COLLEGE CODE: 9623

COLLEGE NAME: AMRITA COLLEGE OF
ENGINEERING AND TECHNOLOGY

DEPARTMENT: COMPUTER SCIENCE AND
ENGINEERING

STUDENT NM-ID:
80E44E07DF9A7519D2F4674C2BBBED5D

ROLL NO: 962323104005

DATE: 12-09-2025

COMPLETED THE PROJECT NAMED AS PHASE 2

TECHNOLOGY PROJECT NAME: PORTFOLIO WEBSITE

SUBMITTED BY,

NAME: S.ABIRAJ

MOBILE.NO:6383668359

Tech Stack Selection

To build a scalable and responsive portfolio website, the following tech stack is selected:

- **Frontend:**
  - HTML5, CSS3, JavaScript (ES6+) – For structure, styling, and interactivity.
  - React.js – For building reusable, dynamic, and component-based UI.
  - Tailwind CSS / Bootstrap – For responsive and modern styling.

- **Backend (Optional – if using dynamic content):**
  - Node.js with Express.js – To handle APIs and server-side logic.

- **Database (Optional – for blog/projects data):**
  - MongoDB / Firebase – To store project details, contact form entries, and blog posts.

- **Deployment & Hosting:**
  - GitHub Pages / Vercel / Netlify – For frontend hosting.
  - MongoDB Atlas / Firebase Cloud – For backend data storage

UI Structure / API Schema Design
UI Structure (Pages & Components)

1. Home Page – Introduction, name, and tagline.
2. About Page – Bio, skills, education, and experience.
3. Projects Page – Portfolio of projects with images, descriptions, and links.
4. Blog Page (Optional) – Articles and learning experiences.
5. Contact Page – Contact form, email, and social links.
6. Navigation Bar & Footer – Persistent across all pages.

API Schema (if backend included)

- o /api/projects → Returns list of portfolio projects.
- o /api/contact → Handles form submissions (name, email, message).
- o /api/blogs → Returns blog posts (title, content, tags)

Data Handling Approach Static Data (Without Backend):

Static Data (Without Backend):

- o Store project and skill data in JSON files or React state.
- o Form submission Can use services like EmailJS or Google Forms.

Dynamic Data (With Backend):

- o API endpoints fetch project/blog data from MongoDB or Firebase.
- o Contact form data stored securely in the database.
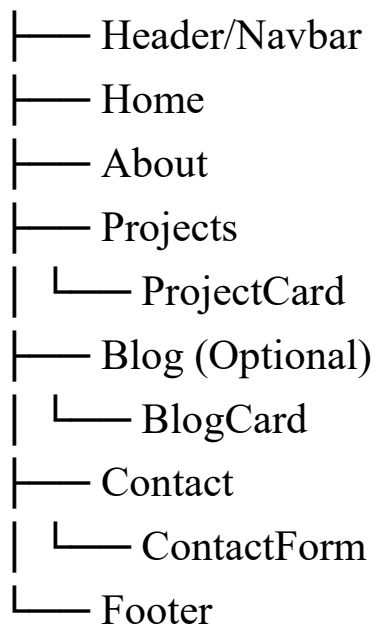- o Use JWT/OAuth for authentication (if admin panel is added)

Component / Module Diagram
Main Components:

- o Header (Navbar)
- o Footer
- o HomeSection
- o AboutSection

- o ProjectsSection
- o BlogSection
- o ContactForm
- o API Service Module (for data fetch & post)

Portfolio Website

```
├── Header/Navbar
├── Home
├── About
├── Projects
│   └── ProjectCard
├── Blog (Optional)
│   └── BlogCard
├── Contact
│   └── ContactForm
└── Footer
```

Basic Flow Diagram

User Flow:

```
[User]
  ↓
[Home Page] → [About Page] → [Projects Page] → [Blog Page] →
[Contact  Page]
        ↓                    ↓
   [Navbar] <--------------------------------> [Footer]
```

Contact Form Flow (with Backend):

User → Fills Contact Form → API (/api/contact) → Backend (Node.js)→Database (MongoDB)

→ Success/Failure Response → User Notification (Toast/Message)