9530

St.MOTHERTHERESAENGINEERINGCOLLEGE

COMPUTER SCIENCE ENGINEERING

NM-ID 9B36DDD6C314B13BE454072B8B669404

REG NO :953023104002

DATE:06-10-2025

Completed the project named as

Phase 5

# FRONT END TECHNOLOGY
# BLOGSITE COMMENT SECTION


SUBMITTED BY:

M.ABINANTHAN

9360660493

# 🧩 PROJECT: BLOG SITE WITH COMMENT SECTION

**Deadline: Week 10**
**Type: Full Stack (or Frontend + Backend) Web Application**
**Goal: Create a platform where users can view, post, and comment on blogs.**

## 🧠 1. Final Demo Walkthrough

### 📹 What to Show in Your Demo Video

**Duration: 3–5 minutes**
**Tool: Loom / OBS Studio / ScreenPal / built-in Windows recorder (Win + Alt + R)**

**Structure of the demo:**

1. **Introduction – "Hi, I'm Balakumar. This is my Blog Site project."**
2. **Homepage – Show list of blog posts.**
3. **View Post – Click on one post to show full content.**
4. **Comment Section – Type and post a comment, then show it appears instantly.**
5. **(Optional) Post Creation – If admin can add posts, demonstrate that.**
6. **Responsive Design – Resize window to show it works on mobile view.**
7. **(Optional) Login / Register – If you added user system, show it.**
8. **Conclusion – Summarize features and mention your GitHub and live site links.**

💡 **Tip: Keep the screen clean, zoom in on key actions, and speak clearly.**

## 🟦 2. Project Report (Detailed Version)

**You'll need to create this in Word (docx) or PDF format.**
**Here's the complete structure with content guidance:**

### 📄 1. Title Page

**Include:**

- **Project Title: "Blog Site with Comment Section"**
- **Course Name, Semester, College Name**
- **Student Name, Register Number**
- **Date of Submission**

### 📝 2. Abstract

**Short paragraph (5–6 lines) describing your project:**

**This project is a blog website that allows users to read, write, and comment on blog posts. It demonstrates the use of web technologies such as HTML, CSS, JavaScript, Node.js, and MongoDB. The system provides an interactive and user-friendly interface with comment functionality to encourage user engagement.**

## 💡 3. Introduction

**Explain what a blog site is, why it's useful, and what problems it solves.**
**Example:**

**Blogging is one of the most popular ways of sharing information and opinions online. This project focuses on developing a blog platform where users can browse articles and participate in discussions through comments. The goal is to create a simple, responsive, and interactive blogging system.**

## 🎯 4. Objectives

- **To design a simple and responsive blog website.**
- **To enable adding and displaying blog posts dynamically.**
- **To allow users to post comments under each blog.**
- **To store posts and comments in a database.**
- **To deploy the application on a live server.**

## ⚙️ 5. Technologies Used

| Layer | Technology | Purpose |
|---|---|---|
| Frontend | HTML, CSS, JavaScript / React .js | UI Design |
| Backend | Node.js, Express.js | API & Server |
| Database | MongoDB (Mongoose) | Store posts & comments |
| Hosting | Vercel / Render / Netlify | Deployment |
| Tools | Git, GitHub | Version control |

## 🧩 6. System Architecture / Design

**Add diagrams like:**

- Architecture Flow:
  Browser → Express Server → MongoDB
- Database Schema:
  Collections:
- Posts {
- id, title, content, author, date
- }
-
- Comments {
- id, postId, username, comment, date
- }

You can draw this using draw.io or PowerPoint.

## 💻 7. Implementation Details

Explain step by step:

- Frontend:
  - Used React components for pages like Home, PostDetail, AddComment.
  - Fetch posts and comments from backend using Axios or Fetch API.
- Backend:
  - Built REST APIs using Express:
    - GET /posts
    - GET /posts/:id
    - POST /posts
    - POST /posts/:id/comments
- Database:
  - MongoDB collections for posts and comments.
- Deployment:
  - Backend → Render
  - Frontend → Vercel

## 🧱 8. Modules Description

| Module | Description |
| --- | --- |
| Home Module | Displays all blog posts |
| Post Module | Shows individual post content |
| Comment Module | Allows users to comment on posts |
| Admin Module (Optional) | Add/edit/delete posts |
| Authentication Module (Optional) | Login/Signup features |

## 🚧 9. Challenges & Solutions

| Challenge | Solution |
| --- | --- |
| Database connection timeout | Used async/await with error handling in MongoDB connection |
| Comments not updating instantly | Used React state update and re-fetch after comment submission |
| CORS error during API calls | Enabled CORS in Express using middleware |
| Deployment failed on Render | Fixed environment variables and added build command |

## ✅ 10. Conclusion

**The Blog Site project successfully provides a platform for reading and commenting on posts. It helped me understand full-stack development concepts like REST APIs, database management, and deployment. Future improvements include user authentication, image uploads, and profile management.**

## 📚 11. References


- https://react.dev

- https://expressjs.com
- https://mongoosejs.com
- https://www.mongodb.com
- YouTube Tutorials / Documentation links used

## 📸 3. Screenshots & API Documentation

### Add screenshots of:

1. **Homepage**
2. **Single Post page**
3. **Comment section**
4. **Post creation (if any)**
5. **Database (optional screenshot of MongoDB collection)**
6. **Deployment success page**

### Example API Docs section:

| Method | Endpoint | Description |
|---|---|---|
| GET | /api/posts | Get all blog posts |
| GET | /api/posts/:id | Get a single post by ID |
| POST | /api/posts | Create a new post |
| POST | /api/posts/:id/comments | Add comment to a post |

## ⚙️ 4. GitHub README & Setup Guide (Full)

**Your README.md file should include:**

1. **Project Title & Description**
2. **Features**
3. **Tech Stack**
4. **Installation & Setup Guide**
5. **Folder Structure**
6. **API Endpoints**
7. **Screenshots (add using ![alt text](image_link))**
8. **Live Demo Link**
9. **Author info**

**link**

## 🏁 5. Final Submission Checklist

✅ **GitHub Repo (Public)**
✅ **Live Deployed Link (Render/Vercel)**
✅ **Project Report (PDF or DOCX)**
✅ **Screenshots & API Docs**
✅ **Demo Video (Optional but recommended)**