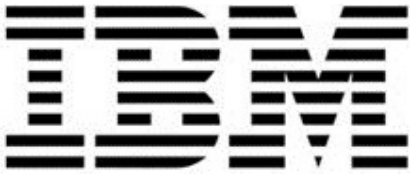


COLLEGE CODE: 9623

COLLEGE NAME: AMRITA COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING

STUDENT NM-ID:
F9F820B4B8EFF12A955428C2077D5ACA

ROLL NO: 962323104059

DATE: 12-09-2025

COMPLETED THE PROJECT NAMED AS PHASE 2

TECHNOLOGY PROJECT NAME: PORTFOLIO WEBSITE

SUBMITTED BY,

NAME: S. MITHUN KRSHNA

MOBILE.NO:6379567322

Tech Stack Selection

Tobuild a scalableandresponsiveportfolio 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)

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

Data Handling Approach Static Data (Without Backend):

Static Data (Without Backend):

- Store projectand skill data inJSONfilesor React state.
- Form submission Can use serviceslikeEmailJS or Google Forms.

Dynamic Data (With Backend):

- APIendpoints fetchproject/blogdatafrom MongoDB or Firebase.
- Contact form data stored securelyinthe database.
- Use JWT/OAuth for authentication(ifadmin panel is added)

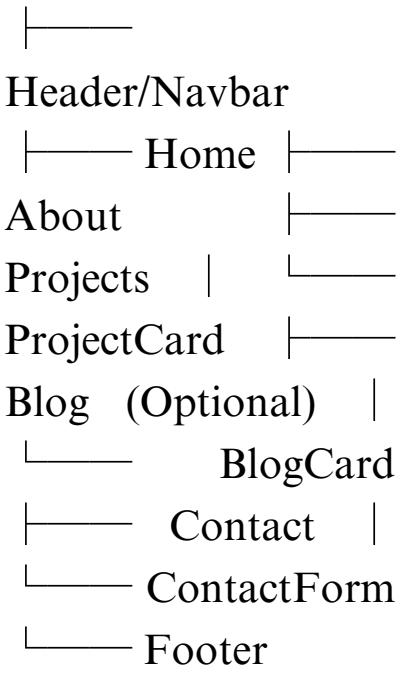
Component / Module Diagram

Main Components:

- Header (Navbar)
- Footer
- HomeSection
- AboutSection

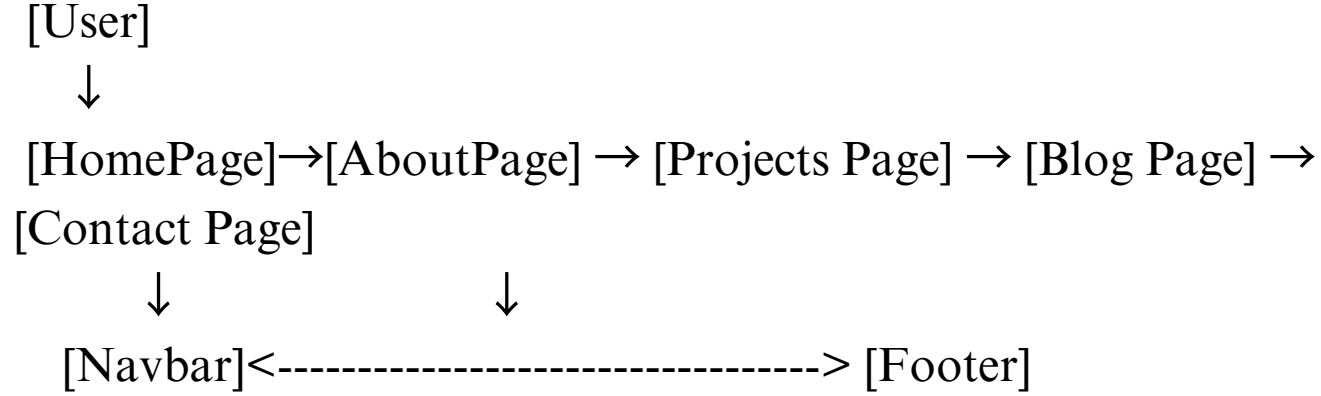
- ProjectsSection
- BlogSection
- ContactForm
- API Service Module (for data fetch & post)

Portfolio Website



BasicFlowDiagram

User Flow:



Contact Form Flow (with Backend):

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

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