GLA University

TOPIC: MINI PROJECT SYNOPSIS ON BLOGGER WEBSITE APP

Submitted By : *Ashwani Suryansh, Ashlesh Khobragade ,Arpit singh, Chyawan Chhonkar*

Submitted to: *Mr. Akash Kumar Choudhary*

Name: *Ashwani Suryansh*

Id: *2315000505*

Name*: Ashlesh Khobragade*

Id: *2315000484*

Name: *Arpit Singh*

Id: *2315000430*

Name: *Chyawan Chhonkar*

Id: *2315000664*

Faculty Name*: Mr. Akash Kumar Choudhary*

Technical Trainer

DECLARATION

I declare that the Mini Project titled ***"Dynamic Blogging Website with User Interaction and Content Management"*** *has been done by me as a part of my course work. I have completed this project under the kind guidance of* ***Mr Akash kumar Chaudhary, Technical Trainer****.*

I confirm that this project is my own original work. The ideas, data, and information used in this report are true and correct as per my knowledge. Whenever I have used information from books, websites, or any other source, I have mentioned it clearly in the references section.

INDEX

This Index is a reference for synopsis only

S.NO	*Topic*
1	*Introduction/ abstract/title*
2	*System Requirements*
3	*Hardware Requirements*
4	*Front End and Back End*
5	*Idea*
6	*Objective*
7	*Module Description*
8	*Availability*
9	*DFD 0level 1 level and 2 level*
10	*BIBLIOGRAPHY*
11	*References*

INTRODUCTION **: –**

Blogging is one of the easiest ways to share thoughts, knowledge, and news with other people. Many people and companies use blogs to express ideas and connect with readers.

This project, ***"Dynamic Blogging Website with User Interaction and Content Management,"*** *is about making a simple website where anyone can write posts, edit them, and share them. People who visit the website can read posts, like them, and write comments.*

The website is designed to work well on computer, tablet, and mobile screens. It also has a small dashboard where posts and users can be managed. This project helps us learn how websites are made, how data is stored in a database, and how to keep everything safe and easy to use.

About the Project:

*This project is about creating a **dynamic blogging website** where people can write and share posts online. The website allows authors to **create, edit, update, and delete** their posts easily. Readers can interact with the content by **reading, liking, and commenting** on the posts.*

*The website is made to be **responsive**, so it works on computers, tablets, and mobile phones. It also has a simple **admin dashboard** to manage users and posts.*

*The main goal of the project is to make blogging easy for everyone and to give a good experience to both writers and readers. It also helps us learn important web development skills like **HTML, CSS, JavaScript, React (for front-end), Node.js (for back-end), and database management***

Primary Reason to Choose This Project:

*The main reason for choosing this project is to gain **practical knowledge of how a real website is built and works**. Blogging is a very common and powerful way to share thoughts, ideas, and information with people. By creating our own blogging website, we get to learn how writers add posts, how readers view and comment on them, and how everything is managed in the background.*

*This project gives us a chance to practice almost every part of web development. We work on **front-end design** to make the website look good and easy to use. We also work on the **back-end** to handle data and user requests. We use a **database** to store posts, comments, and user details safely. This way, we get complete experience of developing a real-world project.*

*Another reason to choose this project is that it is **useful and practical**. A blogging website can be used by students, teachers, professionals, or anyone who wants to share knowledge online. This makes the project meaningful, not just for learning but also for real use.*

*Finally, this project helps us improve skills like **problem-solving, teamwork, and creativity**. We also learn about security, user experience, and responsive design, which are very important for any website today.*

The Main Objective of the Project

*The main objective of this project is to **create a simple, user-friendly, and responsive blogging website** where people can easily write, edit, and share posts online.*

We want to make a platform where:

- ***Authors** can create, update, and delete their blog posts.*
- ***Readers** can read posts, like them, and add comments.*
- ***Admins** can manage users and posts from a simple dashboard.*

The project also aims to:

- *Learn and practice **front-end technologies** (HTML, CSS, JavaScript, React) to make the website look good and easy to use.*
- *Work with **back-end development** (Node.js/Express or Django) to handle data and manage requests.*
- *Use a **database** (MySQL or MongoDB) to store posts, comments, and user information securely.*
- *Make the website **responsive**, so it works well on computers, tablets, and phones.*
- *Ensure a **safe and smooth user experience** by focusing on security and usability.*

*Overall, the main objective is to build a **complete blogging platform** that is simple, fast, and practical — and to gain **real-world project experience** by working on every part of a web application*

Scope Of the Project

*The project aims to create a **simple and responsive blogging website** where authors can write, edit, and delete posts, and readers can like and comment on them. It covers **front-end, back-end, and database development**, giving complete web development experience.*

In the future, the project can be extended by adding features like search filters, media uploads, user profiles, notifications, and analytics to make it more useful and powerful.

Working Methodology Of the Project;

The project follows a step-by-step approach to build the blogging website:

1. ***Planning & Requirement Gathering** – Understanding what features are needed, like post creation, comments, likes, and user roles.*
2. ***Design** – Creating a simple layout (wireframe) and database design for storing posts, users, and comments.*
3. ***Development** –*
   - ***Front-end:** Building web pages using HTML, CSS, JavaScript, and React.*
   - ***Back-end:** Creating REST APIs using Node.js/Express (or Django).*
   - ***Database:** Storing data in MySQL or MongoDB.*
4. ***Testing** – Checking each feature to make sure it works correctly on all devices and browsers.*
5. ***Deployment** – Hosting the website on a server so it can be accessed by users online.*

This step-by-step process ensures the website is easy to use, secure, and works well on all devices.

Details About the Hardware and the Software

***System Requirements*: -**

Supported Operating System

- *Windows 10 / 11*
- *Linux (Ubuntu or similar)*
- *macOS (optional)*

Software Required:-

- ***Front-end:*** *HTML5, CSS3, JavaScript, React.js*
- ***Back-end:*** *Node.js with Express (or Django for Python)*
- ***Database:*** *MySQL or MongoDB*
- ***Version Control:*** *Git, GitHub*
- ***IDE / Code Editor:*** *VS Code, IntelliJ IDEA, Android Studio, or Eclipse*
- ***Browser:*** *Google Chrome / Mozilla Firefox (for testing)*

***Hardware* Requirements: -**

- ***Processor:*** *Intel i3 / AMD Ryzen 3 or higher*
- ***RAM:*** *Minimum 8 GB (Recommended 16 GB for smoother development)*
- ***Storage:*** *At least 500 GB (preferably SSD for faster performance)*
- ***Internet:*** *Stable connection for testing and deployment*

-

For Android Studio and Intelli j: -

- ***Processor:*** *Minimum Intel i5 / AMD Ryzen 5*
- ***RAM:*** *8 GB (16 GB recommended)*
- ***Disk Space:*** *4 GB for IDE + 2 GB for Android SDK*
- ***Graphics:*** *1080p display with GPU support for Emulator*

For Eclipse: -

- ***Processor:*** *Intel i3 / Ryzen 3 or higher*
- ***RAM:*** *Minimum 4 GB (8 GB recommended)*
- ***Disk Space:*** *1 GB for Eclipse + additional space for projects*

Listing Out testing technology

Frontend and Backend: -

Frontend:-

- ***Cross-Browser Testing:*** *Checking website on Chrome, Firefox, Edge*
- ***Responsiveness Testing:*** *Ensuring proper display on mobile, tablet, and desktop*
- ***Manual Testing:*** *Checking navigation, buttons, and forms*

Backend:-

- ***Unit Testing:*** *Testing individual API functions using Jest or Mocha*
- ***Integration Testing:*** *Checking API endpoints with Postman*
- ***Database Testing:*** *Ensuring correct data storage and retrieval*

Module Description:

This project is divided into different modules to make development simple and organized.

1. User Module

- *Allows users to sign up, log in, and log out.*
- *Readers can view blog posts, like them, and add comments.*

2. Author Module

- *Authors can create, edit, and delete their blog posts.*
- *Authors can also view comments on their posts and respond if needed.*

3. Admin Module

- *Admin can manage all users (authors/readers) and their posts.*
- *Admin can remove inappropriate content or block users if necessary.*

4. Post Management Module

- *Handles storage and display of blog posts from the database.*
- *Supports CRUD operations (Create, Read, Update, Delete).*

5. Comment & Like Module

- *Allows readers to like posts and write comments.*
- *Displays number of likes and comments for each post.*

6. Dashboard Module

- *Provides an easy-to-use interface for authors and admin to manage posts and users.*

7. Security Module

- *Handles password encryption, authentication, and safe access to data.*

Data Flow Daigrams

DFD:-

*A **Data Flow Diagram (DFD)** shows how data moves through the system. It focuses on how information flows between different parts of the project, where it is stored, and what processes act on it.*

0 Level DFD: -

*Level 0 DFD shows the **whole system as a single process** with its inputs and outputs.*

- ***Actors:** Users (Readers), Authors, Admin*
- ***Main Process:** Blogging Website*
- ***Inputs:** Login details, blog posts, comments, likes*
- ***Outputs:** Displayed posts, success messages, user feedback*

(This level just shows that users interact with the system and get results.)

1 level DFD: -

*Level 1 DFD breaks the system into **main processes**.*

- ***Process 1:** User Management (Login, Signup)*
- ***Process 2:** Post Management (Create, Edit, Delete, View Posts)*
- ***Process 3:** Interaction (Likes & Comments)*
- ***Process 4:** Admin Controls (Manage Users & Posts)*
- ***Datastores:***
  - *D1: User Database*
  - *D2: Post Database*
  - *D3: Comment/Like Database*

(This level shows how data moves between users, processes, and databases.)

2 level DFD:-

*Level 2 DFD shows **detailed data flow** inside each process.*

- ***User Management:***
  - *User enters login/signup data → System checks User Database → Allows or denies access.*
- ***Post Management:***
  - *Author enters post → Data stored in Post Database → Displayed to readers.*
  - *Author edits/deletes post → Database updated → Frontend refreshed.*
- ***Interaction:***
  - *Reader likes or comments → Data stored in Comment/Like Database → Post interaction count updated.*
- ***Admin Controls:***
  - *Admin fetches data from all databases → Can remove posts/users if needed*

Refrences:

- ***MDN Web Docs*** – *https://developer.mozilla.org*
- ***React.js Official Documentation*** – *https://react.dev*
- ***Node.js Official Documentation*** – *https://nodejs.org*
- ***Django Project Documentation*** – *https://www.djangoproject.com*
- ***TutorialsPoint*** – *"Building Web Applications"*
- ***W3Schools*** – *https://www.w3schools.com*
- ***GeeksforGeeks*** – *https://www.geeksforgeeks.org*


GLA UNIVERSITY
ESTD. 2010