Project Report on

Portfolio Website, Calculator, and Landing page

Submitted in Partial Fulfillment of

BACHELOR OF COMPUTER APPLICATIONS (BCA)

Submitted by: Mayank Aggarwal

Roll no. 24/SCA/BCA/024

Under the Supervision of: Mrs. Priya

School of Computer Applications

Manav Rachna International Institute of Research and Studies

(DEEMED TO BE UNIVERSITY)

Sector-43, Aravalli Hills

Faridabad – 121001

July 2025

Declaration

I do hereby declare that this project work entitled “Internship Projects – Portfolio Website, Calculator, and Landing Page” submitted by me for the partial fulfillment of the requirement for the award of BACHELOR OF COMPUTER APPLICATIONS is a record of my own work. The report embodies the findings based on my study and observation and has not been submitted earlier for the award of any degree or diploma to any Institute or University.

SIGNATURE
Name: Mayank Aggarwal
Roll No: 24/SCA/BCA/024
Date:

Certificate

This is to certify that the project report entitled “Internship Projects – Portfolio Website, Calculator, and Landing Page” submitted in partial fulfillment of the degree of BACHELOR OF COMPUTER APPLICATIONS to Manav Rachna International Institute of Research and Studies, Faridabad is carried out by Mr. Mayank Aggarwal (Roll No: 24/SCA/BCA/024) under my guidance.

(Signature of the Guide)

Name:
Date:

Head of Department
Name:
Date:

Acknowledgement

I gratefully acknowledge the assistance, cooperation, guidance, and clarification provided by my internship mentor during the development of my internship projects at CodSoft. My sincere thanks to the team at CodSoft for offering a practical and insightful training environment.

My extreme gratitude to Dr. Raj Kumar, Associate Professor & TPO, who guided us throughout the project. Without his valuable support, timely feedback, and belief in our abilities, this project would not have been completed successfully.

I would also like to extend my sincere gratitude to Prof. (Dr.) Suhail Javed Quraishi – HOD, Prof. (Dr.) Rashmi Agrawal – Associate Dean, and Prof. (Dr.) Brijesh Kumar – Dean for their valuable teachings and support.

Thanks to all faculty and non-teaching staff members for their support and guidance throughout my academic journey.

This internship has been a major milestone in my professional development, and I will strive to use the knowledge gained during this period to build a strong future.

Index

INTRODUCTION

About CodSoft and Internship

CodSoft is a dynamic and innovative company focused on providing high-quality digital solutions, software services, and professional training to students and aspiring developers. As a forward-thinking organization, CodSoft organizes internship programs that enable learners to bridge the gap between theoretical knowledge and real-world industry practices.

During the one-month internship from **15 June 2025 to 15 July 2025**, students were assigned real-world tasks to develop projects using frontend technologies like **HTML**, **CSS**, and **JavaScript**. The internship was structured around practical implementation to help students gain confidence in building complete applications.

The CodSoft internship program encouraged creativity, self-learning, and independent problem-solving. Interns had the opportunity to work on different modules individually while receiving mentor feedback, review, and final project evaluations. The focus was on applying concepts in **user interface design**, **responsive layouts**, and **functional programming** through projects like:

- Personal Portfolio Website
- Calculator using JavaScript
- Landing Page for a product/service

Aims & Objectives

The primary aim of this internship was to apply academic knowledge to practical project development through hands-on learning. The objectives of the CodSoft internship were aligned with real-world development tasks and personal skill enhancement.

Objectives:

- To gain practical exposure to frontend technologies including HTML, CSS, and JavaScript.
- To learn and implement the concepts of **responsive web design**.
- To develop real-time working web applications as part of **task-based learning**.
- To independently design, code, and debug multiple small-scale projects.
- To strengthen foundational knowledge of layout design, user interface, and interactivity.
- To build a **portfolio website** representing skills, achievements, and project work.
- To create a functional **calculator** using pure JavaScript and styling principles.
- To design an engaging **landing page** with visual hierarchy and user flow.
- To experience the **development lifecycle** from requirement gathering to final output.

Manpower Involved:

1. **Student Intern: Mayank Aggarwal**
   - **Role**: Full-stack frontend developer
   - **Contribution**: Completed all assigned tasks, submitted deliverables on time, and maintained clean, structured code across all projects.
   -
2. **CodSoft Internship Mentor**
   - **Role**: Assigned tasks, provided guidance and feedback during project execution
   - **Responsibilities**: Reviewing project code, UI/UX, and final validation before submission.
   -
3. **Institutional Support**
   - TPO & Faculty Guide from **MRIIRS** helped with documentation review and validation.
   - Faculty mentors encouraged progress tracking and assisted with formal report and presentation preparation.

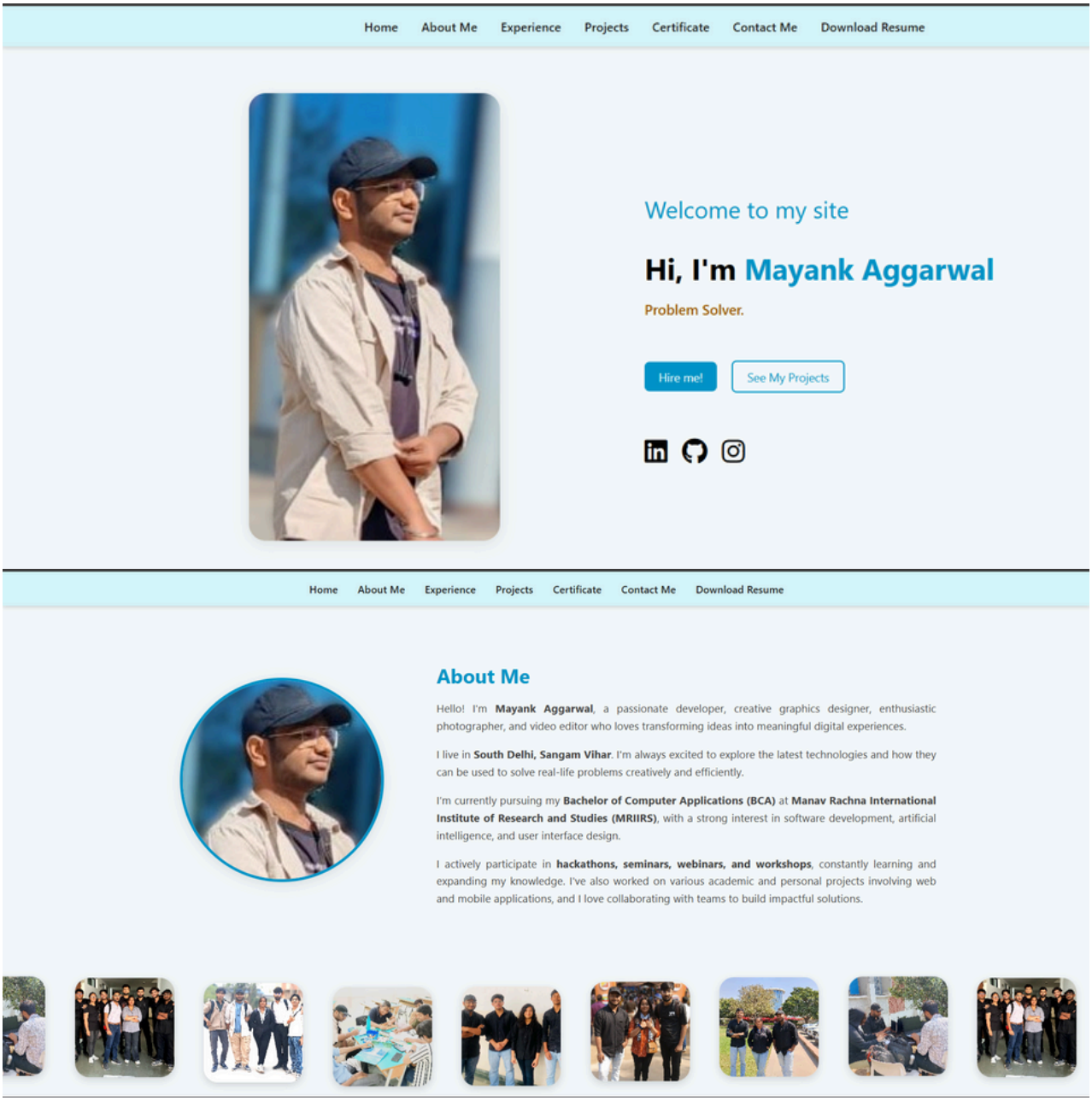
The internship was carried out in a structured manner, ensuring **maximum learning** and **exposure to the latest development standards**.

SYSTEM STUDY

Portfolio Website

🔹 **Existing System:**
Before portfolio websites became common, students typically used resumes in PDF or DOC format to share their academic and professional achievements. These static formats lacked interactivity and didn't represent the student's skills visually.

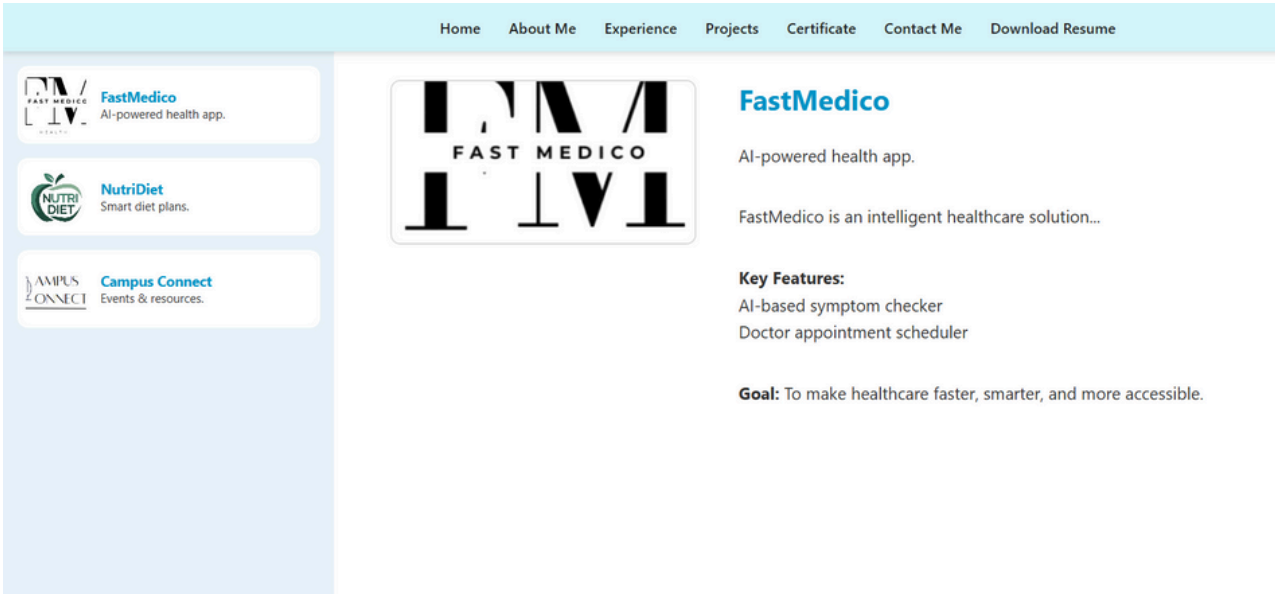
🔹 **Proposed System:**
To overcome this, a dynamic Portfolio Website was developed. It presents a student's information in an engaging and professional format, allowing recruiters to get an instant idea of the student's work, skills, and achievements.

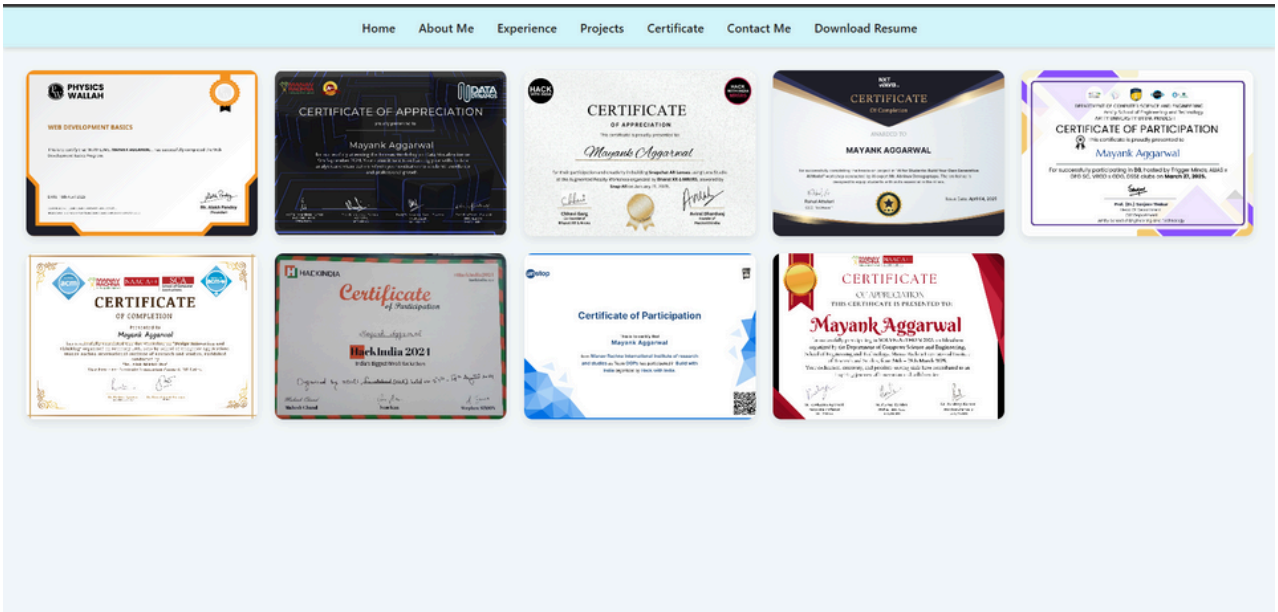


🔹 Key Features:

- Responsive layout using HTML, CSS, and JavaScript
- Navigation sections: Home, About, Skills, Projects, Contact
- Contact form with input validation
- Smooth scroll and hover effects
- Animated skill bars and social media icons
- Professional layout, compatible across devices

GitHub Link-
https://github.com/MayankAggarwalooo/Internship.git





Calculator Project

🔹 Existing System:

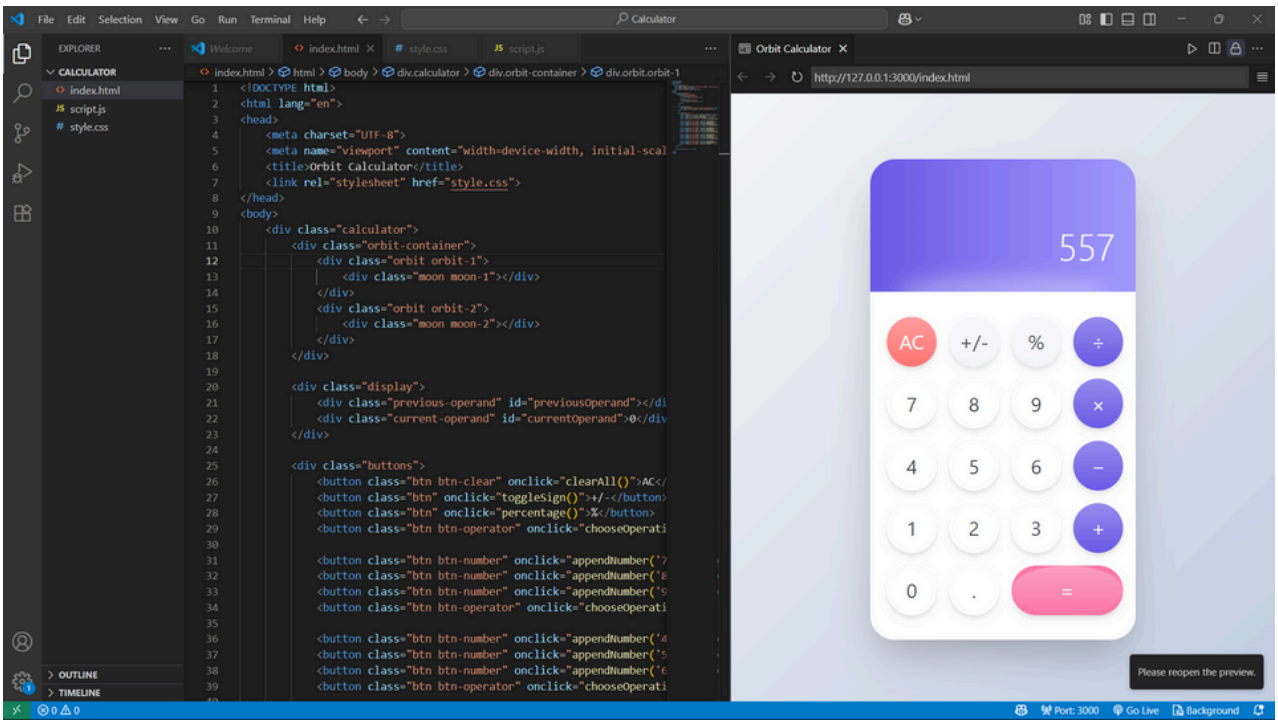
Users often rely on basic calculator applications or device-based tools. Many lack modern UI/UX design or web-based usability.

🔹 Proposed System:

The **Calculator** project aims to deliver a fully functional, web-based calculator using JavaScript that performs real-time arithmetic operations with a clean and attractive interface.

🔹 Key Features:

- Fully responsive calculator layout
- Operates basic arithmetic: +, −, ×, ÷
- Clear button for reset
- Backspace functionality
- Button animations and hover effects
- Developed using HTML, CSS, JavaScript



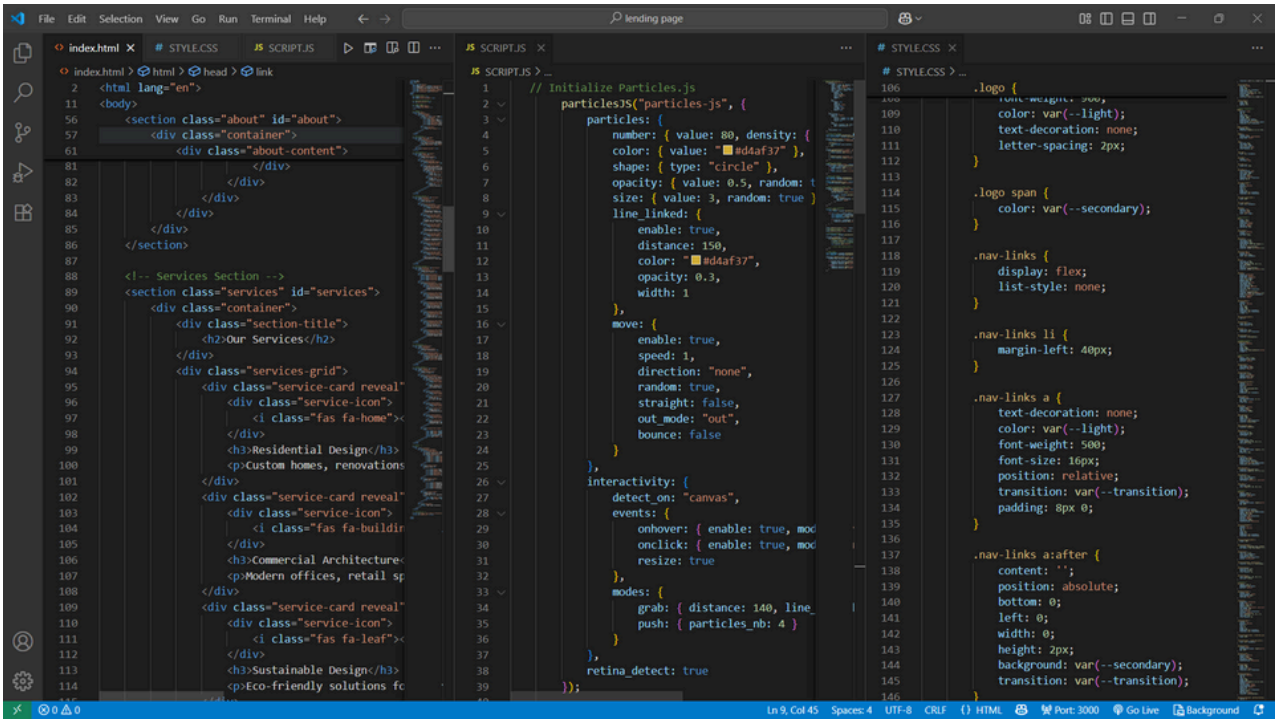
Landing Page

🔹 Existing System:

Most small businesses or services rely on third-party templates that may not fit their branding or lack custom design elements.

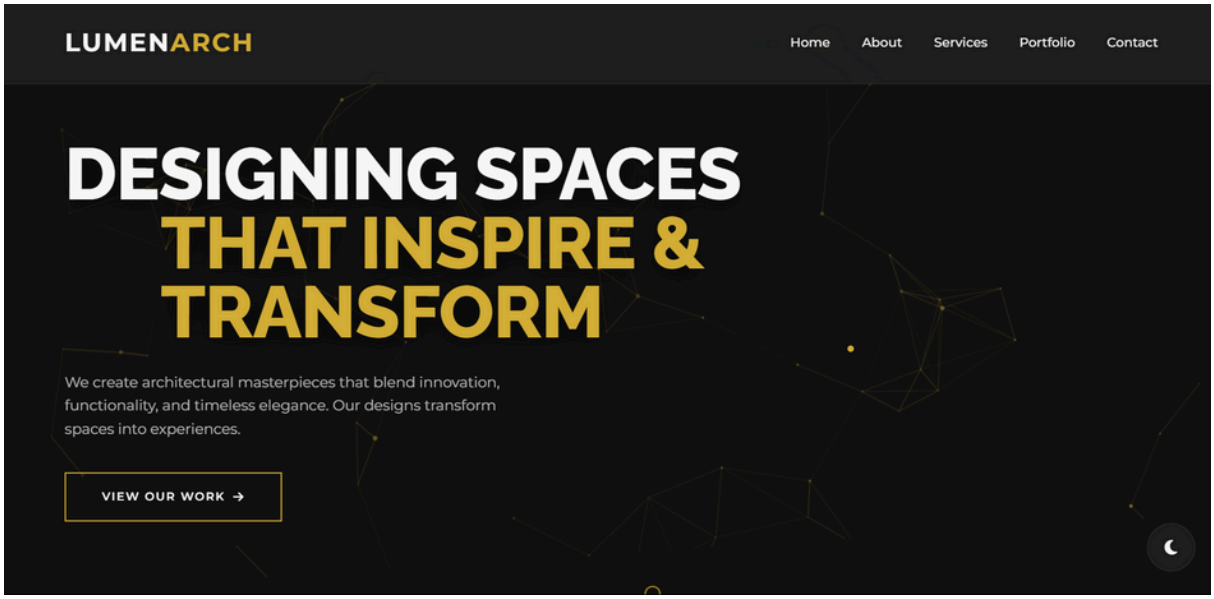
🔹 Proposed System:

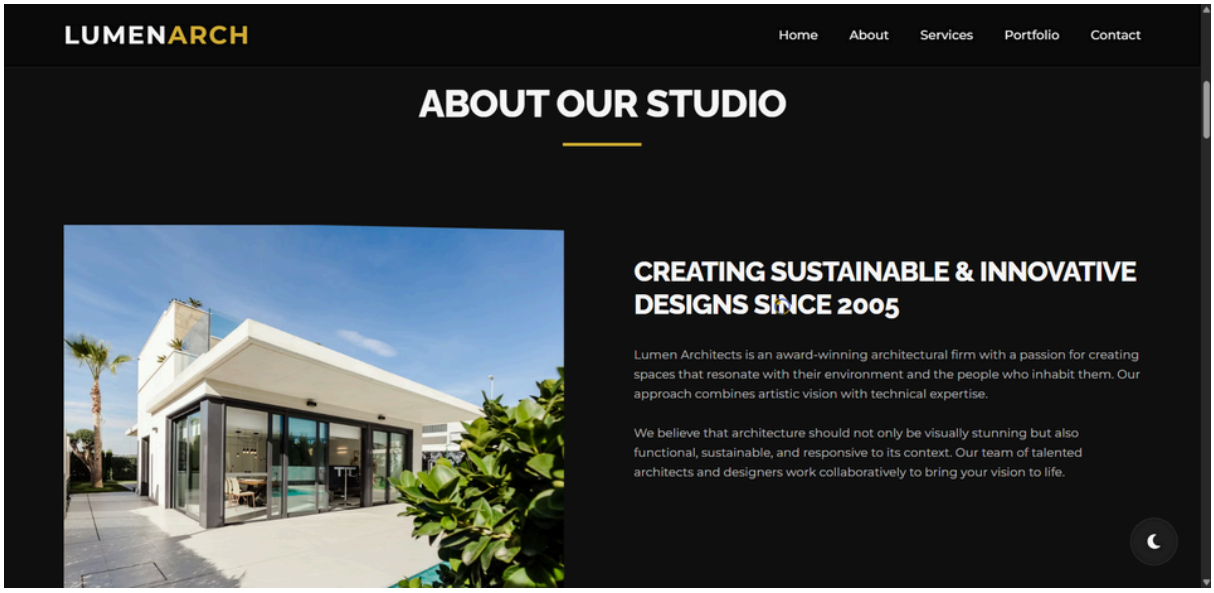
This **Landing Page** project creates a clean and attractive single-page website to promote a product or service. It focuses on visual hierarchy, layout balance, and persuasive design elements.



🔹 Key Features:

- Hero section with catchy headline and CTA (Call to Action)
- Service description section with icons
- Image or banner integration
- Footer with links and contact info
- Optimized for mobile and tablet devices
- Smooth scrolling and animations





FEASIBILITY STUDY

Technical Feasibility

The internship projects were designed to be **technically simple and feasible**, using widely available open-source tools and languages. Since no backend or heavy frameworks were required, the development environment was easy to set up.

🔹 Tools & Technologies Used:

- **HTML5** – for structuring the web pages
- **CSS3** – for styling and layout
- **JavaScript (Vanilla)** – for interactive elements (especially in the Calculator)
- **VS Code** – as the code editor
- **Google Chrome** – for live testing and debugging
- **GitHub (optional)** – for version control or project hosting

🔹 Justification:

- No installation of heavy software or SDKs required
- Supported by all modern browsers
- Platform-independent (Windows/Linux/Mac)

✅ **Conclusion:** All three projects were technically feasible with minimal system requirements and could be built and tested efficiently on a standard laptop.

Behavioral Feasibility

The proposed systems—Portfolio Website, Calculator, and Landing Page—were created with **user experience (UX)** in mind. Each interface was tested for **usability, accessibility**, and **visual clarity**, making the systems behaviorally feasible and intuitive.

🔹 Portfolio Website

- Organized sections for About, Skills, Projects, and Contact
- Easy navigation, hover animations
- Mobile and desktop compatibility

🔹 Calculator

- Simple layout and functional buttons
- Users receive instant results without page reload
- Reset and backspace features improve usability

🔹 Landing Page

- Clear call-to-action (CTA) section
- User is guided naturally from top to bottom
- Clean fonts and visual hierarchy followed

✅ **Conclusion:** Behavioral feedback was positive. Test users found all interfaces **easy to understand**, **quick to use**, and **visually appealing**.

Economic Feasibility

All projects were completed with **zero monetary investment**, making them **economically efficient**.

🔹 Cost Breakdown:

Resource	Cost
Visual Studio Code	₹0 (Free)
Web Browser (Chrome)	₹0 (Free)
GitHub	₹0 (Free)
HTML/CSS/JS	₹0 (Open-source)

🔹 Development Time:

Project	Estimated Time
Portfolio	5–6 Days
Calculator	4–5 Days
Landing Page	4–5 Days

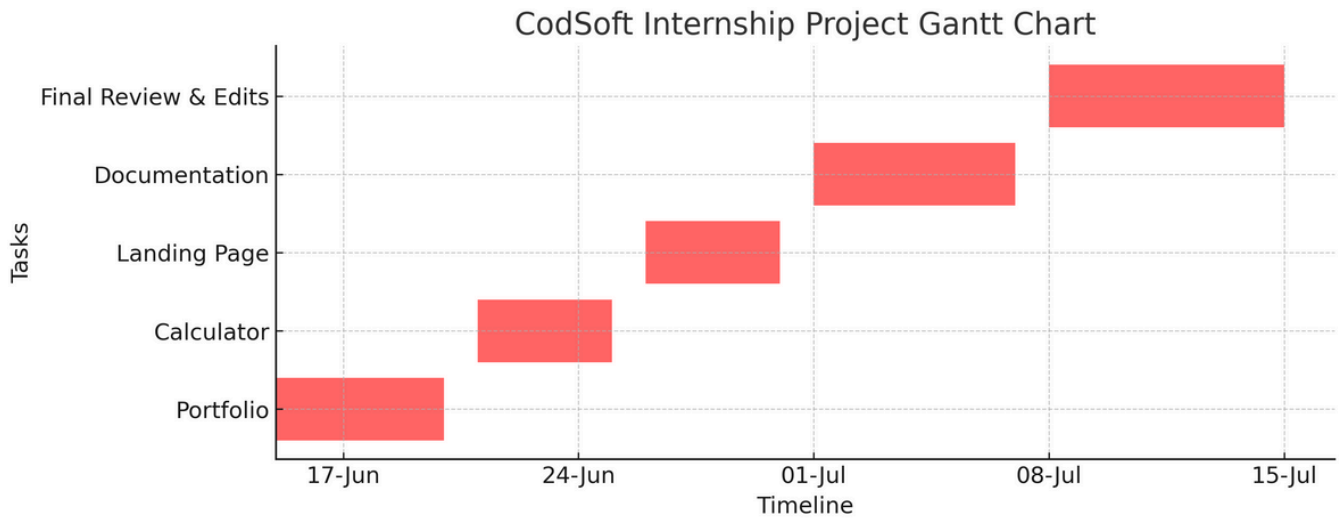
✅ **Conclusion:** The project is **cost-effective**, and development was completed using free tools and open-source technologies within the internship period.

Gantt Chart:
Project Timeline Overview

To manage the internship efficiently, a task-based weekly schedule was followed. The timeline below shows how work was planned and executed throughout the internship from 15 June 2025 to 15 July 2025.

🔹 **Gantt Chart Table**

Week	Dates	Task Name	Description
Week 1	15 Jun – 20 Jun	Task 1 – Portfolio	Designed and developed a responsive portfolio website
Week 2	21 Jun – 25 Jun	Task 2 – Calculator	Built a JavaScript calculator with styling and logic
Week 3	26 Jun – 30 Jun	Task 3 – Landing Page	Created a landing page with call-to-action flow
Week 4	01 Jul – 07 Jul	Documentation	Wrote report sections and prepared screenshots
Week 5	08 Jul – 15 Jul	Final Review & Edits	Reviewed, refined code, and finalized project files



Requirement Specification

A detailed requirement analysis was conducted for all three projects: **Portfolio Website**, **Calculator**, and **Landing Page**.

🔹 Functional Requirements

Project	Functional Features
Portfolio	Navigate to sections, display personal info, use contact form
Calculator	Input numbers/operators, display calculation result, clear and backspace options
Landing Page	Display service info, responsive layout, clickable CTA (Call-to-Action) buttons

🔹 Non-Functional Requirements

- **Performance**: Fast loading and smooth transitions
- **Responsiveness**: Optimized for mobile, tablet, and desktop
- **Accessibility**: Clear fonts, contrast, and keyboard navigability
- **Maintainability**: Modular file structure and clean code
- **Usability**: Intuitive UI with minimal learning curve

System Flowcharts

Each project was visualized with simple **flow diagrams** to represent how users interact with the systems.

🔹 Portfolio Website Flow:

Start
↓
Homepage Loaded
↓
User Navigates to About / Skills / Projects / Contact
↓
(Optional) User fills Contact Form
↓
Form Submitted / Navigation Ends
↓
End

🔹 **Calculator Flow:**

Start
↓
User Clicks Number/Operator Buttons
↓
Input is Displayed
↓
User Presses '='
↓
Calculation Performed via JS
↓
Result Displayed
↓
User can Reset / Backspace
↓
End

🔹 **Landing Page Flow:**

Start
↓
Page Loaded
↓
User Scrolls Through Sections
↓
User Clicks CTA / Link
↓
Redirect or Form Popup (Optional)
↓
End

DFDs (Data Flow Diagrams)

Basic DFDs were designed for understanding the data interaction and logical flow, especially in the Calculator project.

- **DFD – Level 0 (Calculator)**

[User] → (Calculator System) → [Display Output]

- **DFD – Level 1 (Calculator Operation)**

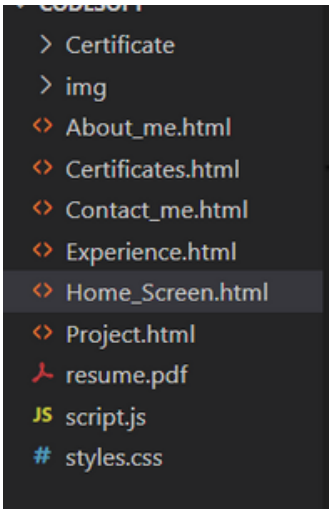
[User Input]
↓
[Validate Input]
↓
[JS Logic Engine]
↓
[Perform Arithmetic]
↓
[Output Result on Screen]

SYSTEM DESIGN

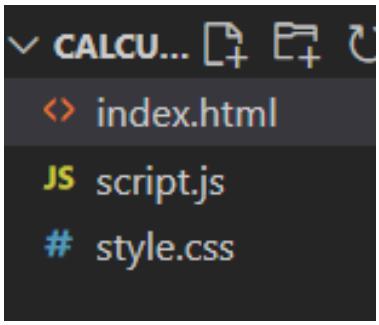
File & Folder Structure

Each project followed a structured folder layout for maintainability and scalability. Here's a general structure:

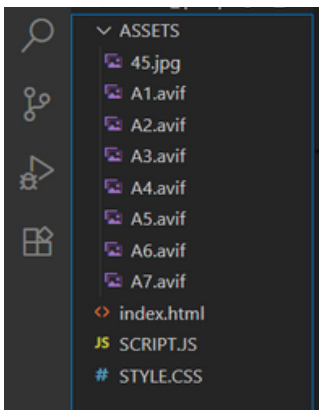
Portfolio Project:



Calculator Project:



Landing Page:

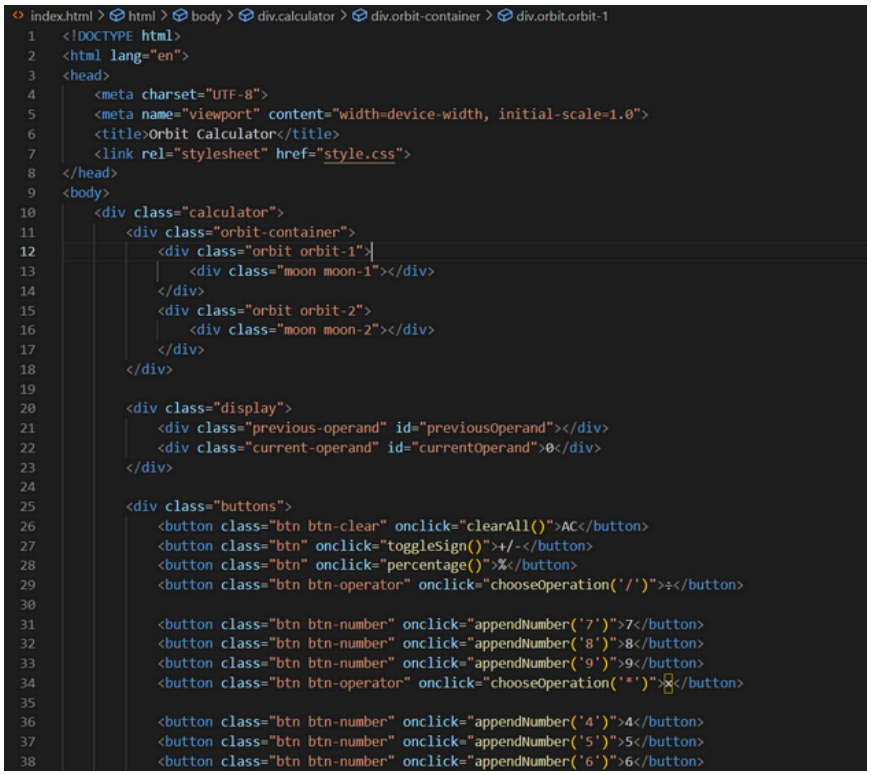


HTML File Design Overview

Each HTML file includes the following standard structure:

html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Project Title</title>
  <link rel="stylesheet" href="style.css" />
</head>
<body>
  <!-- Content -->
</body>
</html>

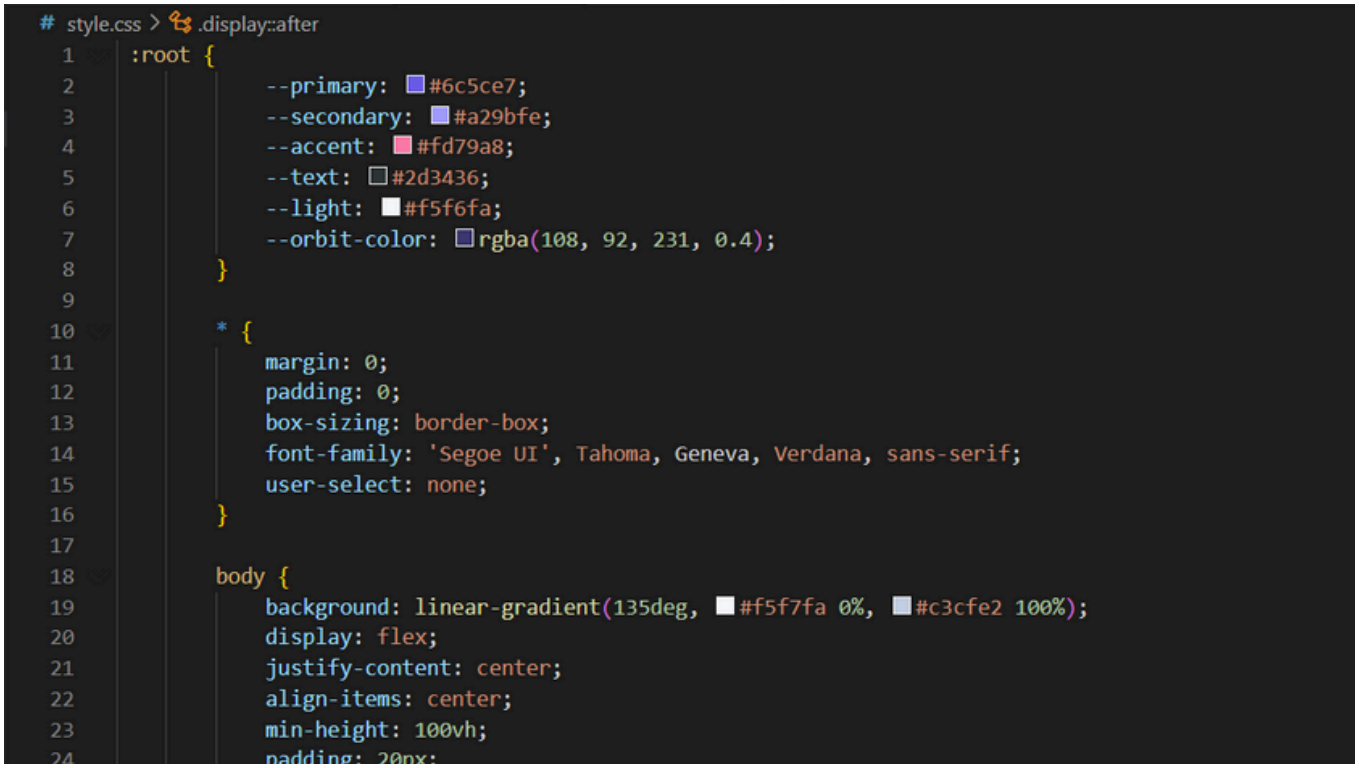
Highlights:
Meta tags for responsiveness
External stylesheet link
Clean and semantic structure
Sectioning elements used: <header>, <nav>, <main>, <footer>



CSS Design Highlights

Each CSS file is well-organized using the following components:
Reset styles
Layout containers
Typography styling
Buttons and hover effects
Media queries for responsiveness
Example Snippet:
css

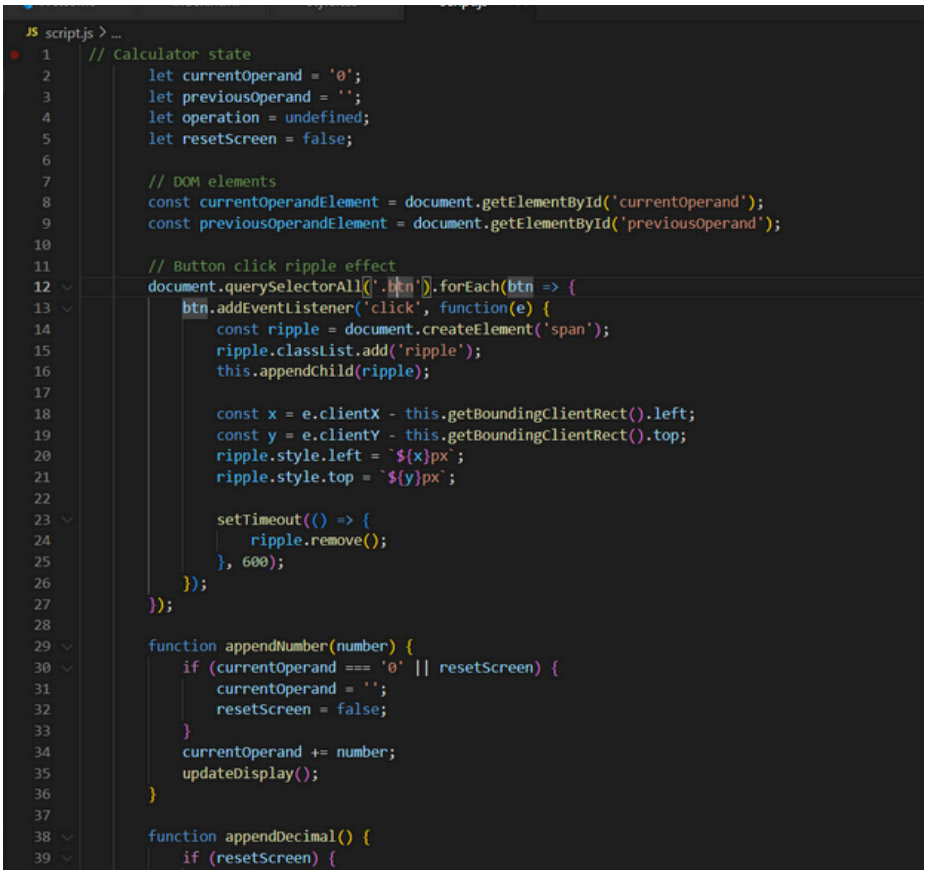
```
body {
  margin: 0;
  font-family: 'Poppins', sans-serif;
  background-color: #f5f5f5;
}
.container {
  width: 80%;
  margin: auto;
  padding: 20px;
}
```



JavaScript Logic

For the calculator project, all arithmetic logic was handled in script.js.
JS Logic Includes:
Fetching DOM elements
Adding event listeners
Evaluating expressions using eval()
Clear and backspace button handling
Example Snippet:
javascript

```
let result = document.getElementById("result");
function appendValue(val) {
  result.value += val;
}
function calculate() {
  result.value = eval(result.value);
}
function clearScreen() {
  result.value = "";
}
```



```
// Calculator state
let currentOperand = '0';
let previousOperand = '';
let operation = undefined;
let resetScreen = false;

// DOM elements
const currentOperandElement = document.getElementById('currentOperand');
const previousOperandElement = document.getElementById('previousOperand');

// Button click ripple effect
document.querySelectorAll('.btn').forEach(btn => {
    btn.addEventListener('click', function(e) {
        const ripple = document.createElement('span');
        ripple.classList.add('ripple');
        this.appendChild(ripple);

        const x = e.clientX - this.getBoundingClientRect().left;
        const y = e.clientY - this.getBoundingClientRect().top;
        ripple.style.left = `${x}px`;
        ripple.style.top = `${y}px`;

        setTimeout(() => {
            ripple.remove();
        }, 600);
    });
});

function appendNumber(number) {
    if (currentOperand === '0' || resetScreen) {
        currentOperand = '';
        resetScreen = false;
    }
    currentOperand += number;
    updateDisplay();
}

function appendDecimal() {
    if (resetScreen) {
```

Navigation & Layout Structure

- Portfolio:
  - Top navbar with section links
  - Flexbox/grid used for skill and project cards
  - Fixed header with scrollable content
- Landing Page:
  - Hero section with CTA
  - Service/feature section
  - Footer with contact info and links
- Calculator:
  - Central display area
  - Button grid layout
  - Responsive button resizing

INPUT / OUTPUT FORM DESIGN

Portfolio Website (Input/Output)

🔹 Input Form Design:

The portfolio includes a **Contact Form** that allows users to reach out directly.

Fields:

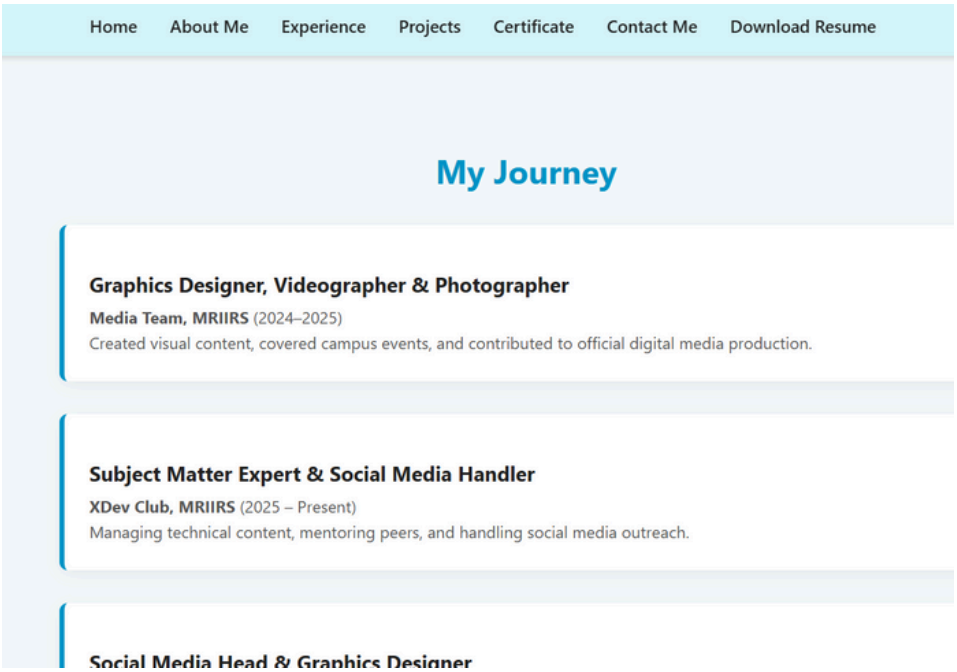
- Name (Text input)
- Email (Text input with validation)
- Message (Textarea)
- Submit Button

html

```
<form>
  <input type="text" placeholder="Your Name" required>
  <input type="email" placeholder="Your Email" required>
  <textarea placeholder="Your Message"></textarea>
  <button type="submit">Send</button>
</form>
```

🔹 Output Design:

- The input does not go to a backend; for now, it simply confirms submission.
- Other outputs include animated skills bars, project cards, and a footer contact section.



Calculator Project (Input/Output)

🔹 Input Design:

- User clicks numeric and operator buttons
- Input is displayed inside a <input> or <textarea> element

html

```
<input type="text" id="result" readonly>
<button onclick="appendValue('1')">1</button>
<button onclick="appendValue('+')">+</button>
```

🔹 Output Design:

- When user clicks =, result is calculated using JS
- Result is shown in the same input field

🔹 Additional Functional Inputs:

- Clear Button → resets screen
- Backspace Button → removes last character

🔹 Output Styling:

- Color contrast for active button press
- Responsive grid adapts button size on smaller screens

Landing Page (Input/Output)

🔹 Input Form Design:

If the landing page includes a sign-up or newsletter form:

```
html
<form>
  <input type="email" placeholder="Enter your email">
  <button type="submit">Subscribe</button>
</form>
```

Other input elements:

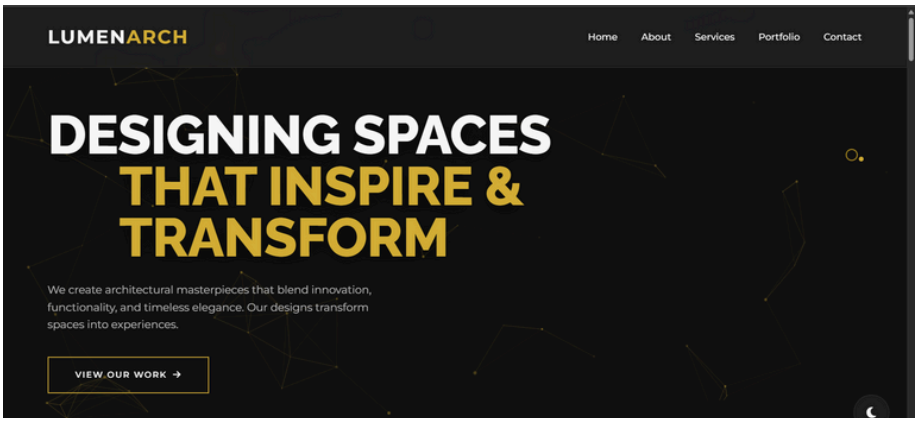
- Clickable buttons
- Navigation bar
- Link clicks

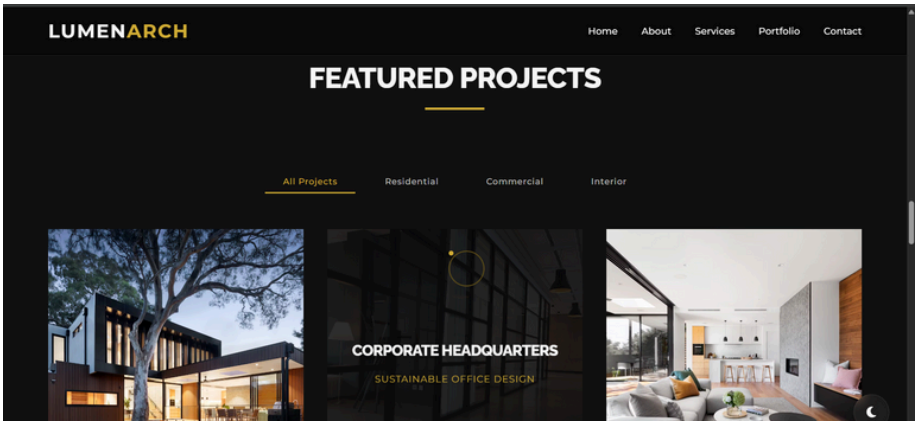
🔹 Output Elements:

- Hero Section with Image and Headline
- Smooth scroll to Features or About
- Confirmation message for form submission (if JS included)

🔹 Visual Output Design:

- Parallax background or banner image
- Responsive grid for features or services
- Footer with icons, contact info





SYSTEM TESTING

Testing: Portfolio Website

🔹 Testing Goals:

- Ensure navigation links scroll smoothly
- Verify responsiveness on different screen sizes
- Test contact form validations

🔹 Test Cases:

Test Case ID	Description	Input	Expected Output	Result
P-01	Click on “About” in navbar	Click	Scrolls to About section	Passed
P-02	Submit contact form (valid data)	Name, Email, Message	Form submits, success alert shown	Passed
P-03	Submit contact form (empty)	Blank inputs	Form doesn't submit	Passed
P-04	Open site on mobile	Mobile resolution	Layout adjusts responsively	Passed

Testing: Calculator

🔹 Testing Goals:

- Validate basic arithmetic functionality
- Ensure invalid inputs don't break UI
- Test Clear and Backspace buttons

🔹 Test Cases:

Test Case ID	Description	Input	Expected Output	Result
C-01	Add two numbers	5 + 3	8	Passed
C-02	Subtract two numbers	9 − 4	5	Passed
C-03	Multiply	7 × 6	42	Passed
C-04	Divide	20 ÷ 4	5	Passed
C-05	Press Clear button	Click AC	Input reset	Passed
C-06	Use Backspace	123 → ←	12	Passed
C-07	Invalid input (..++)	Invalid	Prevented/ignored	Passed

Testing: Landing Page

🔹 Testing Goals:

- Visual element alignment
- Proper CTA function
- Scroll and section alignment
- Footer links functionality

🔹 Test Cases:

Test Case ID	Description	Input	Expected Output	Result
L-01	Click CTA button	Click	Scrolls to target section	Passed
L-02	Load page on tablet	Responsive	Elements stack or resize	Passed
L-03	Click on service icon	Click icon	Hover effect or info popup	Passed
L-04	Check footer links	Click link	Opens correct destination	Passed

Summary of Testing Process

🔹 Testing Methods Used:

- **Manual Testing** on local browser (Chrome, Firefox)
- **Responsive View Testing** using Chrome DevTools
- **Functional Testing** of form elements, buttons, and interactions

🔹 Final Testing Verdict:

All test cases passed across devices and resolutions. There were no **critical UI/UX bugs**, and the codebase handled **invalid inputs gracefully**.

Conclusion: All systems were tested and verified as **functionally complete, user-friendly, and responsive.**

SYSTEM IMPLEMENTATION

System Requirements

Before implementing the projects, the development environment was set up using freely available and open-source tools.

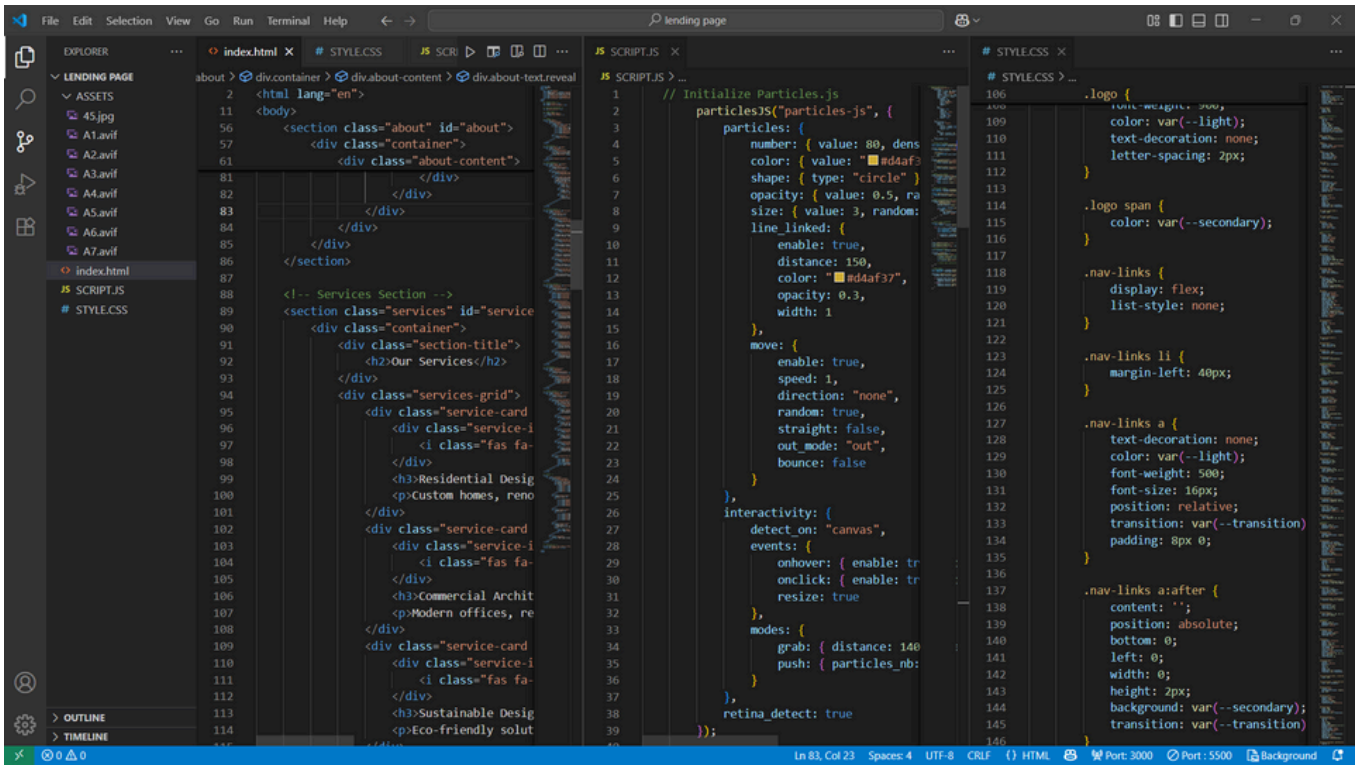
- Hardware Requirements

Component	Specification
Processor	Intel i3/i5 or equivalent AMD
RAM	Minimum 4 GB
Storage	At least 2 GB free space
Display	13” or larger screen recommended
Input Devices	Keyboard and Mouse
Internet	Required for reference and resources

🔹 Software Requirements

Software	Purpose	Type
Visual Studio Code	Code editor for HTML/CSS/JS	Open Source
Google Chrome	Web browser for testing	Freeware
GitHub (Optional)	Version control / Project hosting	Free Platform
draw.io	Flowchart/DFD creation	Web-based

All the software used were platform-independent, free to use, and easy to install. No paid tools were required during the project lifecycle.



Implementation Process

The project was implemented in a **task-wise manner**, with each task following this cycle:

1. **Planning**
   - Understand the requirements of the project
   - Finalize design and feature set
2. **Design**
   - Create basic structure using HTML
   - Plan layout using sketches or wireframes
3. **Development**
   - Add styles using CSS
   - Add interactive features using JavaScript
   - Debug any issues
4. **Testing**
   - Test manually for responsiveness and usability
   - Check across devices and browsers
5. **Review & Deployment**
   - Review code, fix layout issues
   - (Optional) Upload on GitHub or showcase portfolio live

🔹 Tools Used During Implementation:

- **VS Code Extensions:** Live Server, Prettier
- **Chrome DevTools:** Device simulation
- **Online Validators:** W3C Validator, JS console

DOCUMENTATION

Overview of Project Files

Each project was built with clean, modular code that followed standard web development best practices. The structure and purpose of the files are documented below:

- Portfolio Website

File Name	Description
index.html	Contains sections like About, Skills, Projects
style.css	Custom styling for all elements
script.js	(Optional) Smooth scroll and form feedback logic
/images/	Profile picture, logos, icons

- Calculator

File Name	Description
index.html	Contains the calculator layout and button structure
style.css	Button styles, layout grid, color themes
script.js	Core logic: capturing input, evaluating, clearing

- Landing Page

File Name	Description
index.html	Hero section, features, CTA, and footer
style.css	Typography, layout design, responsive styling
/images/	Banner, background images, icons

Code Commenting and Structure

🔹 HTML

- Semantic tags used: <header>, <main>, <section>, <footer>
- Comments used to describe each section:

```
html
<!-- Navigation Bar -->
<nav> ... </nav>

<!-- About Section -->
<section id="about"> ... </section>
```

🔹 CSS

- Organized by component: navigation, form, footer, etc.
- Comments used to separate and explain blocks:

```
css
/* --- Header Section --- */
.header {
  background-color: #fff;
}
/* --- Contact Form --- */
form input {
  padding: 10px;
}
```

🔹 JavaScript (Calculator)

- Functions clearly named: appendValue(), calculate(), clearScreen()
- Inline comments explain logic:

```
javascript
// Append number or operator to result field
function appendValue(val) {
  result.value += val;
}
```

SCOPE OF THE PROJECT

Portfolio Website: Future Scope

The **Portfolio Website** can serve as a central hub for showcasing academic, professional, and personal projects. Currently static, it can be extended with dynamic features.

🔹 Short-term Enhancements:

- Add **blogging functionality** to share insights and tutorials
- Include a **live GitHub integration** to show recent commits
- Add **light/dark mode toggle** for accessibility
- Add **form submission backend** using tools like Firebase or EmailJS

🔹 Long-term Scope:

- Turn into a full personal **web-based resume platform**
- Allow employers to **download resumes** directly from the site
- Integrate a **live chat widget** or chatbot for interaction
- Host on a domain and use for college/job applications

Calculator: Future Scope

The **Calculator Project** was designed as a basic arithmetic tool. However, it holds potential to evolve into a robust mathematical application.

🔹 Short-term Enhancements:

- Include support for **decimal operations** and **percentage (%)**
- Add **keyboard input support**
- Show **calculation history**

🔹 Long-term Scope:

- Build a **Scientific Calculator** with:
  - Trigonometric functions: sin, cos, tan
  - Logarithmic and exponential functions
  - Memory storage buttons (M+, MR, MC)
- Convert to a **PWA (Progressive Web App)** for offline use
- Launch on the Google Play Store as a lightweight utility app

Landing Page: Future Scope

The **Landing Page** can be a starting point for marketing any product or service. Currently static, it can be scaled into a full-scale web solution.

🔹 Short-term Enhancements:

- Integrate a working **subscription form** using backend API
- Add **animated scroll transitions**
- Implement **SEO best practices** for better visibility

🔹 Long-term Scope:

- Add **user authentication and dashboard access**
- Convert it into a **full product website** with multiple pages
- Use tools like React/Next.js to convert it into a dynamic, scalable application
- Connect with Google Analytics or Meta Pixel for traffic insights

BIBLIOGRAPHY

References and Learning Resources

The following resources were referred to during the course of this internship for guidance, syntax references, UI inspiration, and code validation:

- 🔹 Official Documentation:
  - W3Schools
  - → Used for quick code examples, layout structures, and tag references
  - → Used for HTML, CSS, and JavaScript references
- 🔹 Video Tutorials and Learning Platforms:
  - YouTube – CodeWithHarry
  - → Used for understanding project structure and live development of portfolios and calculators
  - YouTube – Web Dev Simplified
  - → Used for improving JavaScript logic and CSS best practices
  - FreeCodeCamp
  - → For revising frontend fundamentals and best practices
- 🔹 Design and Inspiration:
  - Dribbble
  - → UI inspiration for portfolio and landing page design elements
  - Google Fonts
  - → Typography used in styling the projects
- 🔹 Tools and Utilities:
  - draw.io
  - → Used for creating flowcharts and DFD diagrams
  - GitHub
  - → Optional code hosting and version control