COLLEGE CODE : 9623

COLLEGE NAME : Amrita College Of Engineering And Technology

DEPARTMENT : Computer Science and Engineering

STUDENT NM-ID : BDEE936B6F75EC18C6640B3F81D816C5

ROLL NO : 962323104077

DATE : 12-09-2025

Completed the project named as Phase_02_ TECHNOLOGY

PROJECT NAME : Blog Site with Comment Section

SUBMITTED BY,

NAME : R Prenesh Kumar

MOBILE NO : 96293 10500

Phase 2-Solution Design & Architecture

Tech Stack Selection

1. Frontend

- HTML5 – For structuring the web pages.
- CSS3 – For styling, layout, and responsive design.
- JavaScript – For interactivity and dynamic updates on the client side.

2. Backend

- PHP (or Python with Flask/Django / Node.js with Express) – For handling server-side logic, request processing, and connecting the frontend with the database.
- RESTful API – To ensure a clear communication layer between the frontend and backend.

3. Database

- MySQL – A reliable and widely used relational database to store users, posts, and comments.
- Tables:
  - Users (user_id, username, password, role)
  - BlogPosts (post_id, title, content, author_id, date)
  - Comments (comment_id, post_id, user_id, comment_text, date)

4. Development Tools

- Visual Studio Code – Code editor for efficient development.
- XAMPP / WAMP (for PHP) or Python Virtual Environment – For running the local development server.

UI Structure/API Scheme Design

UI Structure

1.Home Page

- Displays a list of all published blog posts.
- Each post includes the title, author, date, and a short excerpt.
- Provides a navigation bar with links to Home, Login/Register, and Admin (if applicable).

2.Blog Post Page

- Shows the full content of a selected blog post.
- Displays author details, post date, and main content.
- Includes a comment section where readers can view existing comments and add their own if logged in.

3.Login/Register Page

- Allows new users to register with username and password.
- Provides login functionality for existing users.
- Redirects users back to the home page after successful login.

API Scheme Design

1. Authentication APIs

- POST /api/register → Registers a new user with username and password.
- POST /api/login → Authenticates a user and generates a session or token.
- POST /api/logout → Ends the active session for a user.

2. Blog Post APIs

- GET /api/posts → Retrieves all blog posts.
- GET /api/posts/{id} → Retrieves details of a specific blog post by its ID.
- POST /api/posts → Creates a new blog post (author-only).
- PUT /api/posts/{id} → Updates an existing blog post (author-only).
- DELETE /api/posts/{id} → Deletes a blog post (author/admin).

3. Comment APIs

- GET /api/posts/{id}/comments → Fetches comments for a specific post.
- POST /api/posts/{id}/comments → Adds a new comment to the post (user must be logged in).
- DELETE /api/comments/{id} → Deletes a comment (author/admin).

Data Handling Approach

1. Data Collection

- User Data: Collected during registration (username, password, role).
- Blog Data: Collected when authors create or edit blog posts (title, content, date, author).
- Comment Data: Collected when readers submit comments (comment text, user ID, post ID, timestamp).

2. Data Storage

- All data is stored in a relational database (MySQL).
- Tables include:
  - Users (UserID, Username, Password, Role)
  - BlogPosts (PostID, Title, Content, AuthorID, DateCreated)
  - Comments (CommentID, PostID, UserID, CommentText, DateCreated)
- Relationships:
  - One User → Many BlogPosts
  - One BlogPost → Many Comments

3. Data Retrieval & Processing

- Frontend requests data via API calls.
- Backend validates the request, processes logic (authentication, permissions), and queries the database.
- Database returns the requested information, which is then formatted and sent back to the frontend.
- Example:
  - When a user opens a blog post, the system retrieves the post details and its associated comments.

Component / Module Diagram

1. User Module

- Handles registration, login, and logout.
- Manages user roles (Reader, Author, Admin).
- Ensures only authorized users can access restricted features like posting or moderating.

2. Blog Management Module

- Allows authors to create, edit, and delete blog posts.
- Displays blog posts to readers on the home page and blog detail page.
- Manages post metadata (title, author, date).

3. Comment Module

- Enables readers to add comments on blog posts.
- Displays all comments for each post in chronological order.
- Provides functionality for authors/admins to delete inappropriate comments.

4. Admin Module

- Provides administrative access to manage users, posts, and comments.
- Includes user blocking/unblocking, post deletion, and comment moderation.
- Ensures the platform maintains quality and security.

Basic Flow Diagram