
  width: 70%;
  margin: auto;

  display: flex;
  align-items: center;
  gap: 3rem;
}

#btn {
  all: unset;
  cursor: pointer;
  transition: 0.5s ease-out;
  font-size: 30px;

  display: none;
}

#btn:hover {
  transform: rotate(180deg);
  color: crimson;
}

main {
  padding-block: 20px;
  margin-top: 90px;
}

.container {
  max-width: 1180px;
  margin-inline: auto;
  overflow: hidden;
}
```

```
.cards-container {
  display: flex;
  align-items: center;
  justify-content: space-between;
  flex-wrap: wrap;
  row-gap: 20px;
  align-items: start;
}

.card {
  width: 360px;
  min-height: 200px;
  border-radius: 5px;
  background-color: #fff;
  cursor: pointer;
  overflow: hidden;
  transition: all 0.3s ease;
}

.card:hover {
  transform: translateY(-4px);
  box-shadow: 1px 1px 8px black;
  background-color: #f9fdff;
}

.card-header {
  display: flex;
  justify-content: center;
  align-items: center;
}

.card-header img {
  margin-top: 25px;
  margin-bottom: 7px;
  height: 200px;
}

.card-content {
  display: flex;
  justify-content: center;
  padding: 12px;
  color: #111;
  background-color: burlywood;
}
```

```
.card-content:hover {
  background-color: crimson;
}

#title-html:hover {
  color: orange;
}

#title-css:hover {
  color: darkblue;
}

#title-js:hover {
  color: yellow;
}

#title-bs:hover {
  color: purple;
}

#title-c:hover {
  color: skyblue;
}

#title-java:hover {
  color: red;
}

.header h3 {
  font-size: 25px;
  color: crimson;
  font-weight: 200;
  text-align: center;
  margin-bottom: 1px;
}

.header h1 {
  font-size: 50px;
  position: relative;
  margin-bottom: 1rem;
  text-align: center;
  text-transform: uppercase;
  letter-spacing: 3px;
}

.image {
  width: 320px;
  height: 320px;
}
```

```
.image img {
  width: 100%;
  height: 320px;
  border-radius: 50%;
  object-fit: cover;
}

.content {
  display: flex;
  flex-direction: column;
  justify-content: center;
  width: 65%;
}

.title {
  font-size: 3.5rem;
  font-weight: 400;
  line-height: 3.3rem;
  margin-bottom: 1rem;
}

.title span {
  font-weight: 700;
}

.subtitle {
  letter-spacing: 2px;
  font-weight: 600;
  margin-bottom: 0.5rem;
  font-size: 1.5rem;
}

.description {
  line-height: 1.6rem;
  margin-bottom: 2rem;
  color: #dfdfdf;
  width: 85%;
}

.buttons {
  display: flex;
  gap: 1rem;
}
```

```
.buttons a {
  font-size: 1rem;
  font-weight: 600;
  letter-spacing: 2px;
  color: black;

  padding: 0.8rem 1.6rem;
  outline: none;
  border: none;
  border-radius: 10px;

  transition: 0.3s ease-out;
  cursor: pointer;
  box-shadow: 3px 3px 0 rgba(0, 0, 0, 0.4);
}

.buttons a:hover {
  transform: scale(1.05);
  color: white;
}

.buttons a {
  text-decoration: none;
  background-color: crimson;
  color: white;
}

.buttons a:nth-child(2) {
  text-decoration: none;
  background-color: white;
  color: #111;
}

.buttons button:hover {
  background-color: crimson;
}

.buttons a:nth-child(2):hover {
  background-color: crimson;
  color: white;
}

.skills-section {
  margin-top: 80px;
}
```

```
.projects-section {
  margin-top: 80px;
  margin-bottom: 15rem;
}

.projects-section .wrapper {
  display: flex;
  justify-content: center;
  flex-wrap: wrap;
  gap: 4rem;
  max-width: 1000px;
  width: 80%;
  height: 100%;
  margin: 30px auto 0 auto;
}

.project-card {
  background: #0d0d0d;
  border-radius: 1rem;
  max-width: 450px;
}

.project-image-container {
  overflow: hidden;
  width: 100%;
  border-radius: 10px 10px 0px 0px;
  height: 200px;
  object-fit: cover;
}

.project-image {
  width: 100%;
  object-fit: cover;
  object-position: center;
  height: 100%;
  transition: 0.2s ease-in-out;
}

.project-image:hover {
  transform: scale(1.05);
}
.project-content {
  border-radius: 0px 0px 10px 10px;
  padding: 1.5rem 1rem;
  background-color: #222;
  display: flex;
  justify-content: space-between;
  padding-inline: 3rem;
}
```

```
.project-content > p {
  color: #ffffff;
  font-size: 24px;
  font-weight: 400;
  line-height: 30px;
  overflow: hidden;
  white-space: pre-wrap;
}

.check-project {
  display: flex;
  gap: 1rem;
}

.check-project a {
  color: white;
  font-size: 20px;
  cursor: pointer;
  transition: 0.2s ease-in-out;
}

.check-project a:hover {
  color: crimson;
}

.contact {
  position: relative;
  display: flex;
  font-size: 2em;
  min-height: 100vh;
  padding-block: 2rem;
}

.contact .container {
  width: 100%;
  margin: auto;
  display: flex;
  flex-direction: column;
  align-items: center;
}

.contact .wrapper {
  font-size: 30px;
}
```

```
.contact .wrapper .grp {
  text-align: center;
  display: flex;
  justify-content: center;
  align-items: center;
  gap: 1rem;
}

.contact .grp:not(:last-child) {
  margin-bottom: 1.5rem;
}

.contact .grp i {
  font-size: 25px;
}

.contact .footer {
  display: flex;
  flex-direction: column;
  align-items: center;
  text-align: center;
  margin-top: 3rem;
  margin-bottom: 2px;
}

.contact .footer p {
  font-size: 40px;
  font-weight: 200;
  color: white;
  margin-bottom: 2rem;
  position: relative;
}

.contact .footer a {
  text-decoration: none;
  background: crimson;
  color: #eee;
  font-weight: bold;
  font-size: 20px;
  border-radius: 4px;
  padding: 1rem 2rem;
}

.contact .footer a:hover {
  background: rgb(216, 71, 100);
}
/* contact  */
```

```
#contact {
  max-width: 900px;
  margin: 40px auto;
  padding: 20px;
}

.contact-info {
  margin-bottom: 20px;
}

.contact-info h3 {
  margin-top: 0;
}

.contact-info p {
  margin-bottom: 10px;
}

.contact-info span {
  font-weight: bold;
}

.contact-form {
  width: 70%;
  margin-top: 20px;
}

.contact-form h3 {
  font-size: 25px;
  color: crimson;
  font-weight: 200;
  text-align: center;
  margin-bottom: 1px;
}

.contact-form label {
  font-size: 30px;
  display: block;
  margin-bottom: 5px;
}

.contact-form input,
.contact-form textarea {
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
  width: 100%;
  margin-bottom: 5px;
}
```

```
.contact-form input[type="button"] {
  font-weight: 200;
  color: white;
  margin-bottom: 2rem;
  width: 40%;
  box-shadow: 3px 3px 0 rgba(0, 0, 0, 0.4);

  background-color: #dc143c;

  font-weight: bold;
  font-size: 20px;
  padding: 1rem 2rem;
  border-radius: 5px;
  border: none;
  cursor: pointer;
}

.contact-form input[type="button"]:hover {
  background-color: #d15971;
  color: white;
}

.buton {
  display: flex;
  justify-content: center;
}

.back-top {
  text-decoration: none;
  font-size: 2rem;
  color: #222;
  background: crimson;
  border-radius: 50%;
  width: 5rem;
  height: 5rem;
  display: flex;
  justify-content: center;
  align-items: center;
  position: fixed;
  bottom: 2%;
  right: 2%;
  box-shadow: 3px 3px 5px rgba(0, 0, 0, 0.859);
  transition: 0.4s ease-out;
  opacity: 0;
  pointer-events: none;
}
```

```
.back-top:hover {
  transform: translateY(-20px);
}

.back-top.active {
  opacity: 1;
  pointer-events: auto;
}

/* Responsivness  */

@media only screen and (max-width: 1100px) {
  .s-container,
  header nav {
    width: 80%;
  }
}

@media only screen and (max-width: 950px) {
  #btn {
    display: block;
  }

  header ul {
    position: absolute;
    right: 5vh;
    top: 15vh;

    width: 300px;
    min-height: 400px;
    border-top-right-radius: 1rem;
    border-bottom-left-radius: 1rem;

    background: #0000007f;
    backdrop-filter: blur(5px);

    flex-direction: column;
    justify-content: center;
    align-items: center;
    display: none;
  }

  .container {
    justify-content: center;
    align-items: center;
    column-gap: 10px;
  }
  header ul.show {
    display: flex;
  }
```

```
header nav {
    width: 90%;
  }

  .s-container {
    width: 90%;
    flex-direction: column-reverse;
    gap: 1rem;
    padding-block: 3rem;
    padding-bottom: 2rem;
  }
  .content {
    width: 90%;
    text-align: center;
  }

  .title {
    font-size: 3rem;
  }

  .description {
    width: 100%;
  }

  .buttons {
    margin: auto;
  }

  .image {
    width: 250px;
    height: 250px;
  }
  .image img {
    height: 250px;
  }
}

@media only screen and (max-width: 768px) {
  .contact .header h1 {
    font-size: 30px;
  }
  .contact .header h3 {
    font-size: 20px;
    margin-bottom: -0.5rem;
  }
  .contact .grp i,
  .contact .grp span {
    font-size: 25px;
  }
  .footer p {
    font-size: 27px !important;
  }

  .footer a {
    font-size: 15px !important;
    padding: -0.7rem;
  }
}

@media only screen and (max-width: 550px) {
  header ul {
    width: 80%;
    left: 50%;
    transform: translateX(-50%);
  }

  .icons {
    display: none;
  }
  .content {
    width: 100%;
  }
  .title {
    font-size: 2.7rem;
  }
  .image {
    width: 220px;
    height: 220px;
  }
  .image img {
    height: 220px;
  }

  .contact .grp i {
    margin-right: 0 !important;
  }
}
```

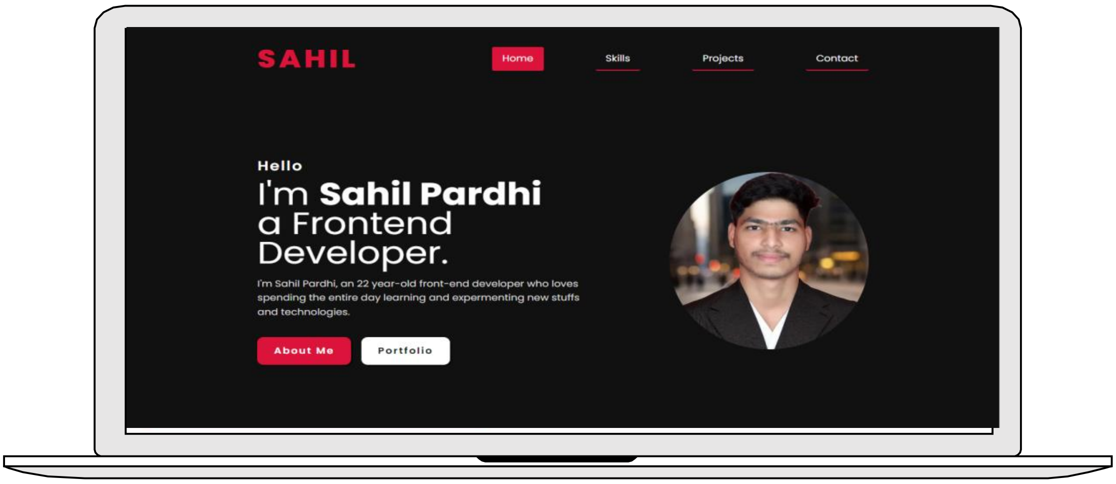
3.1.3. script.js:

```
const btn = document.getElementById("btn");
btn.addEventListener("click", () => {
  document.querySelector("header ul").classList.toggle("show");
});

const toTop = document.querySelector(".back-top");

window.addEventListener("scroll", () => {
  if (pageYOffset > 75) {
    toTop.classList.add("active");
  } else {
    toTop.classList.remove("active");
  }
});
```

3.2 Outputs

1. Desktop Design –

1.1 Home Section Design



Hello

I'm **Sahil Pardhi** a Frontend Developer.

I'm Sahil Pardhi, an 22 year-old front-end developer who loves spending the entire day learning and expermenting new stuffs and technologies.

Fig. 3.2.1 Home Section Design

1.2 Skill Section Design

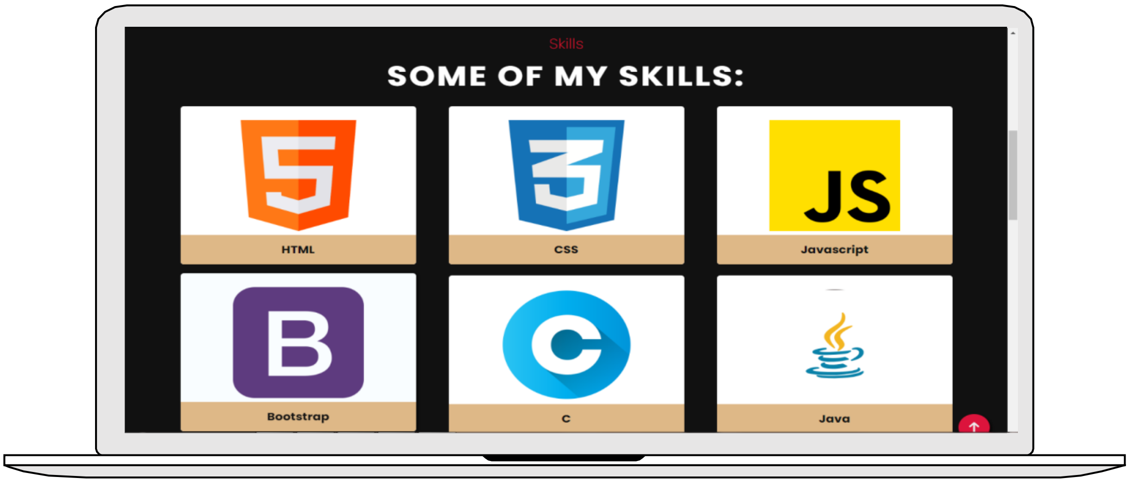
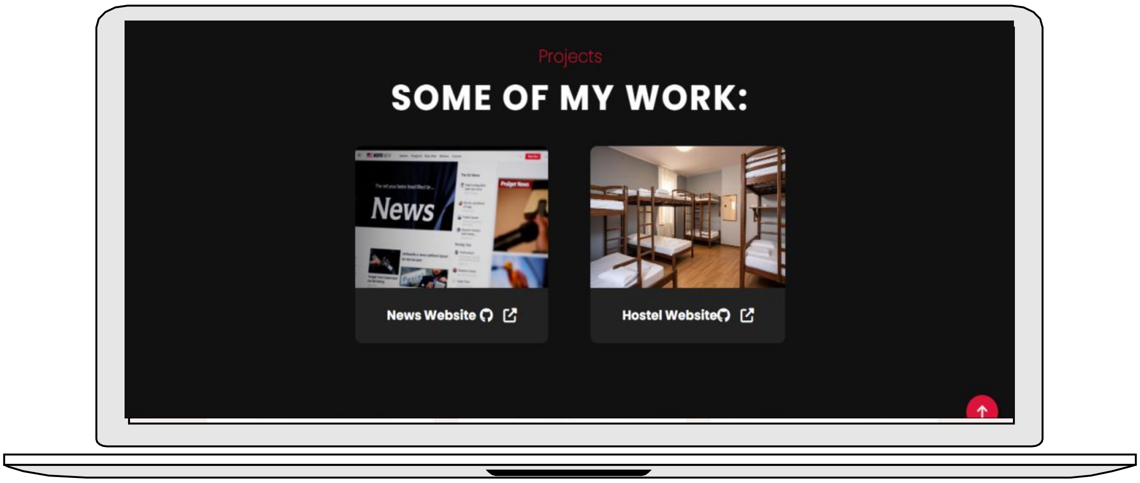


Fig. 3.2.2 Skill Section Design

1.3 Project Section Design

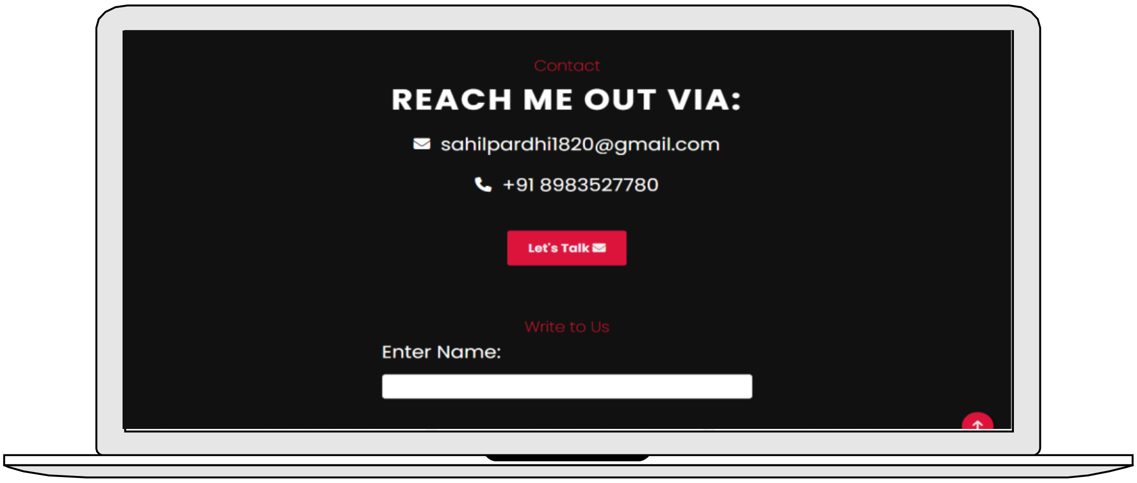


Projects

SOME OF MY WORK:

Fig. 3.2.3 Project Section Design

1.4 Contact Section Design



Contact

REACH ME OUT VIA:

sahilpardhi1820@gmail.com

+91 8983527780

Write to Us

Enter Name:

Fig. 3.2.4 Contact Section Design

Conclusion

4.1 Conclusion

Creating a single-page portfolio website is a dynamic process that goes beyond simply building a web page; it's about crafting a digital narrative that authentically represents the individual behind it. From the initial planning and design stages to the final deployment, every increment contributes to a cohesive and visually appealing platform that showcases skills, projects, and personal brand. The attention to detail in each section—Home, Skills, Projects, and Contact—ensures that the portfolio is not only aesthetically pleasing but also functional and user-friendly. By incorporating responsive design, clear content, and intuitive navigation, the website effectively communicates the user's story and capabilities to potential clients, employers, or collaborators.

However, the creation of this portfolio is not a one-time endeavor. As the user grows professionally and personally, the website will continue to evolve, reflecting new experiences, skills, and accomplishments. Feedback from users and viewers will play a crucial role in this ongoing process, guiding updates and improvements that keep the portfolio relevant and impactful. In the rapidly changing digital landscape, this portfolio website will serve as a living document of the user's journey, continuously adapting to meet new challenges and seize new opportunities. Ultimately, this project is more than just a technical exercise; it's a commitment to personal growth and professional development, embodied in a digital form that can inspire and connect with others.