WEB TECHNOLOGIES

PORTFOLIO GENERATOR

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE & ENGINEERING

(DATA SCIENCE)

Submitted by:

B.PALLAVI- A22126551068

K.SRUTHI– A22126551078

K.SRAVANI- A22126551079

P.LAHARI- A22126551088

M.LAVANYA- A22126551091

M.SUJANA- A22126551097

T.NEERAJA – A22126551120

Under the Guidance of

Mrs. Ch. Sravanthi Sowndarya

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI&ML, DS)

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(UGC AUTONOMOUS)

(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with A^{+})

Sangivalasa, Bheemili Mandal, Visakhapatnam - 531162. (A.P)

2022-2026

CONTENTS

INTRODUCTION:

This portfolio project is a dynamic and interactive web application designed to professionally showcase an individual's profile, skills, and achievements. It serves as a personalized platform for students, developers, and professionals to present their educational background, technical expertise, and projects in a structured and visually appealing manner.

The application is built using **HTML**, **CSS**, and **Flask (Python)** for the backend. It incorporates user input through a detailed form, allowing individuals to submit information such as their name, education, skills, contact details, and projects. Once submitted, this data is rendered across multiple pages—**Home**, **About**, **Skills**, **Projects**, and **Contact**—making the portfolio look dynamic and tailored to the user.

The project focuses on clean UI/UX design, responsiveness, and data consistency. It aims to help users create an impressive online presence, whether for academic purposes, job applications, or freelance opportunities. The form-driven approach ensures that the portfolio remains flexible and can be generated for different users without modifying the core code.

This project also reflects a real-world understanding of full-stack development, combining frontend design with backend logic and routing, making it a strong addition to any developer’s project portfolio.

KEY CONCEPTS:

◇ HTML (HyperText Markup Language)

➤ What is HTML?

HTML stands for HyperText Markup Language, and it is the standard language used to create and design the structure of web pages. It tells the browser what content to display and how to organize it using elements like headings, paragraphs, images, links, lists, forms, and more.
HTML is the backbone of every website. Without it, there would be no organized layout or structure for users to interact with.

➤ Role of HTML in This Project

In this portfolio project, HTML is used to:

- Create the layout of all five pages: Home, About, Skills, Projects, Contact.
- Structure form fields to collect user details (like name, email, skills, etc.).
- Display content like text, headings, project descriptions, and contact information.
- Define navigation through links to different sections.
- Serve as the base for integrating CSS for styling and Flask/Jinja2 for dynamic rendering.

➤ Key Features of HTML

- Simple and easy to learn
- Supports multimedia (images, audio, video)
- Compatible with all modern browsers
- Works seamlessly with CSS, JavaScript, and backend frameworks like Flask

➤ Installation Process

No installation required. Every web browser supports HTML by default. You can create HTML files using any code editor (like VS Code or Sublime Text) and save them with a .html extension.

◇ CSS (Cascading Style Sheets)

➤ What is CSS?

CSS stands for Cascading Style Sheets, and it is the language used to style and visually present HTML content. While HTML provides the structure, CSS brings that structure to life by applying styles such as colors, fonts, spacing**,** layout, backgrounds, animations, and responsiveness.

With CSS, developers can make web pages look attractive and improve the user experience.

➤ Role of CSS in This Project

In this portfolio project, CSS is used to:

- Style the layout of all pages (Home, About, Skills, Projects, Contact)
- Apply custom fonts, spacing, and colors to improve readability
- Add rounded corners, shadows, and animations for a modern design
- Make the design responsive across different screen sizes
- Maintain a consistent look and feel across all pages

CSS is what makes the portfolio look professional and polished.

➤ Key Features of CSS

- Controls the layout and visual presentation of HTML elements
- Enables separation of content and design (cleaner code)
- Supports responsive design using media queries
- Allows reusability through external style sheets
- Works with modern web technologies and frameworks

➤ Installation Process

No separate installation is required for CSS.

You just need a text editor like VS Code and a browser.

◇ JavaScript

➤ What is JavaScript?

JavaScript is a high-level, interpreted programming language that is essential for adding interactivity and dynamic behavior to websites. While HTML provides structure and CSS handles styling, JavaScript controls the behavior of the page. It allows developers to create responsive interfaces, validate form inputs, handle user events (like clicks or keypresses), and manipulate content dynamically.

JavaScript plays a major role in modern web development and works seamlessly with HTML and CSS in the browser.

➤ Role of JavaScript in This Project

In this portfolio project, JavaScript is used to:

• Enhance interactivity by handling user actions
• Add smooth scroll and transition effects
• Enable dynamic updates to the content without reloading the page
• Validate form input on the client-side before submission
• Improve user experience with features like menu toggles, animations, and popups (if added)

JavaScript helps make the portfolio more engaging and interactive.

➤ Key Features of JavaScript

• Works directly in the browser without any setup
• Event-driven and supports asynchronous programming
• Can update HTML and CSS on the fly using DOM manipulation
• Supported by all modern browsers
• Works with libraries like jQuery and frameworks like React, Vue, Angular

➤ Installation Process

No special installation is required for JavaScript:

• It runs directly in any modern web browser
• You can write JavaScript code within HTML using the <script> tag
• Or use an external .js file and link it using:

<script src="script.js"></script>

◇ Python

➤ What is Python?

Python is a high-level, interpreted programming language known for its simplicity, readability, and flexibility. It supports multiple programming paradigms like procedural, object-oriented, and functional programming.

Python is widely used in web development, data science, machine learning, automation, and more. Its vast ecosystem of libraries and frameworks makes it an excellent choice for building robust applications quickly and efficiently.

➤ Role of Python in This Project

In this portfolio project, Python is used to:

- Build the backend of the web application using the Flask framework
- Handle routing and form submission
- Process and render user-submitted data dynamically
- Integrate HTML templates with user data using Jinja2
- Maintain clean server-side logic and manage project workflows

Python makes the project dynamic and interactive instead of just being a static site.

➤ Key Features of Python

• Simple and beginner-friendly syntax
• Supports integration with web frameworks like Flask and Django
• Huge standard library and third-party package support
• Cross-platform compatibility
• Ideal for backend development, automation, and data processing

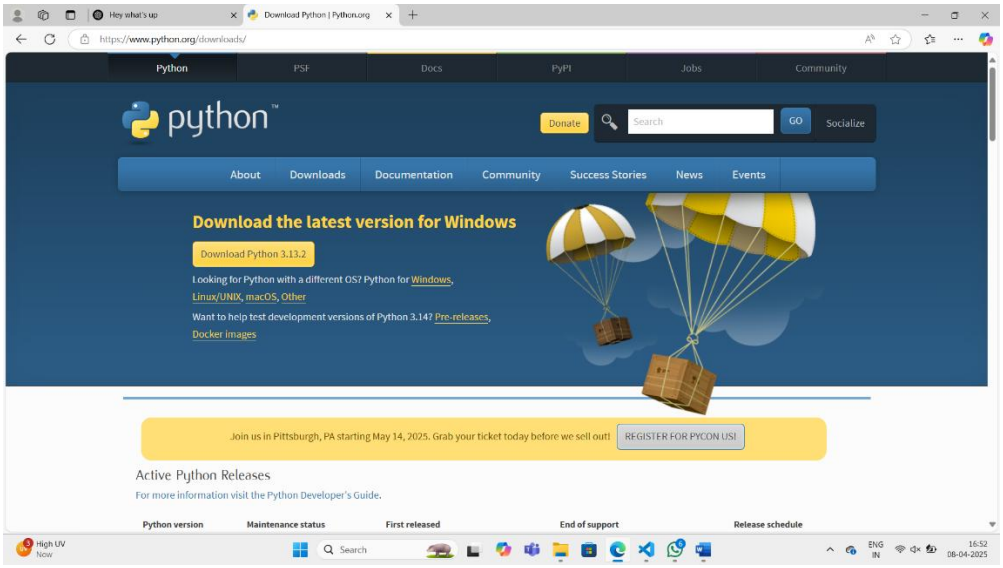
➤ Installation Process

Follow these steps to install Python:

1. Download Python:

   Visit https://www.python.org/downloads/

   - Choose the latest stable version



2.Install Python:

- Run the installer and check the box that says “Add Python to PATH”
- Click Install Now

3.Verify Installation:

- Open your terminal or command prompt
- Type python --version to confirm it's installed

◇ Flask

➤ What is Flask?

Flask is a lightweight and flexible web framework written in Python. It is designed to be simple yet powerful, allowing developers to create web applications quickly and efficiently. Flask follows a “micro” framework approach, meaning it doesn't come with built-in tools like database abstraction or form validation—giving developers the freedom to choose what they need.

Despite being lightweight, Flask is highly extensible and supports various third-party libraries and extensions, making it ideal for both small projects and large-scale applications.

➤ Role of Flask in This Project

In this portfolio project, Flask is used to:

- Serve HTML pages dynamically from the backend
- Handle form submissions and pass user input to the profile page
- Manage routing for different pages (Home, About, Skills, Projects, Contact)
- Integrate HTML templates with data using the Jinja2 templating engine
- Provide a lightweight backend structure to process and display user portfolios

Flask is what makes the static HTML-CSS pages dynamic and interactive in this project.

➤ Key Features of Flask

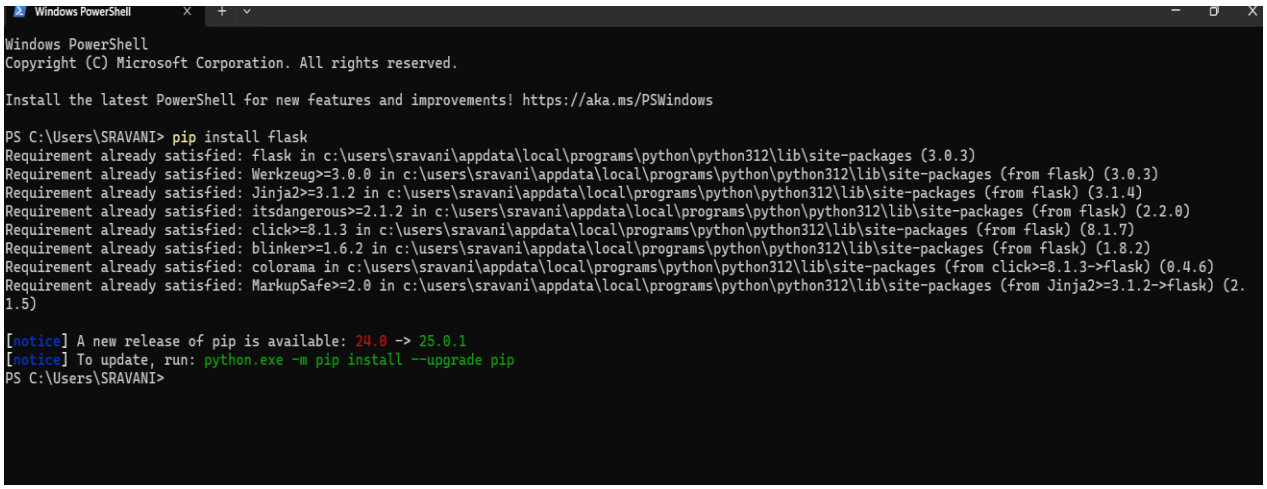
• Lightweight and easy to learn
• Built-in development server and debugger
• Supports Jinja2 templating for dynamic content
• RESTful request handling (GET, POST)
• Extensible with many libraries like Flask-WTF, Flask-SQLAlchemy
• Ideal for prototyping and building small to medium web apps

➤ Installation Process

Follow these steps to install Flask:

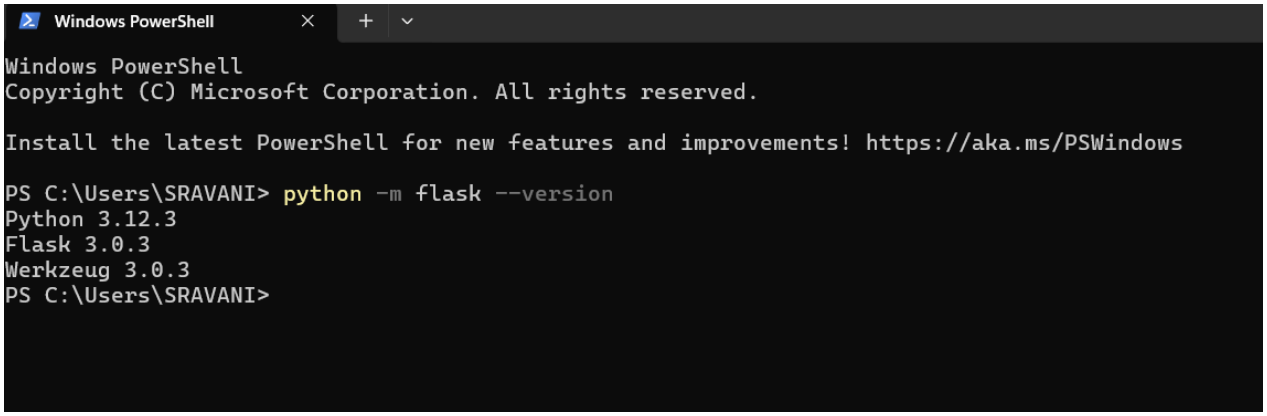
1. Install Python (if not already):
    - Required to use Flask (see previous Python section)
2. Install Flask using pip:

   Open your terminal or command prompt and type



3.Verify Installation:

Run this command to check if Flask is installed.



4.Run the App:

Save the file as app.py and run: **python app.py**

Implementation:

The portfolio web application is built using a combination of frontend and backend technologies to deliver a dynamic, interactive, and professional-looking website. Below is a breakdown of how each technology was implemented in the project…

◇ 1. HTML – Structure

HTML is used to design the basic **structure and layout** of the portfolio. It includes the creation of five main pages:

- **Home**
- **About**
- **Skills**
- **Projects**
- **Contact**

Each page is written using .html files and contains various tags like:

- <nav>, <header>, <section> for organizing content
- <form> elements on the Contact page for collecting user data
- , <li>, <h1> to structure lists and headings
- <a> for navigation links across pages

The HTML files serve as templates for displaying user-submitted data when connected to Flask.

Index.html (welcome page)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Welcome | Portfolio</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='index-style.css') }}">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
  <div class="welcome-box">
    <h1>Welcome to Portfolio Builder</h1>
    <form action="/form" method="get">
      <button type="submit">Start</button>
    </form>
  </div>
</body>
</html>
```

Form.html

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

```
<head>
  <meta charset="UTF-8">
  <title>Portfolio Form</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='form-style.css') }}">
  <script src="{{ url_for('static', filename='form-script.js') }}" defer></script>
</head>
<body>
  <div class="form-container">
    <h2>Create Your Portfolio</h2>
    <form method="POST" enctype="multipart/form-data" action="/generate">
      <label>Full Name:</label>
      <input type="text" name="name" required>

      <label>Education:</label>
      <input type="text" name="education" required>

      <label>Skills:</label>
      <input type="text" name="skills" required>

      <div id="projects-section">
        <div class="project-block">
          <label for="projectName1">Project 1 Title:</label>
          <input type="text" id="projectName1" name="project[]" required>

          <label for="projectDesc1">Project 1 Description:</label>
          <textarea id="projectDesc1" name="description[]" required></textarea>
        </div>
      </div>

      <button type="button" onclick="addProject()">+ Add Another Project</button>

      <label>Upload Photo:</label>
      <input type="file" name="photo" accept="image/*" required>

      <label>Upload Resume (PDF):</label>
      <input type="file" name="resume" accept=".pdf" required>

      <label>Email:</label>
      <input type="email" name="email" required>

      <label>LinkedIn URL:</label>
      <input type="url" name="linkedin">

      <label>GitHub URL:</label>
      <input type="url" name="github">

      <label>Contact Number:</label>
      <input type="tel" name="contact" required>

      <label>About Yourself:</label>
```

```
        <textarea name="about" required></textarea>

        <div class="submit-btn-container">
          <button type="submit">Generate Portfolio</button>
        </div>
      </form>
    </div>
 </body>
 </html>
```

Home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Home - {{ name }}</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='home-style.css') }}">
</head>
<body>
  <nav class="navbar">
    <ul class="nav-links">
      <li><a href="/home">Home</a></li>
      <li><a href="/about">About</a></li>
      <li><a href="/skills">Skills</a></li>
      <li><a href="/projects">Projects</a></li>
      <li><a href="/contact">Contact</a></li>
    </ul>
  </nav>

  <section class="hero">
    {% if photo %}
    <img src="{{ url_for('static', filename='uploads/' + photo) }}" alt="Profile Photo" class="profile-photo">
    {% endif %}
    <h1>Hi, I'm {{ name }}</h1>
    <p class="tagline">Aspiring professional with a passion for technology and innovation.</p>
  </section>

  <section class="about">
    <h2>Description</h2>
    <p>{{ about }}</p>
  </section>
</body>
</html>
```

About.html

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

```
<head>
  <meta charset="UTF-8">
  <title>About - {{ name }}</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='about-style.css') }}">
</head>
<body>
  <nav>
    <ul>
      <li><a href="/home">Home</a></li>
      <li><a href="/about" class="active">About</a></li>
      <li><a href="/skills">Skills</a></li>
      <li><a href="/projects">Projects</a></li>
      <li><a href="/contact">Contact</a></li>
    </ul>
  </nav>

  <div class="container">
    <div class="profile">
      {% if photo %}
      <img src="{{ url_for('static', filename='uploads/' + photo) }}" alt="Profile Photo" class="profile-pic">
      {% endif %}
      <h1>{{ name }}</h1>
    </div>

    <div class="section">
      <h2>Education</h2>
      <p>{{ education }}</p>
    </div>

    <div class="section">
      <h2>Links</h2>
      <ul>
        <li>
          <b>GitHub: </b>
           <a href="{{ github }}" target="_blank">{{ github }}</a>
        </li>
        <li>
           <b>LinkedIn: </b>
           <a href="{{ linkedin }}" target="_blank">{{ linkedin }}</a>
        </li>
      </ul>
    </div>

    <div class="section">
      <h2>Resume</h2>
      {% if resume %}
        <a href="{{ url_for('static', filename='uploads/' + resume) }}"
          target="_blank"
          class="resume-btn">
          🔍 View Resume
```

```
            </a>
        {% endif %}
    </div>

  </div>
</body>
</html>
```

Skills.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Skills - {{ name }}</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='skills-style.css') }}">
</head>
<body>
  <nav>
    <ul>
      <li><a href="/home">Home</a></li>
      <li><a href="/about">About</a></li>
      <li><a href="/skills" class="active">Skills</a></li>
      <li><a href="/projects">Projects</a></li>
      <li><a href="/contact">Contact</a></li>
    </ul>
  </nav>

  <div class="container">
    <div class="header">
      <h1>{{ name }}'s Skills</h1>
    </div>

    <div class="skills-section">
      {% if skills %}
        <ul class="skills-list">
          {% for skill in skills.split(',') %}
            <li>{{ skill.strip() }}</li>
          {% endfor %}
        </ul>
      {% else %}
        <p>No skills provided.</p>
      {% endif %}
    </div>
  </div>
</body>
</html>
```

Projects.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Projects</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='projects-style.css') }}">
</head>
<body>
  <nav>
    <ul>
      <li><a href="{{ url_for('home') }}">Home</a></li>
      <li><a href="{{ url_for('about') }}">About</a></li>
      <li><a href="{{ url_for('skills') }}">Skills</a></li>
      <li><a href="{{ url_for('projects') }}" class="active">Projects</a></li>
      <li><a href="{{ url_for('contact') }}">Contact</a></li>
    </ul>
  </nav>

  <div class="container">
    <h1>My Projects</h1>
    <div class="projects">
      {% for project in projects %}
        <div class="project-card">
          <h3>{{ project }}</h3>
          <p>{{ descriptions[loop.index0] }}</p>
        </div>
      {% endfor %}
    </div>
  </div>
</body>
</html>
```

Contact.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Contact | Portfolio</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='contact-style.css') }}">
</head>
<body>
    <nav>
        <ul>
            <li><a href="{{ url_for('home') }}">Home</a></li>
            <li><a href="{{ url_for('about') }}">About</a></li>
```

```
        <li><a href="{{ url_for('skills') }}">Skills</a></li>
        <li><a href="{{ url_for('projects') }}">Projects</a></li>
        <li><a href="{{ url_for('contact') }}" class="active">Contact</a></li>
      </ul>
    </nav>

    <div class="container">
      <h1>Contact Information</h1>

      <div class="contact-info">
        <p><strong>Name:</strong> {{ name }}</p>
        <p><strong>Email:</strong> <a href="mailto:{{ email }}">{{ email }}</a></p>
        <p><strong>Phone:</strong> {{ contact }}</p>
      </div>

      <div class="message">
        <p>Thank you for taking the time to explore my portfolio. If you think we could create something amazing
together or if you have any opportunities that align with my skills, I'd love to hear from you.</p>
        <p>Feel free to reach out-Let's Build Something Great Together!!</p>
      </div>
    </div>
</body>
</html>
```

◇ 2. CSS – Styling

CSS enhances the **visual appearance** of the website. It is used to:

- Apply a **modern color scheme** and consistent font style
- Add **shadows, rounded borders, hover effects**, and animations
- Style the navigation bar, buttons, and contact form
- Make the layout **responsive** across devices using media queries
- Use **custom classes** to keep design reusable and clean

A separate .css file is linked to the HTML using the <link> tag for maintaining modularity.

Index-style.css

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Segoe UI', sans-serif;
}

body {
    background-image: url('https://images.unsplash.com/photo-1519389950473-47ba0277781c');
    background-size: cover;
    background-position: center;
    height: 100vh;
```

```
    display: flex;
    justify-content: center;
    align-items: center;
}

.welcome-box {
    background-color: rgba(255, 255, 255, 0.6);
    padding: 60px 90px;
    border-radius: 25px;
    text-align: center;
    box-shadow: 0 10px 25px rgba(0, 0, 0, 0.35);
    transition: transform 0.4s ease, background-color 0.3s ease;
    max-width: 600px;
}
.welcome-box h1 {
    font-size: 42px;
    margin-bottom: 20px;
    color: #222;
}
.welcome-box p {
    font-size: 18px;
    margin-bottom: 30px;
    color: #444;
}
.welcome-box button {
    padding: 14px 32px;
    font-size: 18px;
    border: none;
    border-radius: 8px;
    background-color: #007BFF;
    color: white;
    cursor: pointer;
    transition: background-color 0.3s ease;
}
.welcome-box button:hover {
    background-color: #0056b3;
    transform: scale(1.05);
}
```

Form-style.css

```
/* static/form-style.css */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Segoe UI', sans-serif;
}
body {
    background-image: url('https://images.unsplash.com/photo-1519389950473-47ba0277781c');
```

```
    background-repeat: no-repeat;
    background-size: cover;
    background-position: center;
    background-attachment: fixed;
    min-height: 100vh;
    padding: 40px;
    background-color: #f4f4f4;
}
.form-container {
    background-color: rgba(255, 255, 255, 0.75); /* Less opaque */
    max-width: 750px;
    margin: auto;
    padding: 50px 40px;
    border-radius: 20px;
    box-shadow: 0 12px 28px rgba(0, 0, 0, 0.2);
    backdrop-filter: blur(10px); /* More blur */
    -webkit-backdrop-filter: blur(10px); /* Safari support */
}
.form-container h2 {
    text-align: center;
    margin-bottom: 35px;
    color: #333;
    font-size: 32px;
    letter-spacing: 0.5px;
}
form label {
    display: block;
    margin-top: 20px;
    font-weight: 600;
    color: #444;
    font-size: 16px;
}
form input,
form textarea {
    width: 100%;
    padding: 12px;
    margin-top: 8px;
    border: 1px solid #ccc;
    border-radius: 10px;
    font-size: 15px;
    transition: border-color 0.3s ease;
}
form input:focus,
form textarea:focus {
    border-color: #007BFF;
    outline: none;
}
form textarea {
    resize: vertical;
    height: 90px;
```

```
}
button[type="button"] {
    margin-top: 20px;
    background-color: #17a2b8;
    color: white;
    border: none;
    padding: 10px 20px;
    font-size: 15px;
    border-radius: 8px;
    cursor: pointer;
    transition: background-color 0.3s ease;
}
button[type="button"]:hover {
    background-color: #138496;
}
.submit-btn-container {
    text-align: center;
    margin-top: 40px;
}
.submit-btn-container button {
    padding: 14px 42px;
    font-size: 18px;
    border: none;
    border-radius: 12px;
    background-color: #28a745;
    color: white;
    cursor: pointer;
    transition: background-color 0.3s ease;
}

.submit-btn-container button:hover {
    background-color: #218838;
}
```

Home-style.css

```
body {
    margin: 0;
    padding: 0;
    font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
    background-color: #f9f9f9;
    color: #333;
}
.navbar {
    background-color: #0a192f;
    padding: 15px 40px;
    position: sticky;
    top: 0;
    z-index: 1000;
}
```

```
.nav-links {
    list-style: none;
    display: flex;
    justify-content: flex-end;
    gap: 30px;
    margin: 0;
    padding: 0;
}
.nav-links li a {
    color: #ffffff;
    text-decoration: none;
    font-weight: 500;
    transition: color 0.3s ease;
}
.nav-links li a:hover {
    color: #64ffda;
}
.hero {
    text-align: center;
    padding: 50px 20px 20px;
    background-color: #ffffff;
    box-shadow: 0 4px 12px rgba(0, 0, 0, 0.05);
}
.profile-photo {
    width: 150px;
    height: 150px;
    border-radius: 50%;
    object-fit: cover;
    margin-bottom: 20px;
    border: 3px solid #64ffda;
}
.hero h1 {
    margin-bottom: 10px;
    font-size: 2.5rem;
}
.tagline {
    font-size: 1.1rem;
    color: #555;
}
.about {
    padding: 40px 60px;
    background-color: #f0f4f8;
    border-top: 1px solid #e0e0e0;
}
.about h2 {
    font-size: 2rem;
    margin-bottom: 20px;
    color: #0a192f;
}
.about p {
```

```
    font-size: 1.05rem;
    line-height: 1.6;
    color: #333;
    text-align: justify;
}
```

About-style.css

```
body {
    margin: 0;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    background-color: #f4f6f8;
    color: #2c3e50;
}
nav {
    background-color: #181818;
    padding: 16px 0;
    text-align: center;
    box-shadow: 0 2px 6px rgba(0, 0, 0, 0.12);
}
nav ul {
    list-style: none;
    margin: 0;
    padding: 0;
}
nav ul li {
    display: inline;
    margin: 0 24px;
}
nav ul li a {
    color: #cccccc;
    text-decoration: none;
    font-weight: 500;
    font-size: 16px;
    transition: color 0.3s ease;
}
nav ul li a:hover,
nav ul li a.active {
    color: #339af0;
}
.container {
    max-width: 960px;
    margin: 60px auto;
    background-color: #ffffff;
    padding: 50px 40px;
    border-radius: 16px;
    box-shadow: 0 8px 28px rgba(0, 0, 0, 0.06);
    transition: box-shadow 0.3s ease;
}
.profile {
```

```
    text-align: center;
    margin-bottom: 50px;
}
.profile-pic {
    width: 140px;
    height: 140px;
    object-fit: cover;
    border-radius: 50%;
    border: 4px solid #dee2e6;
    box-shadow: 0 6px 16px rgba(0, 0, 0, 0.08);
    transition: transform 0.3s ease;
}
.profile-pic:hover {
    transform: scale(1.06);
}
.profile h1 {
    margin-top: 18px;
    font-size: 28px;
    font-weight: 600;
    color: #212529;
}
.section {
    margin-bottom: 40px;
}
.section h2 {
    font-size: 20px;
    color: #2f3e4e;
    margin-bottom: 12px;
    border-bottom: 2px solid #e3e3e3;
    padding-bottom: 6px;
}
.section ul {
    padding-left: 0;
    list-style: none;
}
.section li {
    font-size: 15px;
    margin-bottom: 12px;
    word-wrap: break-word;
    color: #495057;
}
.section a {
    display: inline-block;
    max-width: 100%;
    overflow-wrap: break-word;
    background-color: #eef3f7;
    color: #1a73e8;
    padding: 7px 12px;
    border-radius: 6px;
    font-weight: 500;
```

```
  text-decoration: none;
   transition: all 0.3s ease;
}
.section a:hover {
   background-color: #dce7f4;
   color: #0a58ca;
}
.resume-btn {
   display: inline-block;
   margin-top: 10px;
   padding: 10px 20px;
   background-color: #1a73e8;
   color: white;
   border-radius: 8px;
   text-decoration: none;
   font-weight: 500;
   transition: background-color 0.3s ease;
}
.resume-btn:hover {
   background-color: #1558c0;
}
```

Skills-style.css

```
body {
   margin: 0;
   font-family: 'Segoe UI', sans-serif;
   background: #f5f7fa;
   color: #333;
}
nav {
   background: #111;
   padding: 16px 0;
   text-align: center;
}
nav ul {
   list-style: none;
   margin: 0;
   padding: 0;
nav ul li {
   display: inline;
   margin: 0 20px;
}
nav ul li a {
   color: #bbb;
  text-decoration: none;
   font-weight: 500;
   transition: color 0.3s;
}
nav ul li a:hover,
```

```
nav ul li a.active {
    color: #1a73e8;
}
.container {
    max-width: 900px;
    margin: 60px auto;
    background: #fff;
    padding: 40px;
    border-radius: 16px;
    box-shadow: 0 8px 24px rgba(0, 0, 0, 0.08);
    text-align: center;
}
h1 {
    margin-bottom: 30px;
    font-size: 28px;
    color: #222;
}
.skills-list {
    padding: 0;
    list-style: none; /* Removes bullets or numbers */
    text-align: left;
    margin: 0 auto;
    max-width: 300px; /* Keeps the list neat and centered */
}
.skills-list li {
    background: #1a73e8;
    color: #fff;
    padding: 10px 18px;
    border-radius: 12px;
    font-size: 14px;
    font-weight: 500;
    box-shadow: 0 2px 6px rgba(0,0,0,0.1);
    margin-bottom: 10px;
    transition: transform 0.2s;
}
.skills-list li:hover {
    transform: scale(1.05);
}
```

Projects-style.css

```
body {
    margin: 0;
    font-family: 'Segoe UI', sans-serif;
    background-color: #f0f2f5;
    color: #333;
}
nav {
    background: #101820;
   padding: 16px 0;
```

```
 text-align: center;
}
nav ul {
    list-style: none;
    margin: 0;
    padding: 0;

nav ul li {
    display: inline-block;
    margin: 0 20px;
}
nav ul li a {
    color: #cdd6f4;
    text-decoration: none;
    font-weight: 500;
    transition: color 0.3s;
}
nav ul li a:hover,
nav ul li a.active {
    color: #4da3ff;
}
.container {
    max-width: 1200px;
    margin: 80px auto;
    padding: 50px 40px;
    background: #ffffff;
    border-radius: 20px;
    box-shadow: 0 12px 30px rgba(0, 0, 0, 0.06);
}
h1 {
    text-align: center;
    margin-bottom: 40px;
    font-size: 34px;
    font-weight: 600;
    color: #1a1a1a;
    position: relative;
}
h1::after {
    content: '';
    display: block;
    width: 60px;
    height: 3px;
    background: #4da3ff;
    margin: 12px auto 0;
    border-radius: 5px;
}
.projects-wrapper {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(280px, 1fr));
    gap: 24px;}
```

```
.project-card {
    background: #f9fbff;
    border: 1px solid #e3e9f2;
    padding: 24px;
    border-radius: 14px;
    transition: all 0.3s ease;
    text-align: left;
    box-shadow: 0 4px 12px rgba(0,0,0,0.04);
}
.project-card:hover {
    transform: translateY(-6px);
    box-shadow: 0 8px 20px rgba(0, 0, 0, 0.1);
}
.project-card h2 {
    margin-top: 0;
    font-size: 20px;
    color: #1a73e8;
    margin-bottom: 10px;
}
.project-card p {
    font-size: 15px;
    line-height: 1.6;
    color: #555;
}
.project-block {
    background: #f9fbff;
    border: 1px solid #d6e2f1;
    border-left: 5px solid #4da3ff;
    padding: 20px;
    margin-bottom: 20px;
    border-radius: 12px;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.05);
}
.project-block label {
    display: block;
    margin-top: 10px;
    font-weight: 500;
    color: #333;
}
.project-block input,
.project-block textarea {
    width: 100%;
    padding: 10px;
    margin-top: 6px;
    border: 1px solid #ccc;
    border-radius: 6px;
    font-size: 14px;
    background-color: #fff;
}
.remove-btn {
```

```
    float: right;
    background: #ff4d4d;
    border: none;
    color: #fff;
    font-weight: bold;
    padding: 4px 10px;
    border-radius: 50%;
    cursor: pointer;
    font-size: 14px;
}
```

Contact-style.css

```
body {
    font-family: 'Segoe UI', sans-serif;
    margin: 0;
    background-color: #f8f9fc;
    color: #333;

nav {
    background-color: #101820;
    padding: 16px 0;
    text-align: center;
}
nav ul {
    list-style: none;
    padding: 0;
    margin: 0;
}
nav ul li {
    display: inline;
    margin: 0 15px;
}
nav ul li a {
    color: #cdd6f4;
    text-decoration: none;
    font-weight: 500;
}
nav ul li a.active,
nav ul li a:hover {
    color: #4da3ff;
}
.container {
    max-width: 800px;
    margin: 60px auto;
    background: #fff;
    padding: 40px;
    border-radius: 12px;
    box-shadow: 0 6px 18px rgba(0, 0, 0, 0.06);
}h1 {
```

```
    text-align: center;
    margin-bottom: 30px;
    font-size: 30px;
    color: #1a1a1a;
}
.contact-info p {
    font-size: 16px;
    margin: 12px 0;
    line-height: 1.5;
}
.message {
    margin: 40px 0 20px;
    font-size: 16px;
    text-align: center;
    color: #444;
    line-height: 1.6;
}
footer {
    margin-top: 40px;
    text-align: center;
    padding: 16px;
    background-color: #f0f2f5;
    font-size: 14px;
    color: #666;
}
```

◇ 3. Python – Backend Logic

Python is used to power the server-side logic of the project. It is responsible for:

- Handling user input submitted through the contact form
- Processing data and storing it temporarily or displaying it on the profile page
- Controlling the flow of data between the form and the HTML templates
- Organizing the project structure for easy development and testing

Python makes the web app dynamic and able to respond to user interactions.

◇ 4. Flask – Web Framework

Flask is used to build the **web server** and manage routing between different pages. The main tasks performed using Flask are:

- Defining **routes** like /home, /about, /contact, etc.
- Rendering templates using render_template()
- Handling POST requests to capture form submissions
- Using **Jinja2** syntax like {{ user_name }} to display dynamic content
- Serving static files like CSS and images using Flask's static folder

Flask connects everything—HTML for structure, CSS for design, and Python for logic.

app.py

```
from flask import Flask, render_template, request, redirect, url_for, session
from werkzeug.utils import secure_filename
import os
app = Flask(__name__)
app.secret_key = 'your_secret_key'
UPLOAD_FOLDER = 'static/uploads'
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
os.makedirs(UPLOAD_FOLDER, exist_ok=True)
@app.route('/')
def welcome():
    return render_template('index.html')
@app.route('/form')
def form():
    return render_template('form.html')
@app.route('/generate', methods=['POST'])
def generate():
    # Collecting form data
    name = request.form.get('name')
    education = request.form.get('education')
    skills = request.form.get('skills')
    projects = request.form.getlist('project[]')
    descriptions = request.form.getlist('description[]')
    email = request.form.get('email')
    linkedin = request.form.get('linkedin')
    github = request.form.get('github')
    contact = request.form.get('contact')
    about = request.form.get('about')
    # Handling file uploads
    photo = request.files.get('photo')
    resume = request.files.get('resume')
    if photo:
        photo_filename = secure_filename(photo.filename)
        photo_path = os.path.join(app.config['UPLOAD_FOLDER'], photo_filename)
        photo.save(photo_path)
        session['photo'] = photo_filename
    else:
        session['photo'] = None
    if resume:
        resume_filename = secure_filename(resume.filename)
        resume_path = os.path.join(app.config['UPLOAD_FOLDER'], resume_filename)
        resume.save(resume_path)
        session['resume'] = resume_filename
    else:
        session['resume'] = None
    # Storing other details in session
    session['name'] = name
    session['education'] = education
    session['skills'] = skills
```

```
    session['projects'] = projects
    session['descriptions'] = descriptions
    session['email'] = email
    session['linkedin'] = linkedin
    session['github'] = github
    session['contact'] = contact
    session['about'] = about
    return redirect(url_for('home'))
@app.route('/home')
def home():
    return render_template('home.html',
       name=session.get('name', ''),
        education=session.get('education', ''),
        skills=session.get('skills', ''),
        projects=zip(session.get('projects', []), session.get('descriptions', [])),
        email=session.get('email', ''),
        linkedin=session.get('linkedin', ''),
        github=session.get('github', ''),
        contact=session.get('contact', ''),
        about=session.get('about', ''),
        photo=session.get('photo'),
        resume=session.get('resume')  )
@app.route('/about')
def about():
    return render_template('about.html',
        name=session.get('name', ''),
        education=session.get('education', ''),
        linkedin=session.get('linkedin', ''),
        github=session.get('github', ''),
        photo=session.get('photo', ''),
        resume=session.get('resume','')  )
@app.route('/skills')
def skills():
    return render_template(
        'skills.html',
        name=session.get('name', ''),
        skills=session.get('skills', '')   )
@app.route('/projects')
def projects():
    return render_template(
        'projects.html',
        projects=session.get('projects', []),
        descriptions=session.get('descriptions', [])  )
@app.route('/contact')
def contact():
    return render_template('contact.html',
        name=session.get('name', ''),
        email=session.get('email', ''),
        contact=session.get('contact', ''),   )
if __name__ == '__main__':
```

```
app.run(debug=True)
```

◇ JavaScript – Frontend Interactivity

JavaScript is used to bring interactivity and dynamic behavior to the frontend of the project. It is responsible for:

- Enhancing the user experience with interactive UI elements
- Handling events like button clicks, form submissions, and menu toggles
- Validating form inputs on the client-side before they are sent to the server
- Enabling smooth scrolling and transitions between sections
- Dynamically updating content without reloading the page (if needed)

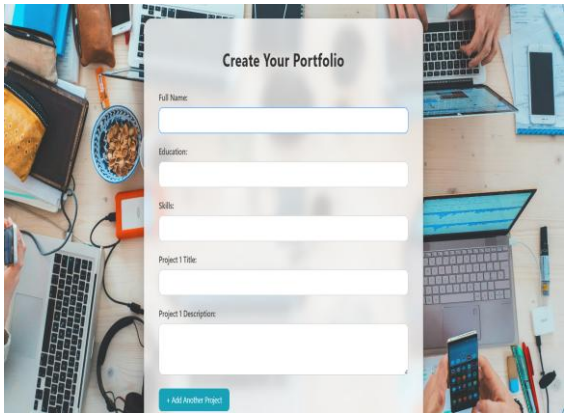
JavaScript ensures that the website feels more responsive, engaging, and user-friendly.

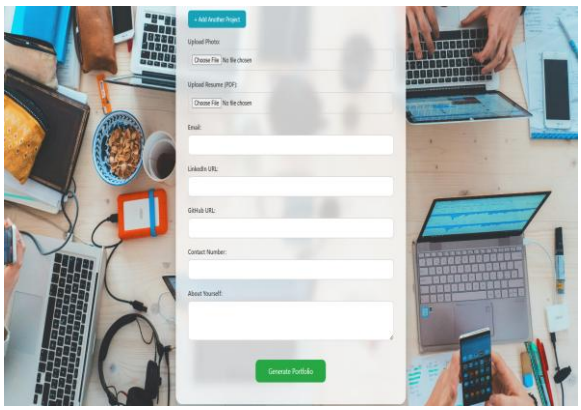
form-script.js

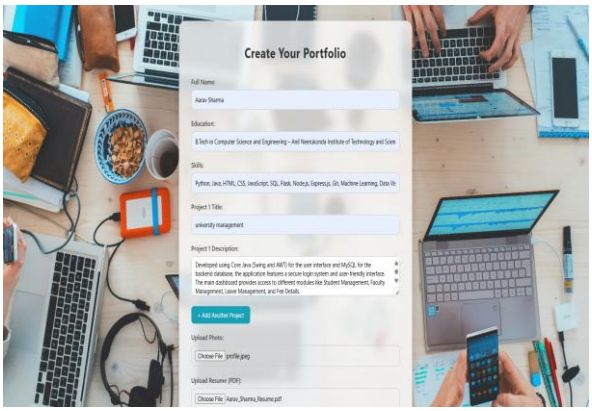
```
function addProject() {
  const container = document.getElementById("projects-section");
   const projectNumber = container.querySelectorAll(".project-block").length + 1;
   const projectBlock = document.createElement("div");
   projectBlock.className = "project-block";
   projectBlock.innerHTML = `
      <button type="button" class="remove-btn" onclick="removeProject(this)">×</button>
      <label for="projectName${projectNumber}">Project ${projectNumber} Title:</label>
      <input type="text" id="projectName${projectNumber}" name="project[]" required
      <label for="projectDesc${projectNumber}">Project ${projectNumber} Description:</label>
      <textarea id="projectDesc${projectNumber}" name="description[]" required></textarea>
   container.appendChild(projectBlock);
}
function removeProject(button) {
   const projectBlock = button.parentElement;
   projectBlock.remove();
   const container = document.getElementById("projects-section");
   const blocks = container.querySelectorAll(".project-block");
   blocks.forEach((block, index) => {
```

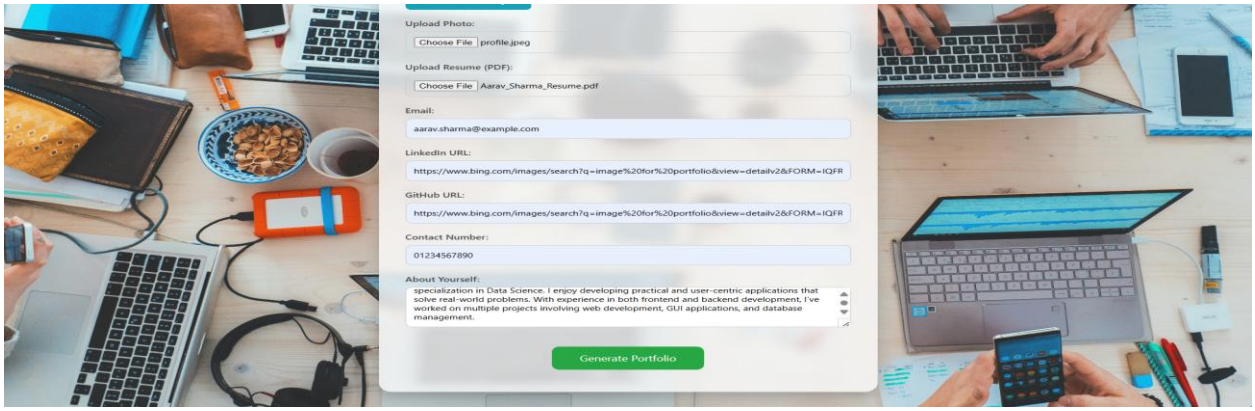
```
        const number = index + 1;
        block.querySelectorAll("label")[0].textContent = `Project ${number} Title:`;
        block.querySelectorAll("input")[0].id = `projectName${number}`;
        block.querySelectorAll("textarea")[0].id = `projectDesc${number}`;
        block.querySelectorAll("label")[1].textContent = `Project ${number} Description:`;
    });
}
```

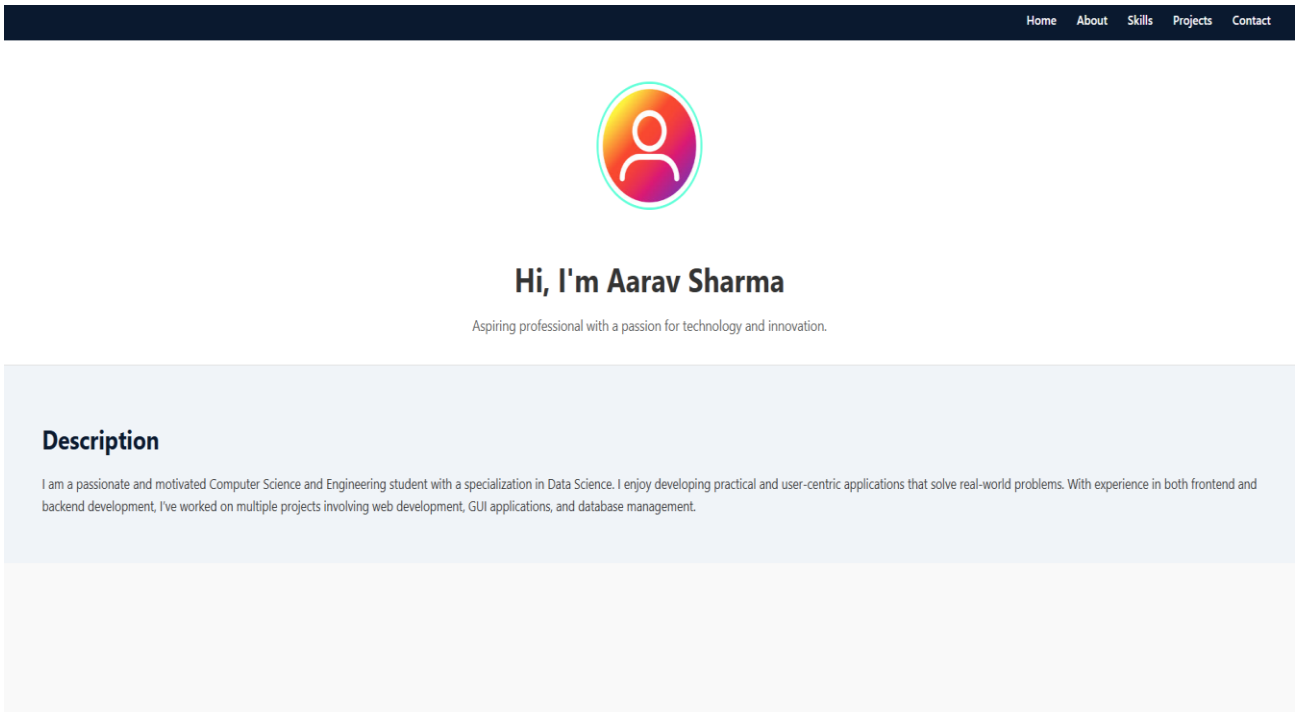
OUTPUT:

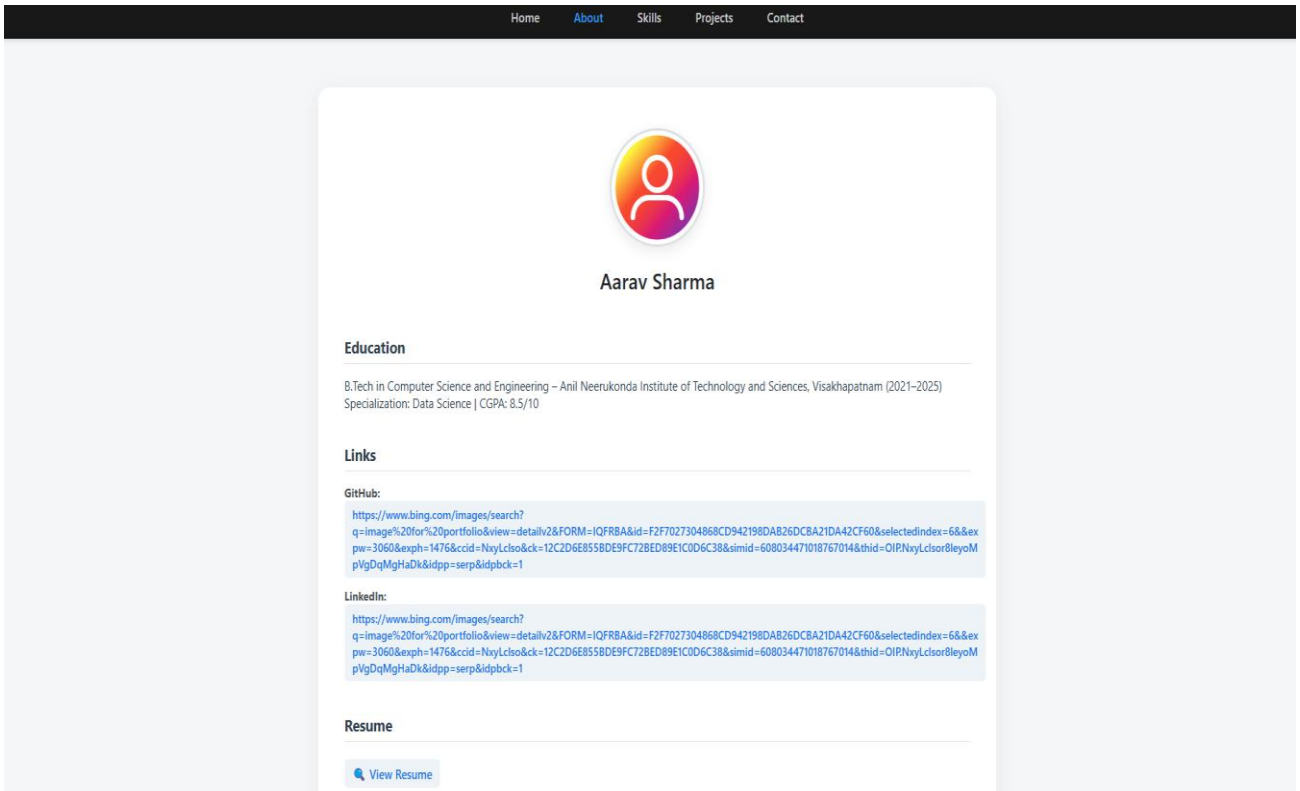









Home
About
Skills
Projects
Contact
Hi, I'm Aarav Sharma
Aspiring professional with a passion for technology and innovation.
Description
I am a passionate and motivated Computer Science and Engineering student with a specialization in Data Science. I enjoy developing practical and user-centric applications that solve real-world problems. With experience in both frontend and backend development, I've worked on multiple projects involving web development, GUI applications, and database management.


Home
About
Skills
Projects
Contact
Aarav Sharma
Education
B.Tech in Computer Science and Engineering – Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam (2021–2025)
Specialization: Data Science | CGPA: 8.5/10
Links
GitHub:
https://www.bing.com/images/search?q=image%20for%20portfolio&view=detailv2&FORM=IQFRBA&id=F2F7027304868CD942198DAB26DCBA21DA42CF60&selectedindex=6&&expw=3060&exph=1476&ccid=NxyLclso&ck=12C2D6E855BDE9FC72BED89E1C0D6C38&simid=608034471018767014&thid=OIP.NxyLclsor8leyoMpVgDqMgHaDk&idpp=serp&idpbck=1
LinkedIn:
https://www.bing.com/images/search?q=image%20for%20portfolio&view=detailv2&FORM=IQFRBA&id=F2F7027304868CD942198DAB26DCBA21DA42CF60&selectedindex=6&&expw=3060&exph=1476&ccid=NxyLclso&ck=12C2D6E855BDE9FC72BED89E1C0D6C38&simid=608034471018767014&thid=OIP.NxyLclsor8leyoMpVgDqMgHaDk&idpp=serp&idpbck=1
Resume
View Resume

Aarav Sharma

Email: aarav.sharma@example.com | Phone: +91 98765 43210

LinkedIn: https://www.linkedin.com/in/aarav-sharma | GitHub: https://github.com/aaravsharma

Education

B.Tech in Computer Science and Engineering
Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam (2021–2025)
Specialization: Data Science | CGPA: 8.5/10

Skills

Python, Java, HTML, CSS, JavaScript, SQL, Flask, Node.js, Express.js, Git, Machine Learning, Data Visualization

Projects

- FarmConnect – A Farmer-to-Consumer Platform

Designed and developed a platform that allows farmers to sell their products directly to consumers, eliminating middlemen. Built using Python, Tkinter for GUI, and SQLite for backend. Enabled farmers to upload product details and customers to browse and purchase efficiently.

- University Management System

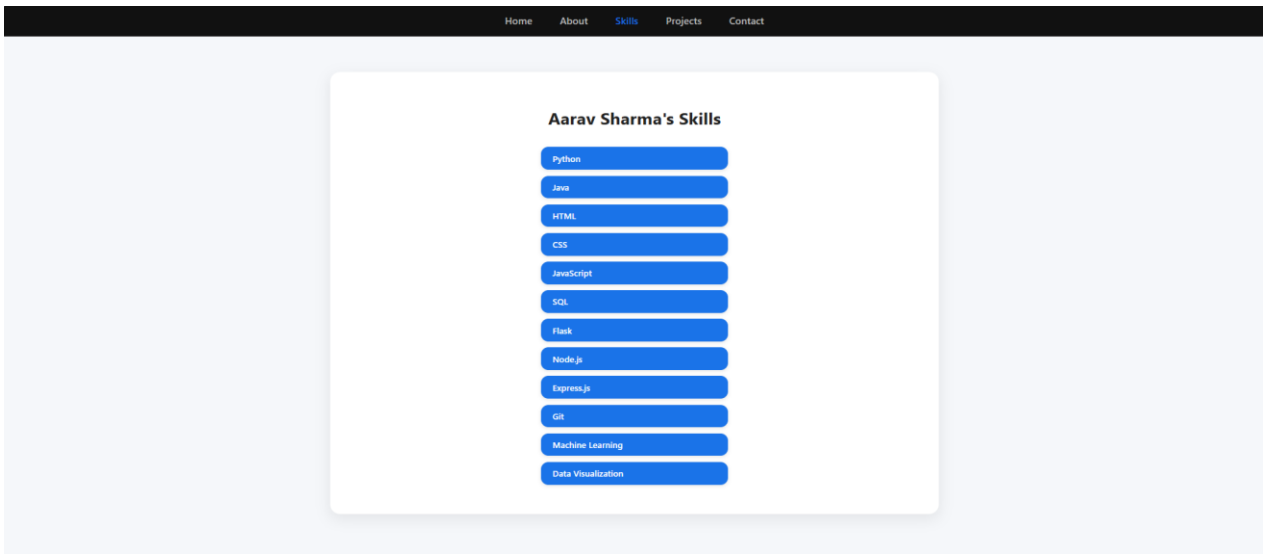
Built a Java application using Swing and AWT for managing student and faculty records. Integrated MySQL for storing data related to fees, leaves, and personal details. Enabled efficient data entry, update, and retrieval by administrators.

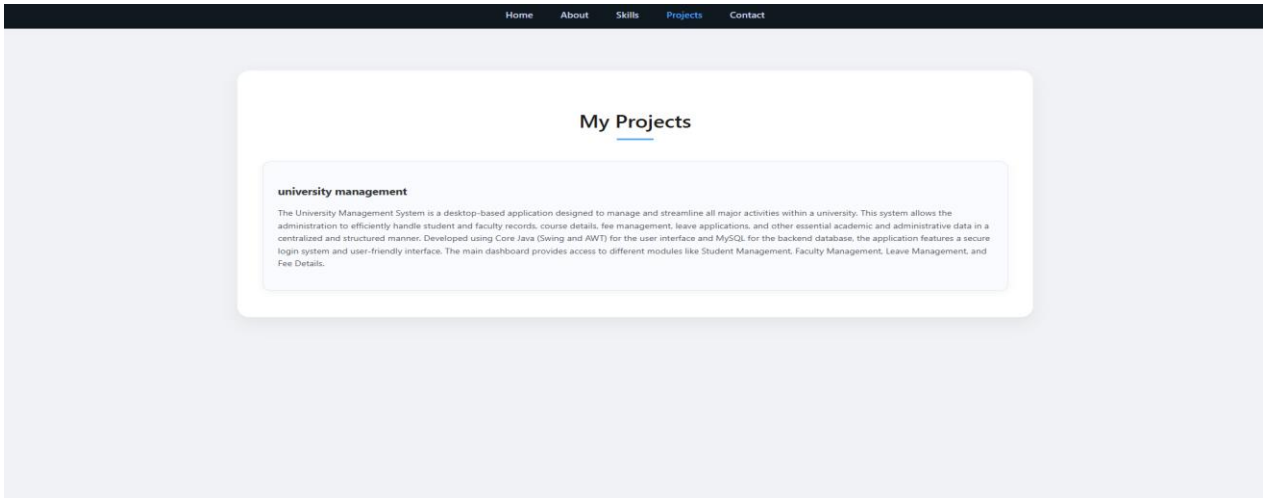
- Portfolio Generator Web App

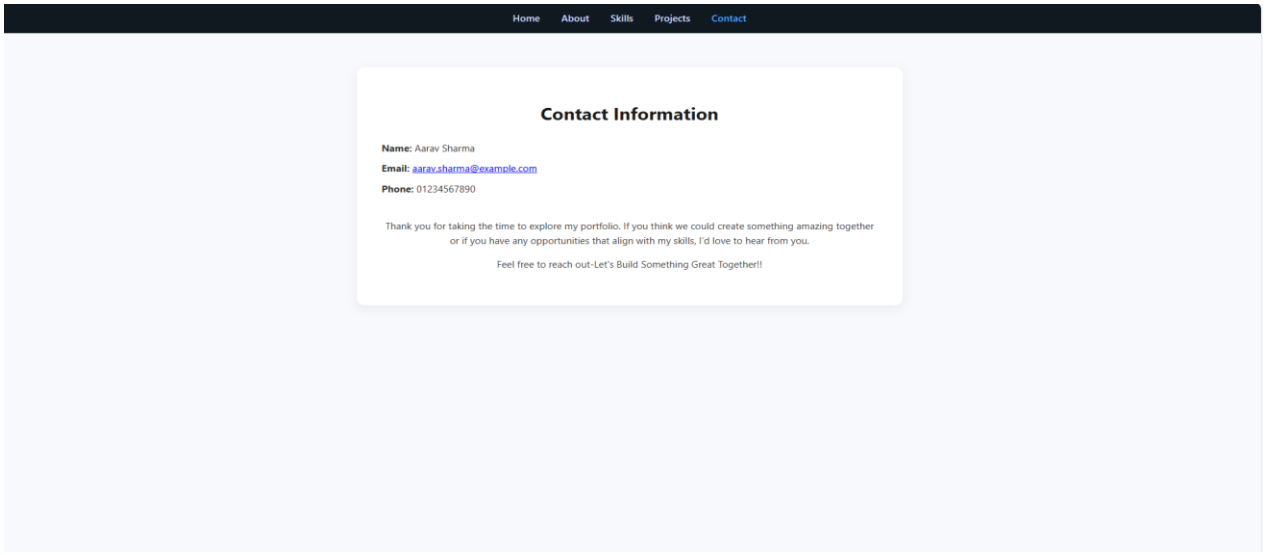
Developed a dynamic portfolio generator web application using HTML, CSS, JavaScript for the frontend, and Flask for the backend. The app allows users to enter personal, educational, and project information through a form and generates a personalized portfolio with a clean, interactive layout.

Profile Summary

A passionate and results-driven Computer Science student specializing in Data Science. With hands-on experience in full-stack development, database management, and machine learning, I enjoy building practical solutions to real-world problems. I thrive in collaborative environments and am always eager to learn and grow.







CONCLUSION:

This portfolio project serves as a comprehensive example of full-stack web development, combining key front-end and back-end technologies to build a fully functional, user-interactive website. It effectively demonstrates how HTML provides the structure, CSS brings visual appeal, JavaScript adds interactivity, and Python with Flask handles backend logic and routing. The seamless integration of these technologies results in a smooth and responsive user experience.

The project allows users to submit their personal details such as name, skills, education, projects, and contact information through a structured form. Upon submission, the entered data is dynamically rendered and displayed across multiple well-organized pages like Home, About, Skills, Projects, and Contact. This not only ensures data-driven behavior but also reflects a real-world approach to building personalized web applications.

By implementing proper navigation, professional UI/UX design, and efficient routing mechanisms, the project ensures a consistent and engaging interface. It also highlights the developer's skills in handling form data, dynamic content rendering, responsive design, and user-focused web application development.

In conclusion, this portfolio project not only strengthens technical knowledge but also builds confidence in deploying real-time, user-based applications—making it a valuable addition to any developer's skill set and resume.