



COLLEGE CODE:9605

COLLEGE NAME: CAPE INSTITUTE OF TECHNOLOGY

DEPARTMENT: BE/CSE 3rd YEAR

STUDENT NM-ID: FF33D7996AB55C9BBC2450AB4E364978

Roll no: 960523104020

Date:06/10/2025

COMPLETED A PROJECT NAME AS PHASE -5

TECHNOLOGY PROJECT NAME IBM-FE-BLOG SITE WITH COMMENT SECTION

SUBMITTED BY

NAME: ANGEL M

PHONE NO: 9942209473

PHASE-5

Project Demonstration & Documentation

1. Final Demo Walkthrough

> A brief written walkthrough of your project, describing how it works.

Overview:

A responsive blog platform where users can:

View blog posts

Read comments on each post

Submit new comments

Admins can create, edit, or delete blog posts

Walkthrough Steps:

1. Homepage: Displays a list of blog posts with titles and summaries.

2. Post Detail Page: Shows the full blog content and associated comments.

3. Comment Section: Users can submit a comment (name, content).

4. Admin Panel: Accessible via login (optional), allows post management.



Tech Stack:

Frontend: HTML, CSS,

JavaScript (React or Vanilla)

Backend: Node.js / Express / Django / Flask

Database: MongoDB / MySQL / PostgreSQL

Hosting: Vercel / Netlify (Frontend), Render / Heroku (Backend)

Key Features:

CRUD for Blog Posts

Nested Comments or Basic Comment System

Authentication (optional)

Responsive UI

3. Screenshots / API Documentation

> \times You'll need to upload your screenshots manually or generate them using your live app.

Typical Screenshots:

1. Home Page with blog listings

2. Single blog post view with comments

3. Add comment form

4. Admin panel (if any)

5. Mobile view (for responsive layout)

API Endpoints Example:

Method Endpoint Description

GET /api/posts Get all posts GET /api/posts/:id Get single post

POST /api/posts Create a post

PUT /api/posts/:id Update a post

DELETE /api/posts/:id Delete a post

POST /api/comments Add comment to post

4. Challenges & Solutions

Challenge Solution

Setting up dynamic routes
Used React Router /
Express router
Comment nesting

implementation Used

recursive data structure or flat threading

Form validation Implemented client-side and server-side validation

Hosting full stack app Used Vercel + Render for frontend/backend deployment

♦ 5. GitHub README & Setup Guide

Your README.md should include:

Project Description

Tech Stack

How to Run Locally:

git clone
https://github.com/yourusern
ame/blog-project.git
cd blog-project
npm install
npm run dev

API Usage (with examples)

Screenshots (Markdown image links)

Live Demo Link

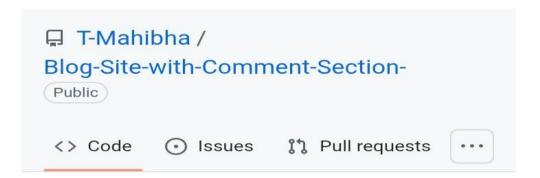
\$\langle\$ 6. Final
Submission

GitHub Repo:

github.com/yourusername/bl og-project (Replace with actual link)

Live Link: blogproject.vercel.app (Replace with actual deployed link)

♦ LOGO PAGE +SCREENSHOT OUTPUT





This repository is empty.

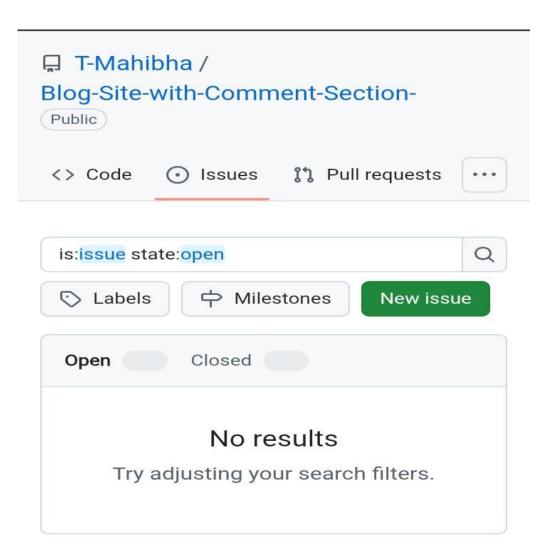
Care to check out the <u>GitHub</u>
Channel on YouTube while you wait?

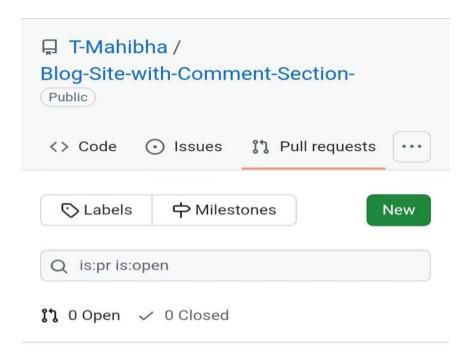
Terms Privacy Security Status Community

Docs Contact Manage cookies

Do not share my personal information







17

Welcome to pull requests!

Pull requests help you collaborate on code with other people. As pull requests are created, they'll appear here in a searchable and filterable list. To get started, you should create a pull request.

GITHUP LINK:

https://github.com/agnesangel1301-

lang/Angel-.git