



**College code:9605**

**College Name: CAPE INSTITUTE OF TECHNOLOGY**

**Department: B.E, CSE 3<sup>RD</sup> YEAR**

**Student NM-ID: 8946199D7884FEB9DD94D582DA0B51CE**

**Reg No :960523104067**

**Date: 07-10-2025**

**TECHNOLOGY PROJECT -IBM FE**

**Node JS Backend For Contact Form**

**SUBMITTED BY**

**Name: I.Rasithabanu**

**Mobile No: 9884028574**

# IBM-FE NODE JS BACKEND FOR CONTACT FORM

## PHASE-5: PROJECT DEMONSTRATION AND DOCUMENTATION

### 1.FINAL DEMO WALKTHROUGH

#### FINAL DEMO WALKTHROUGH

 Project Overview:

Goal: Create a contact form where users fill details →  
backend saves/sends mail.

 Folder Setup:

contact-form/

├ server.js

├ package.json

└ .env

⚙ Step 1: Initialize Node

```
npm init -y
```

```
npm install express nodemailer dotenv body-parser cors
```

🧠 Step 2: Create .env

```
EMAIL_USER=yourgmail@gmail.com
```

```
EMAIL_PASS=your_app_password
```

💻 Step 3: server.js

```
import express from "express";
import nodemailer from "nodemailer";
import bodyParser from "body-parser";
import cors from "cors";
import dotenv from "dotenv";

dotenv.config();
const app = express();
app.use(cors());
app.use(bodyParser.json());

app.post("/contact", async (req, res) => {
  const { name, email, message } = req.body;

  const transporter = nodemailer.createTransport({
    service: "gmail",
    auth: {
      user: process.env.EMAIL_USER,
      pass: process.env.EMAIL_PASS,
```

```
},
```

```
});
```

```
const mailOptions = {
```

```
  from: email,
```

```
  to: process.env.EMAIL_USER,
```

```
  subject: `Contact Form: ${name}`,
```

```
  text: `From: ${name} (${email})\n\nMessage: ${message}`,
```

```
};
```

```
try {
```

```
  await transporter.sendMail(mailOptions);
```

```
  res.status(200).json({ success: true, message: "Mail sent  
successfully!" });
```

```
  } catch (error) {
```

```
    res.status(500).json({ success: false, error: "Failed to send  
mail" });
```

```
  }
```

```
});
```

```
app.listen(5000, () => console.log("Server running on port 5000"));
```

#### Step 4: Frontend Call (example)

```
fetch("http://localhost:5000/contact", {  
  method: "POST",  
  headers: { "Content-Type": "application/json" },  
  body: JSON.stringify({  
    name: "Lavanya",  
    email: "lavanya@gmail.com",  
    message: "Hello from contact form!",  
  }),  
});
```

#### Step 5: Demo Walkthrough

1. Start backend → node server.js

2. Open frontend form → fill & submit
3. Check console → “Mail sent successfully”
4. Open your Gmail inbox → verify message arrived

## **2.PROJECT REPORT**

 Project Report – Node.js Contact Form

 Title: Contact Form using Node.js Backend

 Objective:

To collect user messages from a website and send them to admin email securely.

 Tools Used:

Node.js

Express.js

Nodemailer

dotenv

Body-parser / CORS

**4** Modules:

Frontend: HTML form (name, email, message)

Backend: API endpoint /contact

Mail Service: Sends email using Nodemailer



## 5 Workflow:

User fills form → data sent to backend → backend sends mail  
→ admin receives message.

## 6 Advantages:

Real-time message delivery

Easy to integrate

Secure with environment variables

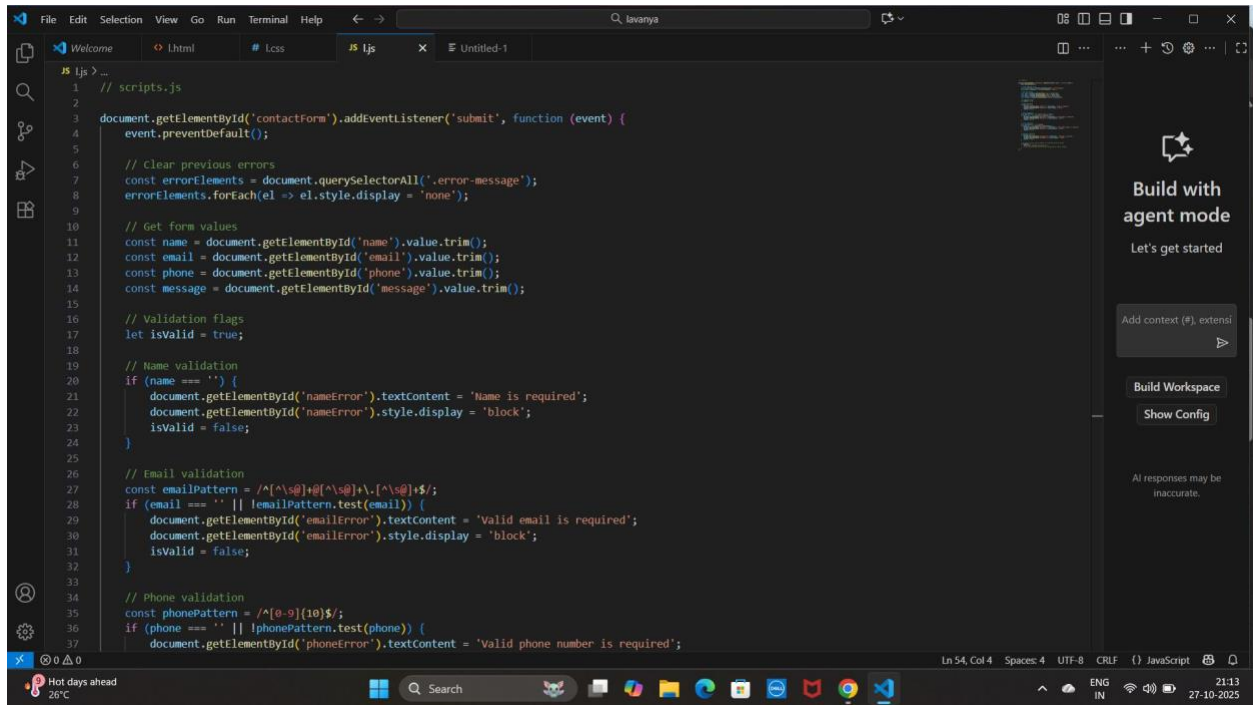
## 7 Result:

Successfully sends form data from website to admin email.

## 8 Conclusion:

Node.js efficiently handles form submissions and email notifications.

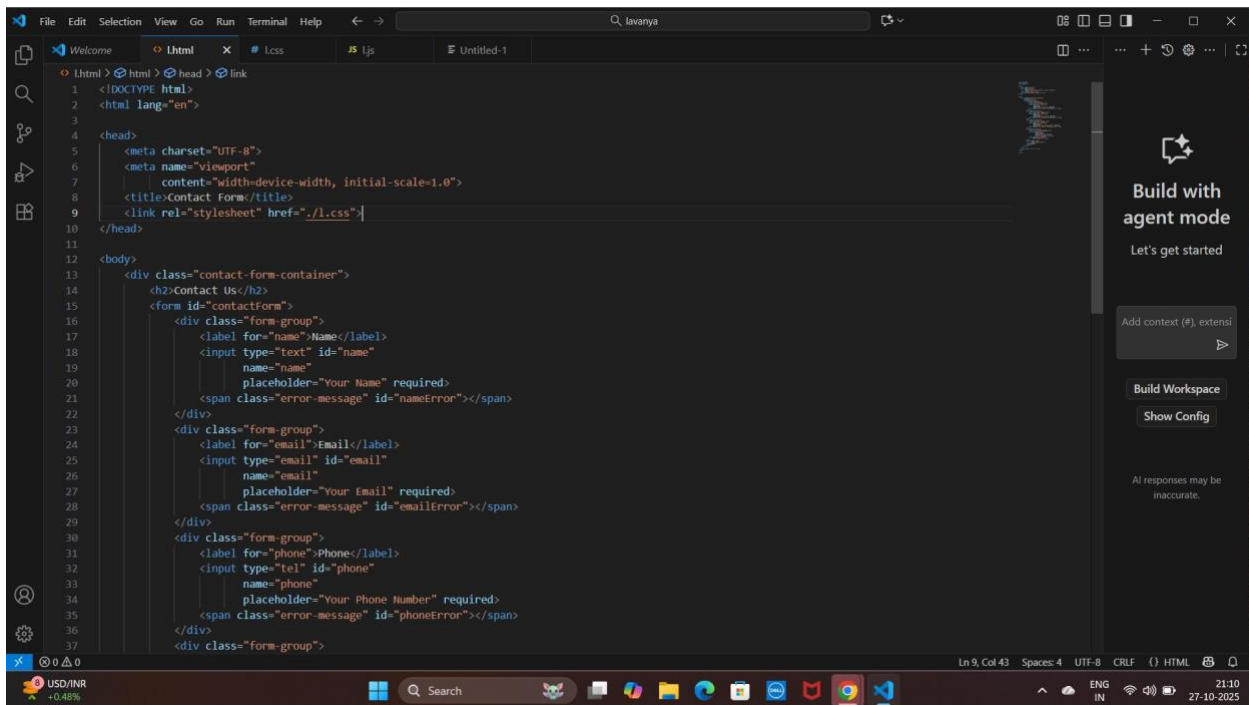
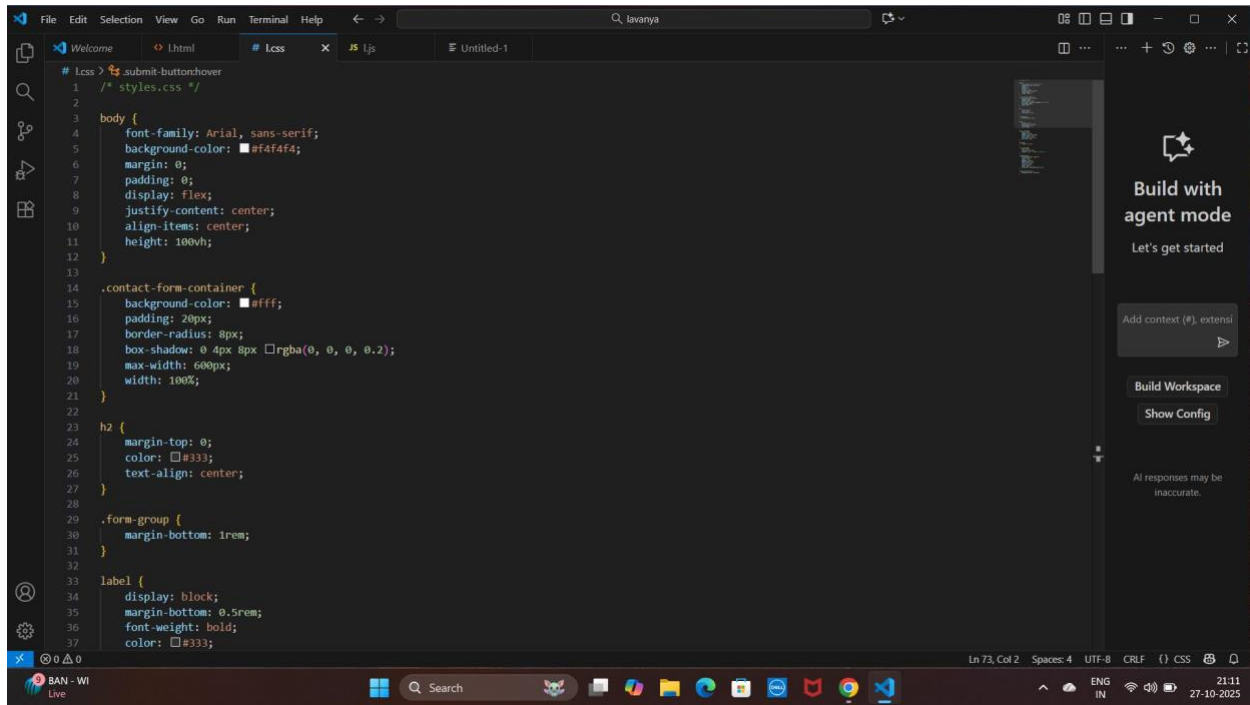
### 3.SCRENSHOTS/API DOCUMENTATION

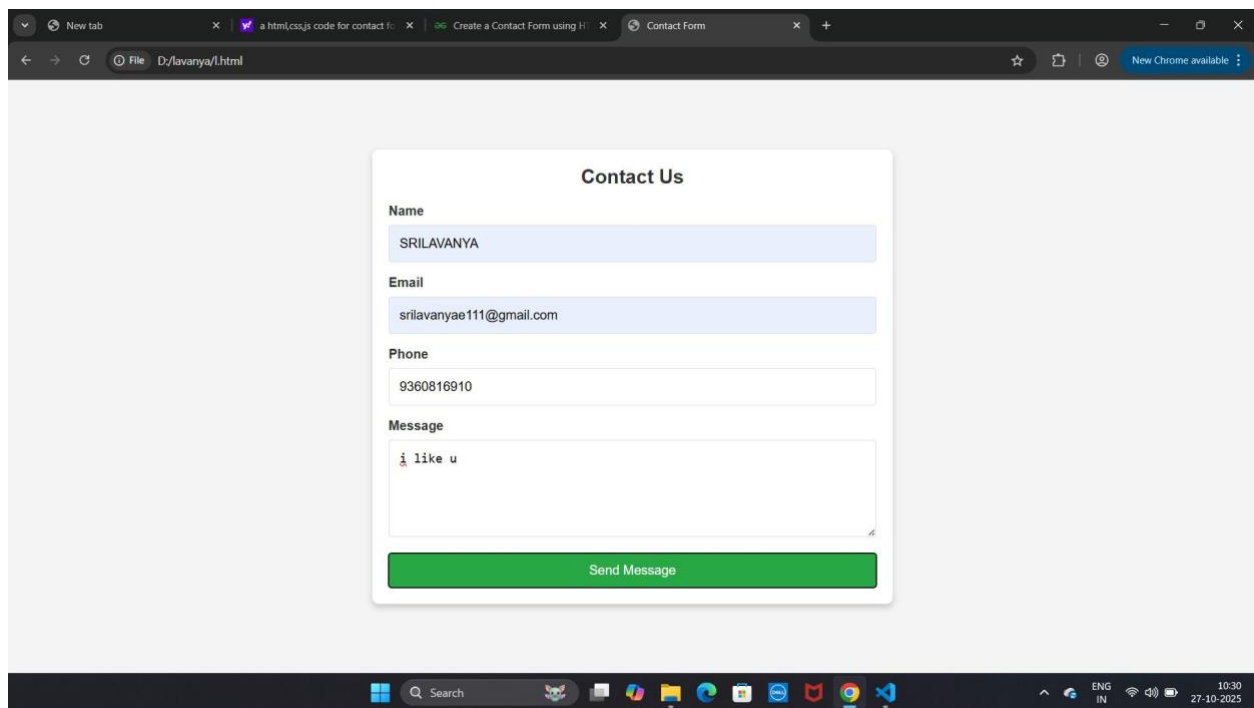
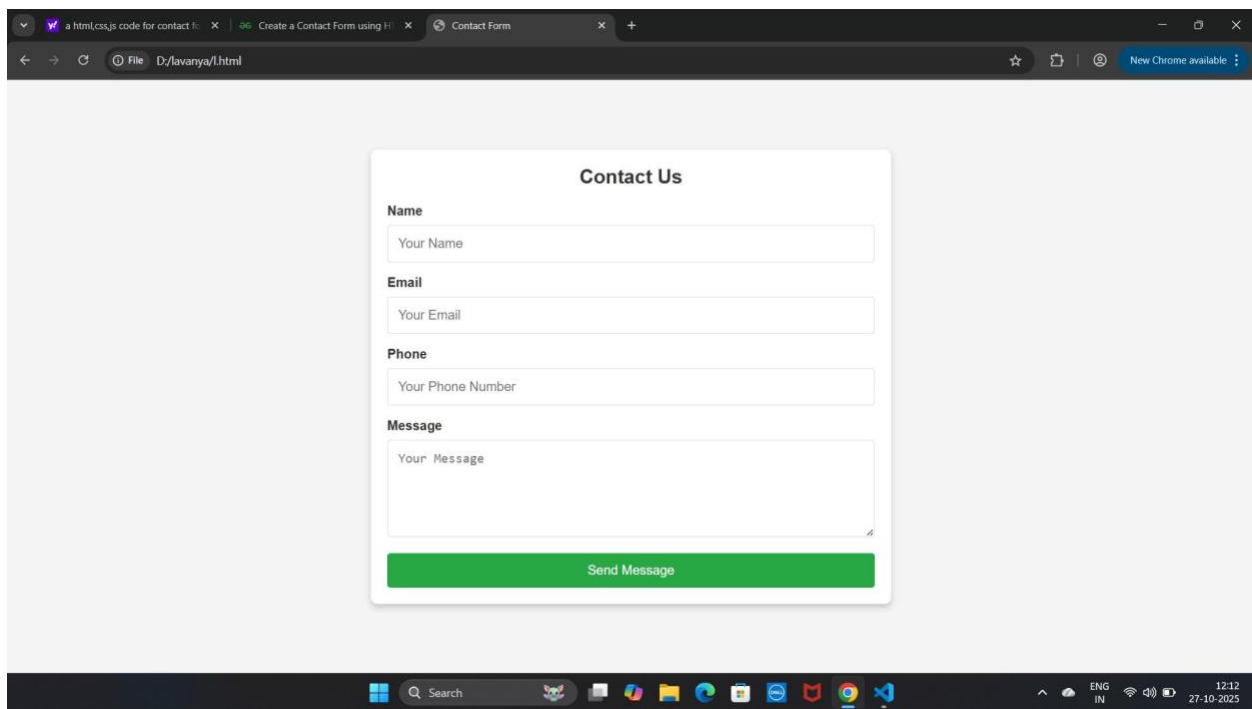


The screenshot displays the Visual Studio Code (VS Code) editor interface. The main editor window shows a JavaScript file named `scripts.js` with the following code:

```
1 // scripts.js
2
3 document.getElementById('contactForm').addEventListener('submit', function (event) {
4     event.preventDefault();
5
6     // Clear previous errors
7     const errorElements = document.querySelectorAll('.error-message');
8     errorElements.forEach(el => el.style.display = 'none');
9
10    // Get form values
11    const name = document.getElementById('name').value.trim();
12    const email = document.getElementById('email').value.trim();
13    const phone = document.getElementById('phone').value.trim();
14    const message = document.getElementById('message').value.trim();
15
16    // Validation flags
17    let isValid = true;
18
19    // Name validation
20    if (name === '') {
21        document.getElementById('nameError').textContent = 'Name is required';
22        document.getElementById('nameError').style.display = 'block';
23        isValid = false;
24    }
25
26    // Email validation
27    const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
28    if (email === '' || !emailPattern.test(email)) {
29        document.getElementById('emailError').textContent = 'Valid email is required';
30        document.getElementById('emailError').style.display = 'block';
31        isValid = false;
32    }
33
34    // Phone validation
35    const phonePattern = /^[0-9]{10}$/;
36    if (phone === '' || !phonePattern.test(phone)) {
37        document.getElementById('phoneError').textContent = 'Valid phone number is required';
```

On the right side of the editor, there is an AI assistant sidebar. It features a search icon, the text "Build with agent mode", "Let's get started", and buttons for "Add context (#), extensi...", "Build Workspace", and "Show Config". At the bottom of the sidebar, it states "AI responses may be inaccurate." The status bar at the bottom of the VS Code window shows "Ln 54, Col 4", "Spaces: 4", "UTF-8", "CRLF", "JavaScript", and a temperature indicator "Hot days ahead 26°C". The system tray at the very bottom shows the date "27-10-2025" and the time "21:13".





## 4.CHALLENGES & SOLUTIONS



### Challenges & Solutions –Node.js Contact Form

**1** Challenge: Email not sending due to Gmail security.

Solution: Used App Password and .env file for safe authentication.

**2** Challenge: CORS error when connecting frontend & backend.

Solution: Added cors() middleware in Express.

**3** Challenge: Handling invalid or empty form data.

Solution: Added input validation before sending mail.

**4** Challenge: Exposing credentials in code.

Solution: Stored sensitive data in .env file.

**5** Challenge: Slow response when sending mail.

Solution: Used async/await to handle mail sending smoothly.

**6** Challenge: Server crash on errors.

Solution: Used try-catch and proper error responses.

## 5.Github README and setup guide

 Node.js Contact Form Backend

 Description

A simple Node.js + Express backend that receives contact form data and sends it as an email using Nodemailer.

 Tech Stack

Node.js

Express.js

Nodemailer

Dotenv

CORS



Setup Guide

**1** Clone the repo

Git clone <https://github.com/yourusername/contact-form-backend.git>

Cd contact-form-backend

**2** Install dependencies

Npm install

**3** Create .env file

EMAIL\_USER=yourgmail@gmail.com


EMAIL\_PASS=your\_app\_password

**4** Run server

Node server.js

Server runs at:

 <http://localhost:5000>

 API Endpoint

POST /contact

{



```
“name”: “Lavanya”,  
“email”: lavanya@gmail.com,  
“message”: “Hello!”  
}
```

Response:

 Mail sent successfully!

 Folder Structure

Contact-form-backend/

├ server.js


├ package.json

└ .env

 Author

Sri Lavanya — Node.js Contact ForExpress.js

## 6.FINAL SUBMISSION (REPOSITORY +DEPLOYED LINK)

 Final Submission – Node.js Contact Form Backend

  GitHub Repository

Repository Name:

nodejs-contact-form-backend

Contents:

 nodejs-contact-form-backend/

├ server.js

├ package.json

├ .env.example

├ README.md

└ .gitignore

 Must include:

Clean, commented code

Proper README

.env.example file (not real credentials)

Screenshot of mail success (optional)

  Deployment Link

Deploy your backend using Render (free & easy):

Steps:

 Go to <https://render.com>

- 2 Click New → Web Service
- 3 Connect your GitHub repo
- 4 Set:

Build Command → npm install

Start Command → node server.js

- 5 Add environment variables:

EMAIL\_USER

EMAIL\_PASS

- 6 Click Deploy

After deploy → you'll get a link like:

 <https://node-contact-form.onrender.com>

 Final Submission Format

ItemLink

GitHub Repo <https://github.com/yourusername/nodejs-contact-form-backend>

Deployed Link <https://node-contact-form.onrender.com>

## **GITHUB LINK:**

<https://Github.com/rasithabanu111/Node-js-backend-for-contact-form>

## **NETLIFY LINK:**

<https://app.netlify.com/teams/rasithabanu111/projects>

