COLLEGE CODE:8203

COLLEGE NAME:A.V.C College of Engineering

DEPARTMENT: Computer Science And Engineering

STUDENT NM-ID:2774A1DBA16809C8C764F7BFA755494C

ROLL NO:820323104091

DATE:08-09-2025

Completed the project named as

Phase1TECHNOLOGY

PROJECT NAME :Blogging Platform using

Node.js, Express, MongoDB, and JWT.

SUBMITTED BY,

NAME: G.Santhosh

MOBILE NO:6369502254

Phase 1 — Problem Understanding & Requirements :

1. <u>Problem Statement:</u>

In today's digital world, blogging has become one of the most common mediums for sharing knowledge, opinions, and experiences. Many individuals and organizations rely on blogging platforms to publish articles, tutorials, news updates, or personal journals. However, most existing bloggingplatforms suffer from the following limitations:

1. **Complexity** – Many platforms provide too many advanced features, which overwhelm beginners who only want a simple interface to write and publish blogs.
2. **Lack of customization** – Users often cannot control how their blogs look or manage content freely due to platform restrictions.
3. **Security issues** – Weak authentication and poorly designed systems make user accounts vulnerable to unauthorized access.
4. **Scalability challenges** – Some platforms struggle to handle large numbers of users or blogs efficiently.

This project aims to address these issues by designing and implementing a **simple, secure, and scalable blogging platform**. The system will allow users to **register, login, write blogs, and manage them** with minimal complexity. The architecture will focus on **security** using JWT-based authentication and on **scalability** by adopting a clean backend structure with MongoDB.

2. Users & Stakeholders:

To understand the problem better, it is important to identify the **intended users** and **stakeholders** involved in the system.

- **<u>Users:</u>**
    1. **Authors (Bloggers):**
        - These are registered users who can write, edit, and delete blogs.
        - They form the core content creators of the platform.
        - They also have the ability to comment on other blogs.
    2. **<u>Readers:</u>**
        - These users consume the content by reading blogs.
        - They may or may not be registered, but if registered, they can comment on blogs.
- **<u>Stakeholders:</u>**
    1. **Project Team:** Responsible for planning, designing, developing, and testing the platform.
    2. **Mentors/Guides:** Provide technical and domain-related guidance during the project.
    3. **Administrators:** Oversee user activities, ensure rules are followed, and maintain the system's smooth functioning.

By identifying these groups, we ensure that the requirements are collected with the end-users mind and the platform aligns with their needs.

3. User Stories:

User stories describe the functionality from the **end-user perspective**. They are small, structured statements that clarify what a user needs and why.

1. **User Authentication**
   - As a user, I want to register/login securely so that I can access my account.
2. **Blog Management**
   - As a blogger, I want to create blogs so that I can share my knowledge.
   - As a blogger, I want to edit my existing blogs so that I can improve or update them.
   - As a blogger, I want to delete blogs so that I can remove outdated or unwanted content.
3. **Content Consumption**
   - As a reader, I want to view all blogs so that I can read and gain insights from different authors.
4. **Engagement**
   - As a user, I want to comment on blogs to engage in discussions and express my opinions.

These user stories act as the **foundation for the system's features** and guide development priorities.

4. MVP Features:

The **Minimum Viable Product (MVP)** focuses on providing the essential features required to test and validate the platform. The identified MVP features are:

1. **User Authentication with JWT**
   - Secure login and registration using JSON Web Tokens (JWT).
   - JWT ensures session security without storing sensitive session data on the server.
2. **CRUD Operations on Blogs**
   - Create: Bloggers can write and publish blogs.
   - Read: Readers can view blogs from all authors.
   - Update: Bloggers can edit their own blogs.
   - Delete: Bloggers can delete their own blogs.
3. **View All Blogs**
   - Readers can browse a list of all available blogs, sorted by latest posts.
4. **Basic Comment System**
   - Users can comment on blogs to encourage interaction and discussions.

These features strike a balance between **simplicity** and **functionality**, ensuring that the platform remains usable and valuable even in its initial version.

5. Wireframes & API Endpoints:

5.1 Wireframes

Wireframes represent the **visual layout** of the application's interface. The proposed wireframes are:

1. **Login Page:**
   o Simple form with email/username and password fields.
   o "Login" and "Register" buttons.
2. **Register Page:**
   o Fields for name, email, and password.
   o "Create Account" button.
3. **Blog List Page:**
   o Displays a list of blogs with title, author name, and short description.
   o Options for viewing a blog or creating a new one (if logged in).
4. **Blog Editor Page:**
   o Form with fields for blog title and content.
   o Buttons for "Save", "Update", or "Delete".

5.2 API Endpoint List

The backend will expose a set of RESTful APIs to handle operations. Example endpoints include:

- **Authentication:**
  o `POST /api/auth/register` – Register a new user.
  o `POST /api/auth/login` – Login and receive a JWT token.
- **Blog Operations:**
  o `GET /api/blogs` – Get all blogs.
  o `POST /api/blogs` – Create a new blog (requires authentication).
  o `PUT /api/blogs/:id` – Update an existing blog (requires authentication).
  o `DELETE /api/blogs/:id` – Delete a blog (requires authentication).
- **Comments:**
  o `POST /api/blogs/:id/comments` – Add a comment to a blog.

These endpoints define the **contract between frontend and backend**, ensuring consistent communication.

6. Acceptance Criteria:

Acceptance criteria define **when a feature can be considered complete and successful**. For Phase1, the acceptance criteria are:

1. **Authentication**
   o Users can register and login successfully.
   o JWT tokens are generated and validated for secure access.
2. **Blog Management**
   o Users can create, update, and delete their own blogs.
   o Blogs are stored securely in **MongoDB**.
3. **Content Retrieval**
   o All blogs can be retrieved through the API.
   o The blog list is correctly displayed in the frontend.
4. **Comment System**
   o Users can post comments on blogs.
   o Comments are associated with the respective blog posts.
5. **API Functionality**
   o All listed endpoints return correct responses when tested (status codes, data format, and error handling).

When all acceptance criteria are met, **Phase 1 can be considered complete**.

7. Conclusion:

Phase 1 lays the foundation for the blogging platform by clearly defining the **problem, stakeholders, requirements, MVP features, and acceptance criteria**. The system aims to solve the issues of complexity, customization, and security by providing a lightweight platform with essential features such as authentication, blog management, and commenting.

This phase ensures that development moves forward with a **clear vision**, user-centered requirements, and measurable success criteria. Once implemented, this phase will provide a working prototype of the blogging platform, which can be tested and improved in future phases.