Portfolio Website Documentation
Prashant Koirala

A Comprehensive Guide to the Codebase and Features

Author: Prashant Koirala

Date: August 10, 2025

Submitted for: Web Development Presentation

Team: Prashant Koirala (Project Leader)

Contents

1 Introduction

This document provides a detailed, beginner-friendly explanation of my portfolio website, designed to showcase my skills as a full-stack developer and designer. The website, built with the assistance of Cursor AI, features a modern, interactive design inspired by award-winning websites, such as those recognized by Awwwards. It includes two main pages: the homepage (`index.html`) and the contact page (`contact.html`), with smooth animations, responsive design, and a professional aesthetic.

1.1 Purpose of the Document

The goal of this document is to prepare me for a presentation on August 11, 2025, by explaining the entire codebase in detail. It covers the project's structure, technologies, file-by-file breakdown, key features, deployment setup, and potential enhancements. The document is designed to be 20–30 pages, ensuring a thorough explanation suitable for beginners with little to no web development knowledge. Code snippets, screenshot placeholders, and simple language make the content accessible.

1.2 Project Overview

The portfolio website serves as a digital showcase of my expertise in frontend and backend development, UI/UX design, and web applications. Key features include:

- A bold hero section with animated text and cycling images.
- A featured work section with 3D parallax effects to display projects.
- Service cards highlighting my skills in development and design.
- A contact page with an interactive image trail effect and a project inquiry form.
- Smooth page transitions and scrolling for a polished user experience.

The site is built using HTML for structure, CSS for styling, JavaScript for interactivity, GSAP for animations, Lenis for smooth scrolling, Vite as the build tool, and Vercel for deployment.

1.3 Target Audience

This document is written for beginners, assuming minimal prior knowledge of web development. Technical terms like HTML (the structure of web pages), CSS (the visual styling), and JavaScript (the interactivity) are explained in simple terms. Code snippets are annotated, and screenshot placeholders are included to visualize key features.

2 Project Architecture

The portfolio is a static website with two HTML pages, styled with modular CSS files and enhanced with JavaScript for interactivity. This section outlines the file structure and technologies used.

2.1 File Structure

The project is organized into folders for HTML, CSS, JavaScript, and images, with configuration files at the root. Below is the file structure:

```
portfolio/
 css/
     globals.css
     home.css
     menu.css
     transition.css
     about.css
     project.css
     contact.css
     footer.css
     fonts.css
 js/
     transition.js
     lenis-scroll.js
     menu.js
     hero.js
     featured-work.js
     services.js
     footer.js
     about.js
     contact.js
 images/
     global/
         site-icon.png
     hero/
     services/
     work-items/
         work-item-1.jpg
         work-item-2.jpg
         ...
 index.html
 contact.html
 package.json
 vercel.json
 vite.config.js
```

2.2 Technologies Used

- HTML (HyperText Markup Language): Defines the structure of web pages, like a blueprint for content.
- CSS (Cascading Style Sheets): Controls the visual appearance, such as colors, fonts, and layouts.
- JavaScript: Adds interactivity, like animations and form handling.
- GSAP (GreenSock Animation Platform): A library for creating smooth, complex animations.
- Lenis: A library for smooth scrolling, enhancing the user experience.
- Vite: A modern build tool for fast development and production builds.

- Vercel: A platform for deploying the website with clean URLs.

3 Codebase Explanation

This section provides a detailed breakdown of each file, including its purpose, structure, and functionality, with code snippets and screenshot placeholders.

3.1 HTML Files

HTML files define the content and structure of the website's pages.

3.1.1 *index.html* (Homepage)

The homepage is the main entry point, showcasing my skills and projects. It includes:

- Head Section: Metadata, title, favicon, and CSS links for styling.
- Body Section:
  - Page Transition: Five `div` elements for animated transitions between pages.
  - Navigation: A logo and menu toggle button, with an overlay for navigation links.
  - Hero Section: Displays my name with large text and a cycling image.
  - About Hero: Introduces my skills with a bio and portrait image.
  - Featured Work: Lists projects with images and scroll-based animations.
  - Services Header: Highlights my expertise with a profile icon.
  - Services: Four cards for frontend, backend, UI/UX, and web applications.
  - Contact CTA: A gradient button linking to the contact page.
  - Footer: Links, social media, and a particle explosion effect.
- Scripts: JavaScript files for animations and interactivity.

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Prashant Koirala | Portfolio</title>
    <link rel="icon" href="/images/global/site-icon.png" />
    <link rel="stylesheet" href="/css/transition.css" />
    <link rel="stylesheet" href="/css/fonts.css" />
    <link rel="stylesheet" href="/css/globals.css" />
    <link rel="stylesheet" href="/css/menu.css" />
    <link rel="stylesheet" href="/css/home.css" />
  </head>
  <body>
    <div class="transition">
      <div class="transition-overlay overlay-1"></div>
      <div class="transition-overlay overlay-2"></div>
      <div class="transition-overlay overlay-3"></div>
      <div class="transition-overlay overlay-4"></div>
      <div class="transition-overlay overlay-5"></div>
    </div>
    <div class="page home-page">
      <nav>
        <div class="logo">
          <p class="mn"><a href="/">P   K</a></p>
        </div>
        <div class="menu-toggle-btn">
          <p class="mn open-label">Menu</p>
          <p class="mn close-label">Close</p>
        </div>
      </nav>
      <section class="hero">
        <h1>Prashant</h1>
        <h1>Koirala</h1>
        <div class="hero-footer">
          <p class="mn"><a href="https://drive.google.com/...">Fetch // Resume</a></p>
          <p class="mn">Showcase Mode: ON</p>
        </div>
      </section>
      <!-- Other sections: About Hero, Featured Work, Services, Contact CTA, Footer -->
    </div>
    <script type="module" src="/js/transition.js"></script>
    <script type="module" src="/js/lenis-scroll.js"></script>
    <script type="module" src="/js/menu.js"></script>
    <script type="module" src="/js/hero.js"></script>
    <script type="module" src="/js/featured-work.js"></script>
    <script type="module" src="/js/services.js"></script>
    <script type="module" src="/js/footer.js"></script>
    <script type="module" src="/js/about.js"></script>
  </body>
</html>
```

Listing 1: Excerpt from index.html

Screenshot Placeholder: Insert a screenshot of the homepage here.

3.1.2 *contact.html* (Contact Page)

The contact page allows users to submit project inquiries. It includes:

- Head Section: Metadata, title, and specific CSS files.
- Body Section:
  - Page Transition: Same as homepage for consistency.
  - Navigation: Simplified with only the logo.
  - Contact Section: Contact info and a form with fields for name, email, phone, project type, and details.
  - Floating Elements: Background animations for visual appeal.

- Scripts: JavaScript for form handling and image trail effect.

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Contact | Prashant Koirala | Portfolio</title>
    <link rel="stylesheet" href="/css/transition.css" />
    <link rel="stylesheet" href="/css/fonts.css" />
    <link rel="stylesheet" href="/css/globals.css" />
    <link rel="stylesheet" href="/css/menu.css" />
    <link rel="stylesheet" href="/css/contact.css" />
  </head>
  <body>
    <div class="transition">
      <div class="transition-overlay overlay-1"></div>
      <!-- Other overlays -->
    </div>
    <div class="page contact-page">
      <nav>
        <div class="logo">
          <p class="mn"><a href="/">P   K</a></p>
        </div>
      </nav>
      <section class="contact trail-container">
        <div class="contact-left">
          <h1>Let's Connect</h1>
          <p>Got a project idea? Drop me a line...</p>
          <div class="contact-info">
            <p><a href="mailto:prashantkoirala465@gmail.com">prashantkoirala465@gmail.com</a></p>
            <p><a href="https://x.com/arkynox_">@arkynox_</a></p>
          </div>
        </div>
        <div class="contact-form-container">
          <form class="contact-form" id="contactForm">
            <input type="text" id="firstName" name="firstName" placeholder="Your first name" required>
            <input type="text" id="lastName" name="lastName" placeholder="Your last name" required>
            <input type="email" id="email" name="email" placeholder="Your email address" required>
            <input type="tel" id="phone" name="phone" placeholder="Your phone number">
            <select id="projectType" name="projectType" required>
              <option value="" disabled selected>Select project type</option>
              <option value="website">Website Development</option>
              <option value="app">Web Application</option>
            </select>
            <textarea id="projectDetails" name="projectDetails" placeholder="Tell me about your project..." required></textarea>
            <button type="submit" class="submit-btn">Send Message</button>
            <div class="success-message" style="display: none;">Thank you! I'll get back to you soon.</div>
          </form>
        </div>
        <div class="floating-elements"></div>
      </section>
    </div>
    <script type="module" src="/js/transition.js"></script>
    <script type="module" src="/js/contact.js"></script>
  </body>
</html>
```

Listing 2: Excerpt from contact.html

Screenshot Placeholder: Insert a screenshot of the contact page here.

3.2 CSS Files

CSS files control the website's visual appearance, including colors, layouts, and typography.

3.2.1 *globals.css*

This file sets global styles for the entire site:

- CSS Variables: Defines reusable colors (e.g., `--bg` for background, `--fg` for text, `--accent1` to `--accent4` for highlights).
- Reset Styles: Removes default margins and padding for consistency.
- Typography: Sets fonts (`rader`, `formula-narrow`, `supply-mono`) and sizes for headers and paragraphs.
- Contact CTA: Styles a full-screen section with an animated gradient button.

```
:root {
  --bg: #edf1e8;
  --fg: #141414;
  --accent1: #ed6a5a;
  --accent2: #f4a261;
  --accent3: #9bc1bc;
  --accent4: #5ca4a9;
}
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
body {
  background-color: var(--bg);
  color: var(--fg);
  overflow-x: hidden;
}
.contact-cta .contact-button {
  width: 60%;
  height: 300px;
  border: 0.75em solid #000;
  border-radius: 20em;
  background: linear-gradient(45deg, var(--accent1), var(--accent2), var(--
    accent3), var(--accent4));
  animation: gradientMove 10s ease-in-out infinite;
}
@keyframes gradientMove {
  0% { background-position: 0% 50%; }
  50% { background-position: 100% 50%; }
  100% { background-position: 0% 50%; }
}
@media (max-width: 1000px) {
  .contact-cta .contact-button {
    width: 80%;
    height: 200px;
  }
}
```

Listing 3: Excerpt from globals.css

Screenshot Placeholder: Insert a screenshot of the contact CTA button here.

3.2.2 *home.css*

Styles homepage-specific sections:

- Hero: Large, centered text for my name and a footer with resume and showcase links.
- Hero Image: A scaled and rotated image that animates on scroll.
- Featured Work: Horizontal scrolling titles and 3D image cards with a progress indicator.
- Services Header: Centers a profile icon and text.
- Services: Four cards with different accent colors for each skill.

```
.hero {
  width: 100vw;
  height: 100svh;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
}
.hero h1 {
  font-size: 18vw;
  font-family: "rader", sans-serif;
  text-transform: uppercase;
  font-style: italic;
}
.hero-img-holder .hero-img {
  width: 30%;
  transform: translateY(-110%) scale(0.25) rotate(-15deg);
  border: 0.3em solid var(--fg);
}
.featured-titles {
  width: 400%;
  display: flex;
}
.service-card {
  background-color: var(--accent1);
  padding: 2em;
}
@media (max-width: 1000px) {
  .hero h1 { font-size: 12vw; }
  .service-card { width: 100%; }
}
```

Listing 4: Excerpt from home.css

Screenshot Placeholder: Insert a screenshot of the hero section here.

3.2.3 *menu.css*

Styles the navigation:

- Nav Bar: Fixed at the top with a logo and toggle button.
- Nav Overlay: Full-screen menu with animated links and social media footer.

```
nav {
  position: fixed;
  top: 0;
  width: 100vw;
  display: flex;
  justify-content: space-between;
  padding: 1em;
}
.nav-overlay {
  position: fixed;
  width: 100vw;
  height: 100svh;
  background-color: var(--bg2);
  opacity: 0;
  pointer-events: none;
}
.nav-overlay-item a.active {
  color: var(--accent1);
}
@media (max-width: 1000px) {
  .nav-overlay-item { font-size: 8vw; }
}
```

Listing 5: Excerpt from menu.css

Screenshot Placeholder: Insert a screenshot of the nav overlay here.

3.2.4 *transition.css*

Styles page transition overlays:

- Five full-screen `div` elements with different colors, animated to cover or reveal the page.

```
.transition {
  position: fixed;
  width: 100vw;
  height: 100svh;
  pointer-events: none;
}
.transition-overlay {
  position: absolute;
  width: 100%;
  height: 100%;
  transform: scaleY(1);
}
.transition-overlay.overlay-1 { background-color: var(--fg); }
.transition-overlay.overlay-2 { background-color: var(--accent1); }
.transition-overlay.overlay-3 { background-color: var(--accent2); }
.transition-overlay.overlay-4 { background-color: var(--accent3); }
.transition-overlay.overlay-5 { background-color: var(--accent4); }
```

Listing 6: Excerpt from transition.css

Screenshot Placeholder: Insert a screenshot of the transition effect here.

3.2.5 *work.css*, *about.css*, *contact.css*, *footer.css*, *fonts.css*

- `work.css`: Styles a potential "Work" page (currently unused, with commented-out nav links).
- `about.css`: Styles the about section (used for `.about-hero` on homepage).
- `contact.css`: Styles the contact page's form and layout.
- `footer.css`: Styles the footer's links and particle container.
- `fonts.css`: Defines custom fonts (`rader`, `formula-narrow`, `supply-mono`).

Note: These files are modular, supporting future expansion (e.g., a dedicated "Work" page).

3.3 JavaScript Files

JavaScript files add interactivity and animations using GSAP and Lenis.

3.3.1 *transition.js*

Manages page transitions:

- Reveal Transition: Shrinks overlays to reveal the page on load.
- Animate Transition: Expands overlays to cover the page before navigating.

```
import gsap from "gsap";
document.addEventListener("DOMContentLoaded", () => {
  gsap.to(".transition-overlay", {
    scaleY: 0,
    duration: 0.6,
    stagger: -0.1,
    ease: "power2.inOut"
  });
});
const links = document.querySelectorAll("a");
links.forEach(link => {
  link.addEventListener("click", (e) => {
    if (!link.href.includes("http") && !link.href.includes("mailto") && !link.href.includes("tel")) {
      e.preventDefault();
      gsap.to(".transition-overlay", {
        scaleY: 1,
        duration: 0.6,
        stagger: 0.1,
        ease: "power2.inOut",
        onComplete: () => window.location = link.href
      });
    }
  });
});
```

Listing 7: Excerpt from transition.js

Screenshot Placeholder: Insert a screenshot of the transition animation here.

3.3.2 *lenis-scroll.js*

Implements smooth scrolling:

- Uses Lenis with different settings for mobile and desktop.
- Integrates with GSAP's ScrollTrigger for scroll-based animations.

```
import Lenis from "lenis";
import gsap from "gsap";
import { ScrollTrigger } from "gsap/ScrollTrigger";
gsap.registerPlugin(ScrollTrigger);
let lenis;
function initLenis() {
  lenis = new Lenis({
    duration: window.innerWidth > 900 ? 1.2 : 0.8,
    easing: (t) => Math.min(1, 1.001 - Math.pow(2, -10 * t)),
    smooth: true
  });
  lenis.on("scroll", ScrollTrigger.update);
  gsap.ticker.add((time) => lenis.raf(time * 1000));
}
initLenis();
window.addEventListener("resize", initLenis);
```

Listing 8: Excerpt from lenis-scroll.js

3.3.3 *menu.js*

Handles menu animations:

- Toggles the navigation overlay with smooth animations.
- Locks scrolling when the menu is open.

```
import gsap from "gsap";
const menuToggleBtn = document.querySelector(".menu-toggle-btn");
let isMenuOpen = false;
menuToggleBtn.addEventListener("click", () => {
  if (!isMenuOpen) {
    gsap.to(".nav-overlay", { opacity: 1, duration: 0.3, pointerEvents: "auto" });
    gsap.to(".nav-overlay-item", { y: 0, opacity: 1, duration: 0.5, stagger: 0.1 });
    document.body.style.position = "fixed";
    isMenuOpen = true;
  } else {
    gsap.to(".nav-overlay", { opacity: 0, duration: 0.3, pointerEvents: "none" });
    gsap.to(".nav-overlay-item", { y: 50, opacity: 0, duration: 0.5 });
    document.body.style.position = "";
    isMenuOpen = false;
  }
});
```

Listing 9: Excerpt from menu.js

Screenshot Placeholder: Insert a screenshot of the menu animation here.

3.3.4 *hero.js*

Animates the hero image:

- Cycles through 10 images every 250ms.
- Animates position, scale, and rotation on scroll.

```
import gsap from "gsap";
import { ScrollTrigger } from "gsap/ScrollTrigger";
gsap.registerPlugin(ScrollTrigger);
const heroImg = document.querySelector(".hero-img img");
let currentImageIndex = 1;
setInterval(() => {
  currentImageIndex = currentImageIndex % 10 + 1;
  heroImg.src = `/images/work-items/work-item-${currentImageIndex}.jpg`;
}, 250);
gsap.to(".hero-img", {
  y: "0%",
  scale: 1,
  rotate: 0,
  scrollTrigger: {
    trigger: ".hero-img-holder",
    start: "top bottom",
    end: "bottom top",
    scrub: true
  }
});
```

Listing 10: Excerpt from hero.js

Screenshot Placeholder: Insert a screenshot of the hero image animation here.

3.3.5 *featured-work.js*

Animates the featured work section:

- Creates indicators and image cards.

- Animates titles and images with 3D effects based on scroll progress.

```
import gsap from "gsap";
import { ScrollTrigger } from "gsap/ScrollTrigger";
gsap.registerPlugin(ScrollTrigger);
const indicator = document.querySelector(".featured-work-indicator");
for (let i = 1; i <= 5; i++) {
  const sectionNumber = document.createElement("p");
  sectionNumber.textContent = `0${i}`;
  indicator.appendChild(sectionNumber);
}
gsap.to(".featured-titles", {
  x: "-75%",
  scrollTrigger: {
    trigger: ".featured-work",
    start: "top top",
    end: "bottom bottom",
    scrub: true,
    pin: true
  }
});
```

Listing 11: Excerpt from featured-work.js

Screenshot Placeholder: Insert a screenshot of the featured work section here.

3.3.6 *services.js*

Animates service cards:

- Pins cards during scroll and animates their inner content.

```
import gsap from "gsap";
import { ScrollTrigger } from "gsap/ScrollTrigger";
gsap.registerPlugin(ScrollTrigger);
document.querySelectorAll(".service-card:not(:last-child)").forEach(card => {
  gsap.to(card.querySelector(".service-card-inner"), {
    y: "-100%",
    scrollTrigger: {
      trigger: card,
      start: "top top",
      end: "bottom top",
      scrub: true,
      pin: true
    }
  });
});
```

Listing 12: Excerpt from services.js

Screenshot Placeholder: Insert a screenshot of the service cards here.

3.3.7 *footer.js*

Creates a particle explosion in the footer:

- Generates 10 image particles that animate when the footer enters the viewport.

```
import gsap from "gsap";
import { ScrollTrigger } from "gsap/ScrollTrigger";
gsap.registerPlugin(ScrollTrigger);
class Particle {
  constructor(imgSrc, container) {
    this.element = document.createElement("img");
    this.element.src = imgSrc;
    container.appendChild(this.element);
  }
}
const container = document.querySelector(".explosion-container");
for (let i = 1; i <= 10; i++) {
  new Particle(`/images/work-items/work-item-${i}.jpg`, container);
}
```

Listing 13: Excerpt from footer.js

Screenshot Placeholder: Insert a screenshot of the footer explosion here.

3.3.8 *about.js*

Animates the about section:

- Animates the portrait and (unused) tags/stats on scroll.

```
import gsap from "gsap";
import { ScrollTrigger } from "gsap/ScrollTrigger";
gsap.registerPlugin(ScrollTrigger);
gsap.to(".about-hero-portrait", {
  x: "20%",
  rotate: 5,
  scrollTrigger: {
    trigger: ".about-hero",
    start: "top bottom",
    end: "bottom top",
    scrub: true
  }
});
```

Listing 14: Excerpt from about.js

Screenshot Placeholder: Insert a screenshot of the about section here.

3.3.9 *contact.js*

Manages the contact page:

- Image Trail: Images follow the cursor on desktop.
- Floating Elements: Animated background elements.
- Form Handling: Simulates submission with a success message.

```
import gsap from "gsap";
if (window.innerWidth > 1000) {
  document.querySelector(".trail-container").addEventListener("mousemove",
    (e) => {
    const img = document.createElement("img");
    img.src = `/images/work-items/work-item-${Math.floor(Math.random() * 10) + 1}.jpg`;
    img.classList.add("trail-img");
    document.querySelector(".trail-container").appendChild(img);
    gsap.to(img, { scale: 1, duration: 0.5, onComplete: () => img.remove() });
  });
}
document.querySelector("#contactForm").addEventListener("submit", (e) => {
  e.preventDefault();
  gsap.to(".success-message", { display: "block", opacity: 1, duration: 0.5 });
});
```

Listing 15: Excerpt from contact.js

Screenshot Placeholder: Insert a screenshot of the contact page image trail here.

3.4 Configuration Files

These files manage development and deployment.

3.4.1 *package.json*

Defines project metadata and dependencies:

- Scripts: Commands for development (`npm run dev`), building (`npm run build`), and previewing.
- Dependencies: GSAP, Lenis, and Vite.

```
{
  "name": "prashant-koirala-portfolio",
  "version": "1.0.0",
  "description": "Portfolio website for Prashant Koirala",
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview",
    "host": "vite --host"
  },
  "dependencies": {
    "gsap": "^3.13.0",
    "lenis": "^1.3.3",
    "vite": "^6.3.5"
  }
}
```

Listing 16: Excerpt from package.json

3.4.2 *vercel.json*

Configures deployment on Vercel:

- Build Command: Runs `npm run build`.
- Output Directory: Specifies `dist` for built files.

- Rewrites: Maps URLs to HTML files (e.g., `/contact` to `contact.html`).

```json
{
  "buildCommand": "npm run build",
  "outputDirectory": "dist",
  "rewrites": [
    { "source": "/home", "destination": "/index.html" },
    { "source": "/contact", "destination": "/contact.html" }
  ]
}
```

Listing 17: Excerpt from vercel.json

3.4.3 *vite.config.js*

Configures Vite:

- Input Files: Specifies `index.html` and `contact.html`.
- Assets: Includes image file types.

```javascript
import { defineConfig } from "vite";
export default defineConfig({
  build: {
    rollupOptions: {
      input: {
        main: "index.html",
        contact: "contact.html"
      }
    },
    assetsInclude: ["**/*.jpg", "**/*.png"]
  },
  publicDir: "public"
});
```

Listing 18: Excerpt from vite.config.js

4 Key Features and Animations

This section highlights the website's standout features and how they are implemented.

4.1 Page Transitions

Five colored overlays animate to cover or reveal pages, using GSAP's `scaleY` property with a stagger effect for a layered transition.

4.2 Smooth Scrolling

Lenis provides smooth scrolling, with GSAP's ScrollTrigger syncing animations to scroll position. Settings adjust for mobile and desktop.

4.3 Navigation Menu

The menu toggles a full-screen overlay with animated links, locking the page scroll to focus on navigation.

4.4 Hero Section

The hero image cycles through 10 project images every 250ms, with scroll-based animations for position, scale, and rotation.

4.5 Featured Work

Projects are displayed with a horizontal scrolling title and 3D image cards that move in depth (`z` axis) based on scroll progress.

4.6 Service Cards

Four cards highlight my skills, pinned during scroll with inner content animating upward for a dynamic effect.

Footer Particle Explosion When the footer enters the viewport, 10 image particles animate outward, reusing project images for a creative touch.

Contact Page Image Trail On desktop, images follow the cursor in the contact section, scaling up and disappearing after a short lifespan.

5 Deployment and Setup

This section explains how to set up and deploy the website.

5.1 Development Setup

To run the site locally:

1. Install Node.js.
2. Clone the project repository.
3. Run `npm install` to install dependencies.
4. Run `npm run dev` to start the Vite development server.

5.2 Deployment on Vercel

To deploy:

1. Push the project to a GitHub repository.
2. Connect the repository to Vercel.
3. Vercel uses `vercel.json` to build and route the site.

6 Challenges and Solutions

- Challenge: Understanding Cursor AI-generated code. Solution: Analyzed each file, used console logs for debugging, and tested animations.
- Challenge: Ensuring responsiveness. Solution: Used media queries in CSS and conditional JavaScript for mobile devices.

7 Future Enhancements

- Add a dedicated “Work” page using `work.css` and uncommented nav links.
- Expand the about section into a full page.
- Add more interactive features, like a project gallery with filters.

8 References

- GreenSock. (2025). GSAP Documentation. Retrieved from https://greensock.com/docs/
- Lenis. (2025). Lenis GitHub Repository. Retrieved from https://github.com/studio-freight/lenis
- Vite. (2025). Vite Documentation. Retrieved from https://vitejs.dev/guide/
- Vercel. (2025). Vercel Documentation. Retrieved from https://vercel.com/docs