COLLEGE CODE: 9623
COLLEGE NAME: AMRITA COLLEGE OF ENGINEERING AND TECHNOLOGY
DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
STUDENT NM-ID: 9463727837FA379BB3646D8E9B2C5BCF
ROLL NO: 962323104061
DATE: 22-09-2025

COMPLETED THE PROJECT NAMED AS PHASE 3

TECHNOLOGY PROJECT NAME: PORTFOLIO WEBSITE

SUBMITTED BY,

NAME:Y.MOHAMED FARIZ
MOBILE.NO:9385434283

1. Project Setup

The Portfolio Website project was set up in two major parts:
Frontend and Backend, along with supporting tools for development.

Frontend Setup:

The frontend was implemented using React.js for component-based development. The UI was styled with Tailwind CSS, enabling a clean, modern, and responsive design.
The structure of the project was divided into different reusable components:

- Navbar → Provides navigation across sections (Home, About, Projects, Contact).
- Home → Displays an introduction with name, role, and tagline.
- About → Highlights background, skills, and achievements.
- Projects → Showcases academic and personal projects dynamically.
- Contact → Contains a contact form for communication.
- Footer → Includes copyright and quick links.

Backend Setup:

The backend was implemented with Node.js and Express.js. This layer handled APIs to serve project data and store messages received from the contact form.

Development Environment & Tools:

- Editor: Visual Studio Code (VS Code)
- Version Control: Git + GitHub Repository
- Runtime Environment: Node.js (v18)
- Database: MongoDB (Cloud/Atlas)

- Testing Tool: Postman

2. Core Features Implementation

The core functionality of the portfolio website was carefully developed to match the Minimum Viable Product (MVP) requirements.

Frontend Features:

1. Home Section: Includes a welcoming headline and tagline.
2. About Section: Provides a short bio, academic background, and list of skills.
3. Projects Section: Dynamically loads project details (title, description, image, and GitHub/demolink).
4. Contact Section: Features a form with fields: Name, Email, and Message.
5. Responsive Design: Mobile-friendly design ensured.

Backend Features:

- Project API: Returns a list of projects stored in MongoDB.
- Contact API: Stores user queries into a database collection.

3. Data Storage (Local State / Database) Two

approaches were considered:

- Local State: During the early development phase, project data was stored in a JavaScript arrayinside the React application.
- Database (MongoDB): In the final implementation, data was shifted to MongoDB for persistence.

Projects Collection Example:

{ "title": "Portfolio Website", "description": "A personal portfolio website built with React and Node.js.", "link" :https://github.com/user/portfolio,"image": "portfolio.png" }

Contacts Collection Example:

{ "name": "John Doe", "email": "johndoe@gmail.com", "message": "Interested in your projects.", "date": "2025-09-20" }

4. Testing Core Features FrontendTesting:

- Navigation between sections was tested manually.
- The Projects section was checked for dynamic rendering.
- Contact form validated input fields.
- Responsive testing across devices.

Backend Testing:

- APIs tested with Postman.
- GET /api/projects returned correct project data.
- POST /api/contact stored user queries in MongoDB.

Integration Testing:

- Ensured frontend and backend worked together without issues.

5. Version Control (GitHub)

- Repository Setup: A GitHub repository was created for hosting code.
- Commit History: Initial commit (base setup), feature commits, final commit.
- Branching Strategy: main (stable), dev (active development).
- Deployment: Frontend deployed on GitHub Pages, Backend on Render/Heroku.

Conclusion of Phase 3

By the end of Phase 3, the Portfolio Website MVP was successfully implemented with:

- A functional frontend showcasing personal details, skills, projects, and contact form.
- A backend API connected to MongoDB.
- Testing ensured correctness, responsiveness, and reliability.
- Version-controlled repository with deployment.

This marks the completion of the MVP stage, making the project ready for further enhancements such as authentication and analytics.

-