



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Report File FULL STACK

Student Name: Souradeep Banerjee

UID: 23BAI70654

Branch: BE-AIT-CSE

Section/Group: 23AIT-KRG-G2

Semester: 5th

Date of Performance: 25th Aug, 2025

Subject Name: Full Stack

Subject Code: 23CSP-339

Project Report: Blogging Website

1. AIM

The aim of this project is to build a simple **Blogging Website** where users can:

- Create and read blog posts.
- Edit and delete their own posts.
- Store blog posts in a database.

The project helps beginners learn **frontend (HTML, CSS, JS)** and **backend (Node.js with Express)** basics, along with database usage.

2. TECH STACK

Frontend

- **HTML5** → Page structure.
- **CSS3** → Styling and layout.
- **Vanilla JavaScript** → Interactivity and fetching data from backend.

Backend

- **Node.js** → JavaScript runtime for backend.
- **Express.js** → Framework to handle routes and APIs.

Database

- **JSON file / MongoDB**
 - Use a local **JSON file** to store posts.

Tools

- **VS Code** → Code editor.
 - **Browser** → To run the application.
-

3. THEORY / BACKGROUND

The Blogging site follows the **client-server model**:

- **Client (HTML, CSS, JS)** → Displays posts and has forms for creating/editing blogs.
- **Server (Node + Express)** → Handles requests (create, read, update, delete posts).
- **Database (JSON/MongoDB)** → Stores blog data.

We will design REST APIs:

- GET /posts → Fetch all posts.
 - POST /posts → Create a new post.
 - PUT /posts/:id → Edit a post.
 - DELETE /posts/:id → Delete a post.
-

4. PROCEDURE / IMPLEMENTATION

Step 1: Setup Environment

1. Install **Node.js**.
 2. Create a new folder blog-site.
 3. Run:
 4. `npm init -y`
 5. `npm install express body-parser cors`
-

Step 2: Run the Project

1. Start the backend:
 2. `node server.js`
 3. Open index.html in your browser.
 4. Create, view, and delete blog posts.
-

5. CONCLUSION

This project shows how to build a **simple full-stack Blogging Site** using **JavaScript, HTML, CSS, and Node.js (Express)**.

It introduces:

- **Frontend (UI)** → HTML, CSS, JS.
- **Backend (APIs)** → Node.js + Express.
- **Data Storage** → In-memory or JSON file.