**COLLEGE CODE :** 9111

**COLLEGE NAME :** SRM Madurai College for Engineering and Technology

**DEPARTMENT :** Computer Science and Engineering

**STUDENT NM-ID : 4BF0134536EB1DB9BD0436F56619B6E2**

**ROLL NO :** 911123104010

**DATE :** 06/10/2025

**Completed the project named as Phase 5**

**TECHNOLOGY PROJECT NAME** : IBM-FE-Blog Site with Comment Section

**SUBMITTED BY,**

**NAME :** Gokularam M

**MOBILE :** 8870213343

# IBM-FE-Blog Site with Comment Section

# Phase 5 Report

**1. Final Demo Walkthrough**
**Overview:**
DevBlog is a modern blogging platform built for developers to share knowledge, insights, and tutorials. It provides user authentication, post creation, comment sections, trending posts, search and filter options, and a save/bookmark feature.

**Key User Roles:**
- **Guest Users:** Can view posts, search, and read comments.
- **Registered Users:** Can create posts, comment, save blogs, and manage their profile.

**Walkthrough Steps:**
1. **Home Page:**
   Displays all blog posts with featured/trending highlights. Users can browse posts or use filters.
2. **Login / Signup Page:**
   Allows users to register or log in securely using Supabase Authentication.
3. **Create Post:**
   Authenticated users can create and publish new blog posts with title, description, tags, and cover image.
4. **Search & Filter:**
   Search bar helps users find posts by keywords, tags, or authors.
5. **Trending Section:**
   Displays top posts based on engagement metrics like likes and comments.
6. **Comment Section:**
   Users can comment on posts, reply to others, and discuss ideas.
7. **Save / Bookmark Feature:**
   Users can save favorite posts for quick access later.
8. **Logout:**
   Ends the user session and redirects back to the homepage.

---

**2. Project Report**

**Project Objective:**
To design and develop an interactive blogging platform that enables developers to share and engage with content easily.

**Motivation:**
Developers often need a clean, ad-free, and open-source environment to write blogs, share tutorials, and interact with a technical audience. DevBlog bridges this gap.

**Scope:**

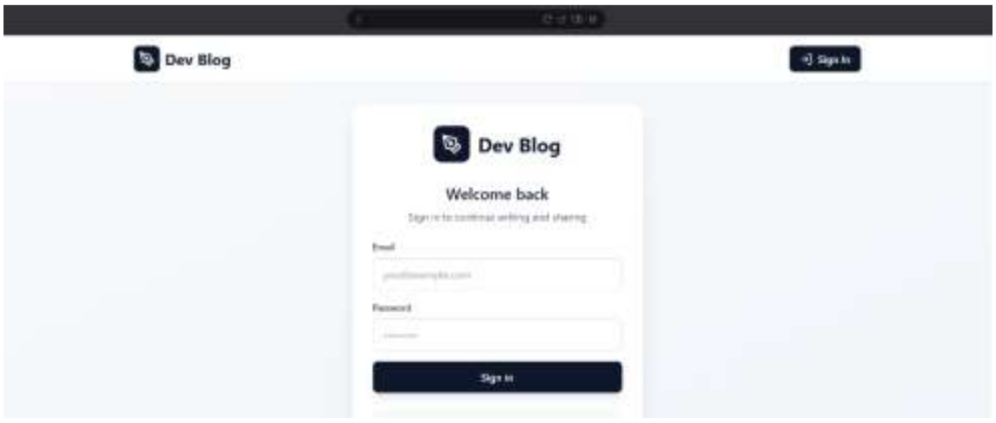
- Secure authentication (login/register)
- CRUD operations for blog posts
- Comment and save features
- Trending post algorithm
- Fully responsive UI

**Modules:**

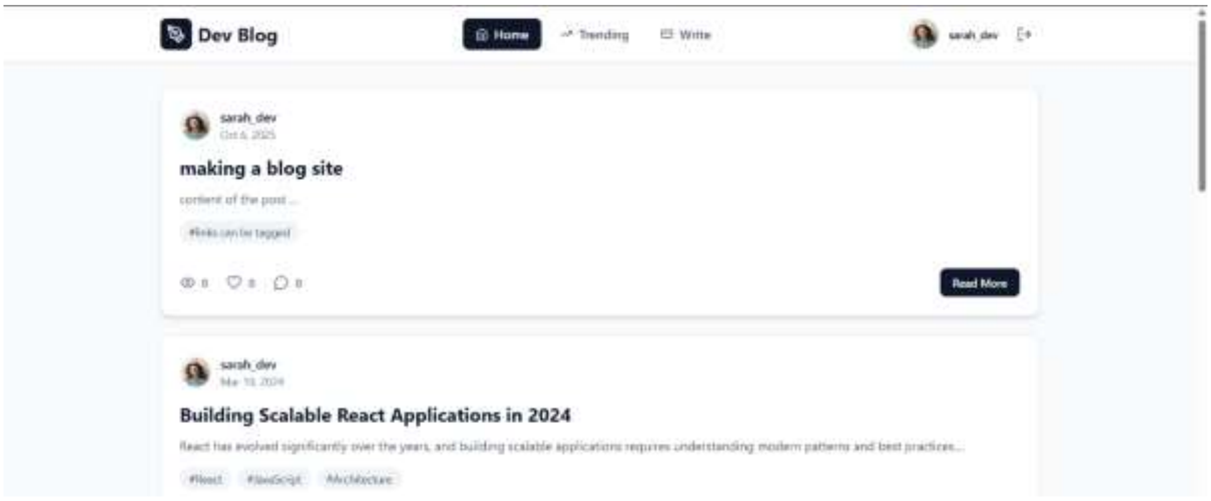
1. **Authentication Module** – Login, Register, Logout via Supabase.
2. **Blog Management Module** – Create, Read, Update, Delete blog posts.
3. **Comment Module** – Real-time comment system using Supabase database.
4. **Search & Filter Module** – Keyword and tag-based search functionality.
5. **Save/Bookmark Module** – User-specific saved posts collection.
6. **Trending Module** – Ranks popular posts based on likes/comments.

## 3. Screenshots / API Documentation

### Login Page



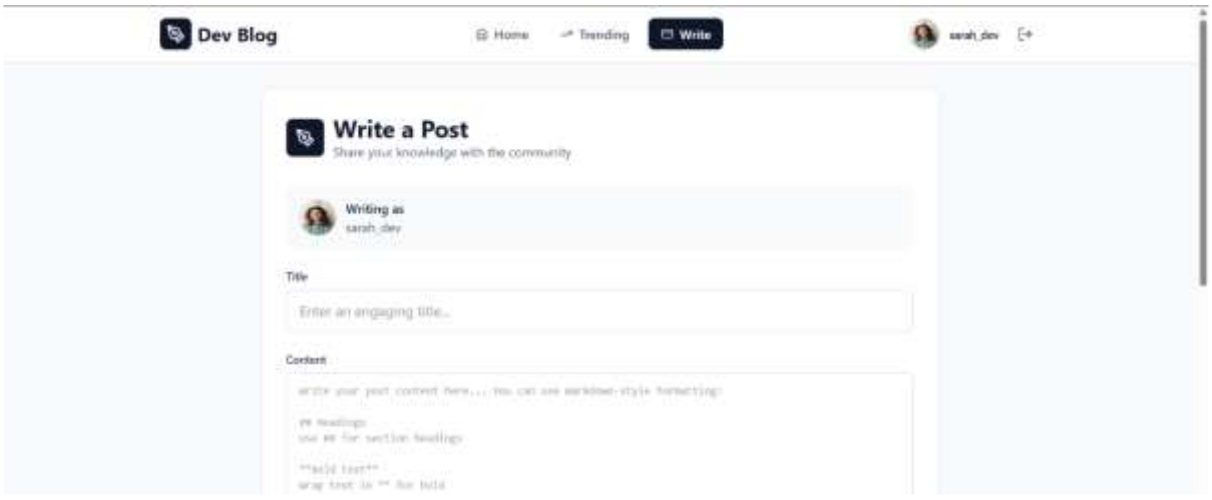
### Home Page



sarah_dev
Oct 6, 2025
making a blog site
content of the post ...
#links can be tagged

sarah_dev
Mar 15, 2024
Building Scalable React Applications in 2024
React has evolved significantly over the years, and building scalable applications requires understanding modern patterns and best practices...
#React #JavaScript #Architecture

## Create Post



Write a Post

Share your knowledge with the community

Writing as

sarah_dev

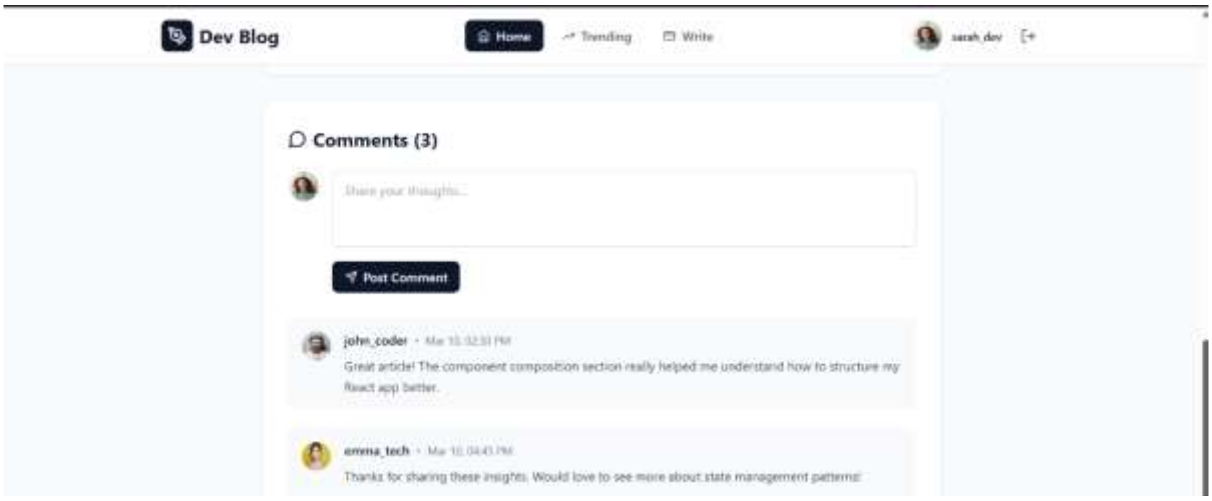
Title

Enter an engaging title...

Content

```
Write your post content here... You can use markdown-style formatting:

## Headings
Use ## for section headings

**Bold text**
Wrap text in ** for bold
```

## Comment Section



Comments (3)

Share your thoughts...

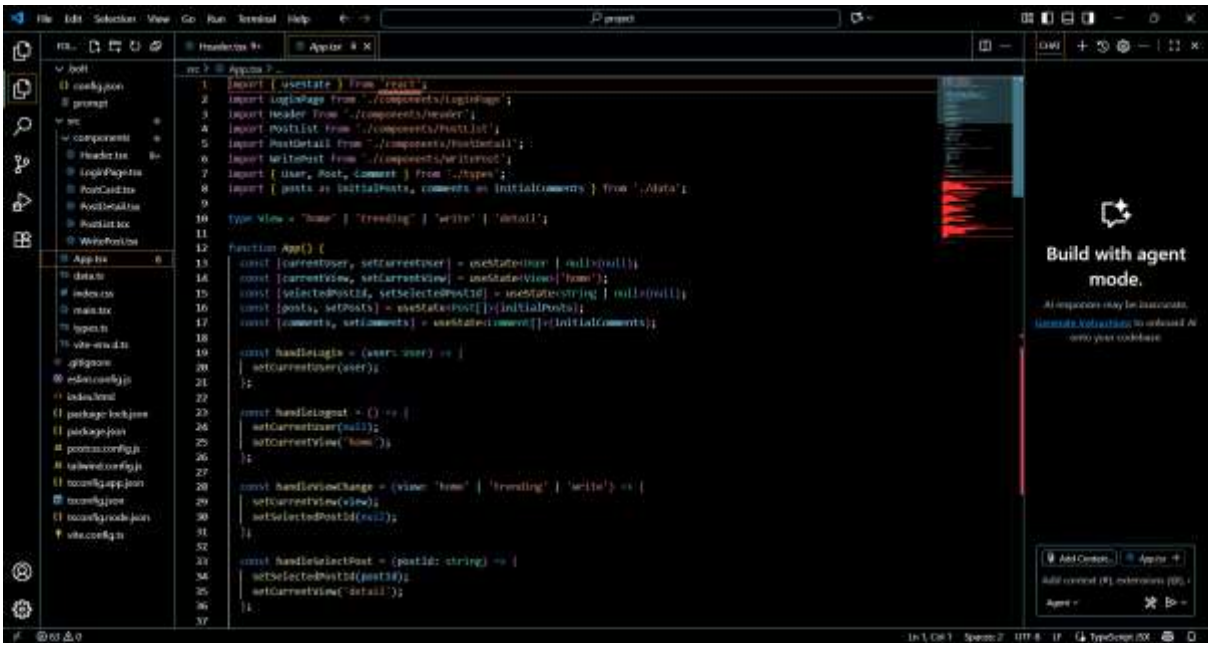
john_coder · Mar 10, 02:30 PM

Great article! The component composition section really helped me understand how to structure my React app better.

emma_tech · Mar 10, 04:45 PM

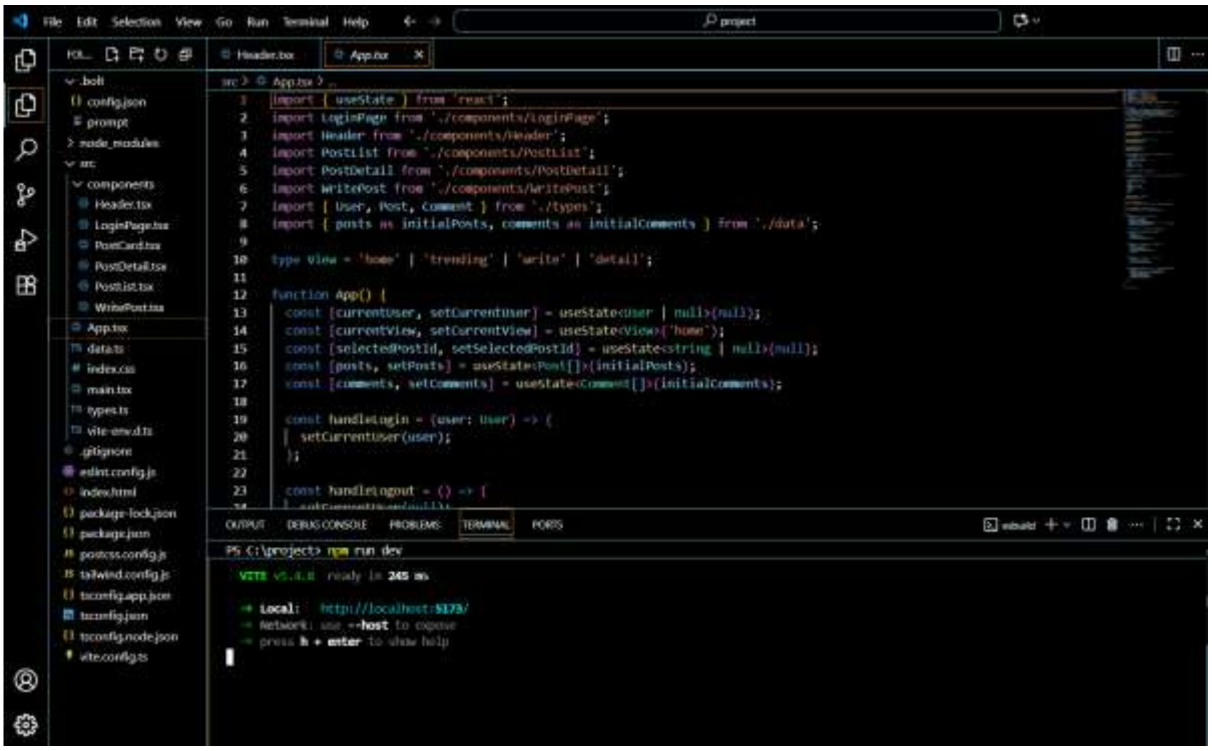
Thanks for sharing these insights. Would love to see more about state management patterns!

## API Documentation



Build with agent mode.
AI responses may be inaccurate.
Generate Instructions to onboard AI onto your codebase.
Add Context...
Agent

```
import { useState } from 'react';
import LoginPage from './components/LoginPage';
import Header from './components/Header';
import PostList from './components/PostList';
import PostDetail from './components/PostDetail';
import WritePost from './components/WritePost';
import { User, Post, Comment } from './types';
import { posts as initialPosts, comments as initialComments } from './data';

type View = 'home' | 'trending' | 'write' | 'detail';

function App() {
  const [currentUser, setCurrentUser] = useState<User | null>(null);
  const [currentView, setCurrentView] = useState<View>('home');
  const [selectedPostId, setSelectedPostId] = useState<string | null>(null);
  const [posts, setPosts] = useState<Post[]>(initialPosts);
  const [comments, setComments] = useState<Comment[]>(initialComments);

  const handleLogin = (user: User) => {
    setCurrentUser(user);
  };

  const handleLogout = () => {
    setCurrentUser(null);
    setCurrentView('home');
  };

  const handleViewChange = (view: 'home' | 'trending' | 'write') => {
    setCurrentView(view);
    setSelectedPostId(null);
  };

  const handleSelectPost = (postId: string) => {
    setSelectedPostId(postId);
    setCurrentView('detail');
  };

```



```
import { useState } from 'react';
import LoginPage from './components/LoginPage';
import Header from './components/Header';
import PostList from './components/PostList';
import PostDetail from './components/PostDetail';
import WritePost from './components/WritePost';
import { User, Post, Comment } from './types';
import { posts as initialPosts, comments as initialComments } from './data';

type View = 'home' | 'trending' | 'write' | 'detail';

function App() {
  const [currentUser, setCurrentUser] = useState<User | null>(null);
  const [currentView, setCurrentView] = useState<View>('home');
  const [selectedPostId, setSelectedPostId] = useState<string | null>(null);
  const [posts, setPosts] = useState<Post[]>(initialPosts);
  const [comments, setComments] = useState<Comment[]>(initialComments);

  const handleLogin = (user: User) => {
    setCurrentUser(user);
  };

  const handleLogout = () => {
```

```
PS C:\project> npm run dev

  VITE v5.4.8  ready in 245 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help
```

**Base URL:** `http://localhost:5173/api`

| Endpoint | Method | Description | Request Body | Response |
|----------|--------|-------------|--------------|---------|
| `/auth/register` | POST | Register a new user | `{ "username": "string", "email": "string", "password": "string" }` | `200 OK` - Returns user object |
| `/auth/login` | POST | Login user | `{ "email": "string", "password": "string" }` | `200 OK` - Returns JWT token |
| `/posts` | GET | Get all posts | - | Array of post objects |
| `/posts` | POST | Create new post | `{ "title": "string", "content": "string", "tags": ["string"] }` | `201 Created` - Returns created post |
| `/posts/:id` | PUT | Update a post | `{ "title": "string", "content": "string", "tags": ["string"] }` | `200 OK` - Returns updated post |
| `/posts/:id` | DELETE | Delete a post | - | `200 OK` - Post deleted |
| `/posts/:id/comments` | POST | Add comment to post | `{ "text": "string" }` | `201 Created` - Returns created comment |
| `/posts/:postId/comments/:commentId` | PUT | Update comment | `{ "text": "string" }` | `200 OK` - Returns updated comment |
| `/posts/:postId/comments/:commentId` | DELETE | Delete comment | - | `200 OK` - Comment deleted |
| `/posts/trending` | GET | Get trending posts | - | Array of post objects |

# 4. GitHub README & Setup Guide

## 4.1 README Structure

1. Project Title
2. Description
3. Features
4. Technologies Used
5. Installation / Setup
    - git clone <repo_url>
    - npm install
    - npm start
6. Usage Instructions
7. Screenshots
8. Demo Link / Deployed URL
9. License

## 5. Final Submission

DEMO LINK:
https://drive.google.com/drive/folders/17MnaZJhBP7niGV9FilcbNHQ8qyyitnxR?usp=drive_link

### Github id's

| NAME | GITHUB ID |
|---|---|
| Aanandha ruban MRK | https://github.com/aanandharuban |
| Gokularam M | https://github.com/gokularam12/gokularamrm |
| Mahesh MSK | https://github.com/Mahee0117 |
| Sanjay Karthi RP | https://github.com/sanjaykarthirp |
| Aathish Rao AB | https://github.com/Aathish14 |