COLLEGE CODE: 9528

COLLEGE NAME: SCAD COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT: COMPUTER SCIENCE ENGINEERING

STUDENTNMID: 38F4C5B02A28985899E4581E179929F8

Roll no : 952823104172

DATE : 24.10.2025

Completed the project named as:

Phase 5

TECHNOLOGYPROJECTNAME :

NODEJS BACKEND FOR CONTACT FORM

SUBMITTED By,

NAME: P.THANGASARANYA

MOBILE NO: 8883721303

Phase 5 - PROJECT DEMONSTRATION AND DOCUMENTATION

1.Final Demo Walkthrough

The Contact Form backend is designed to handle and store user queries securely.
It allows users to submit their name, email, and message, and processes this information through the backend using Node.js and Express.js.

The demo begins with the user navigating to the contact page, where fields such as name, email, subject, and message are displayed. Upon entering the details, the form performs client-side validation to ensure all mandatory fields are filled correctly. Once submitted, the frontend sends the form data to the backend API using a POST request.

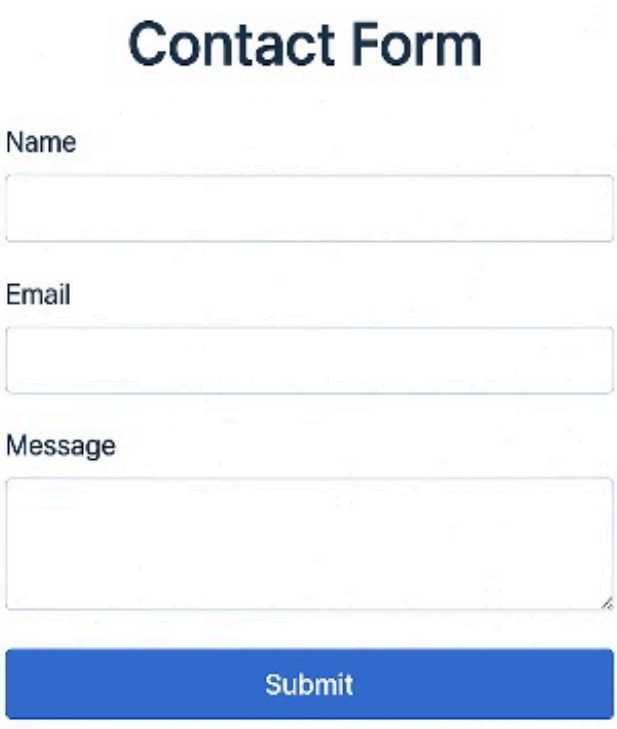
On the backend side, the Node.js and Express server receive the form data, validate it again for security, and either store it in a database or display it on the server console for confirmation. The backend then sends a response back to the frontend confirming the successful submission. This real-time data flow between the frontend and backend is highlighted during the demo to show the integration, validation, and communication layers working together efficiently.

2. Project Report

This project aims to create a fully functional contact form system where the frontend and backend communicate effectively through RESTful APIs. The goal is to demonstrate how user data from a simple web form can be processed, validated, and managed securely using a Node.js backend. The frontend provides an easy-to-use interface, while the backend handles the business logic, input sanitization, and response management.

The project uses Node.js and Express.js for the backend and HTML/CSS/JavaScript (or React) for the frontend. The backend ensures scalability, while middleware like body-parser and CORS provide smooth data handling. The project highlights how to build and deploy a full-stack system that could be extended for real-world business applications such as customer feedback collection or support ticket generation.

3.Screenshots / API Documentation



4.Challenges & Solutions

One major challenge faced during development was handling validation errors between the frontend and backend. Initially, some form submissions were failing because of incorrect or incomplete input data. To overcome this, input validation was implemented using both client-side checks and server-side middleware to ensure data accuracy. Another issue was managing CORS errors when connecting the frontend and backend hosted on different platforms. This was resolved by enabling the CORS package in the backend configuration.

Another challenge involved deployment and environment configuration. The backend required correct setup of environment variables and routes to work on cloud platforms like Render or Vercel. Debugging these issues helped improve understanding of deployment pipelines and environment setups. Testing the deployed version across multiple devices also ensured that the app worked consistently on both mobile and desktop platforms.

5.GitHub README & Setup Guide

The GitHub repository for the IBM-FE-NodeJS Backend for Contact Form contains all essential files, including the source code for both the frontend and backend, configuration details, and supporting documentation. The README file provides a clear overview of the project, outlining its purpose, technology stack, and core features. It also explains how the frontend contact form communicates with the Node.js backend through API calls, demonstrating the flow of data from form submission to backend processing. This documentation serves as a quick reference for developers or reviewers who wish to understand the structure and logic of the project without having to go through the entire codebase.

The setup guide gives step-by-step instructions to run the project locally or deploy it to cloud platforms like Render, Vercel, or Netlify. It begins with cloning the GitHub repository, installing dependencies using npm install, and running the backend server with npm start. Developers are also guided on how to configure environment variables in a .env file and connect the frontend to the backend API. Additionally, it includes details on testing the /api/contact endpoint, troubleshooting CORS issues, and verifying successful data submission. This ensures that anyone can set up and test the project easily, promoting reusability and collaboration.

6. Final Submission

Repository link :
https://github.com/Thangasaranya/project