

**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

**A FINAL REPORT ON**

“**My Blogging System”**

**Submitted to:**

**Caribbean College of Management**

***In partial fulfillment of the requirements for the bachelor’s in computer***

***Application***

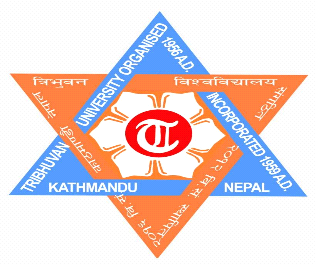
**Submitted by:**

Susan Thakuri

Urgen Tamang

**Under the Supervision of**

Mr. Sampurna Shakya

  
  
**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

**College Name**

**Supervisor’s Recommendation**

I hereby recommend that this project prepared under my supervision by Susan Thakuri and Urgen Tamang entitled “**My Blogging System ''** in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

**…..................................................**

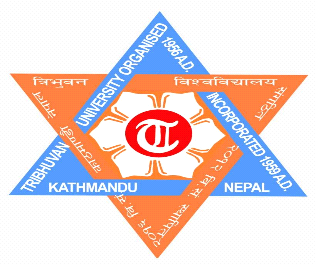
Mr. Sampurna Shakya

**SUPERVISOR**

Caribbean College of Management

BCA

Mahalaxmisthani, Lalitpur



**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

**College Name**

**LETTER OF APPROVAL**

This is to certify that this project prepared by Susan Thakuri and Urgen Tamang entitled “**My Blogging System”** in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

|  |  |
| --- | --- |
| **…..................................................** | **…..................................................** |
| Mr. Sampurna Shakya | Ganesh Shrestha (HOD of BCA) |
| BCA | BCA |
| Caribbean college | Caribbean college |
| Mahalaxmisthan, Lalitpur | Mahalaxmisthan, Lalitpur |
|  |  |
| **…..................................................** | **…..................................................** |
| **--------------** | **--------------** |
| **Internal Examiner** | **External Examiner** |
|  |  |

# Acknowledgement

We would like to express my gratitude and appreciation to all those who gave us the possibility to complete this project. Special thanks to my supervisor Mr. Sampurna Shakya whose help, simulating suggestions and encouragement helped us in all time of fabrication process and in writing this project. We also sincerely thank you for the time spent proofreading and correcting my many mistakes.

We specially acknowledge him for his advice, supervision and the vital contribution as and when required during this project. His involvement has triggered and nourished intellectual maturity that will help me for a long time to come.

# Abstract

**Building a User-Friendly Blogging System**

This project is all about making a simple and easy-to-use Blogging System with fun features. People can easily sign up, write and share blog posts, and talk to others through comments. Bloggers can also change how their blogs look and share stuff on social media.

We'll do this step by step, starting with getting the project going, finding out what people want, designing the system, making it, checking it, and finally, putting it out there. Our aim is to create a reliable system that makes blogging enjoyable for everyone.

At the end of this project, we want to have a really great Blogging System that makes it simple for people to share their thoughts and stories online.

*Keywords: My Blogging System, Portfolio, information sharing, Personal Growth, Lea*

**Table of Contents**

[Acknowledgement i](#_Toc156678700)

[Abstract ii](#_Toc156678701)

[List of Figures v](#_Toc156678702)

[List of Tables vi](#_Toc156678703)

[List of Abbreviations vii](#_Toc156678704)

[Chapter: 1 Introduction 1](#_Toc156678705)

[1.1 Introduction 1](#_Toc156678706)

[1.2 Problem Statement 1](#_Toc156678707)

[1.3 Objectives 1](#_Toc156678708)

[1.4 Scope and Limitations 2](#_Toc156678709)

[1.4.1 Scope 2](#_Toc156678710)

[1.4.2 Limitations 2](#_Toc156678711)

[1.5 Report Organization 2](#_Toc156678712)

[Chapter 2: Background Study and Literature Review 4](#_Toc156678713)

[2.1 Background Study 4](#_Toc156678714)

[2.1.1 Fundamental Theories: 4](#_Toc156678715)

[2.2 Literature Review 4](#_Toc156678716)

[2.2.1 Review of Similar Projects: 4](#_Toc156678717)

[Chapter 3: System Analysis and Design 5](#_Toc156678718)

[3.1 System Analysis 5](#_Toc156678719)

[3.1.1 Requirement Analysis 6](#_Toc156678720)

[3.1.2 Feasibility Analysis 8](#_Toc156678721)

[3.1.3. Data Modeling: ER Diagram 9](#_Toc156678722)

[3.1.4. Process Modelling (DFD): 10](#_Toc156678723)

[3.2. System Design 10](#_Toc156678724)

[3.2.1. Architectural Design 10](#_Toc156678725)

[3.2.2. Database Schema Design 11](#_Toc156678726)

[3.2.3. Interface Design (UI Interface / Interface Structure Diagrams) 11](#_Toc156678727)

[Chapter: 4 Implementation and Testing 16](#_Toc156678728)

[4.1. Implementation 16](#_Toc156678729)

[4.1.1. Technologies Used 16](#_Toc156678730)

[4.1.3. Database Platforms: 17](#_Toc156678731)

[4.1.2. Implementation Details of Modules 17](#_Toc156678732)

[4.2. Testing 18](#_Toc156678733)

[4.2.1. Test Cases for Unit Testing 18](#_Toc156678734)

[4.2.2. Test Cases for System Testing 19](#_Toc156678735)

[Chapter: 5 Conclusion and Future Recommendations 20](#_Toc156678736)

[5.1. Outcome: 20](#_Toc156678737)

[5.2 Conclusion: 20](#_Toc156678738)

[5.3. Future Recommendations 20](#_Toc156678739)

[Appendices 21](#_Toc156678740)

[Bibliography 33](#_Toc156678741)

# 

# 

# List of Figures

[Figure 1: Agile Methodology 5](#_Toc156678742)

[Figure 2: Use Case Diagram 7](#_Toc156678743)

[Figure 3: Er Diagram 9](#_Toc156678744)

[Figure 4: DFD 10](#_Toc156678745)

[Figure 5: Architectural Design 10](#_Toc156678746)

[Figure 6: Database Schema Design 11](#_Toc156678747)

# List of Tables

[Table 1: Gantt Chart 9](#_Toc156678748)

[Table 2: Test Case For Unit Testing 18](#_Toc156678749)

[Table 3: Test Case For System Testing 19](#_Toc156678750)

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# List of Abbreviations

|  |  |
| --- | --- |
| **Abbreviations** | **Full-Form** |
| CSS | Cascading Style Sheet |
| DFD | Data Flow Diagram |
| Er-Diagram | Entity Relation Diagram |
| HTML | Hypertext Markup Language |
| IDE | Integrated Development Environment |
| MYSQL | My Structured Query Language |
| PHP | Hypertext Preprocessor |
| UI | User Interface |
| UX | User Experience |
| JS | Java Script |

# Chapter: 1 Introduction

## 1.1 Introduction

In today's digital world, blogging is an important way for people and organizations to share ideas globally. With the always-changing digital world, there's a need for a new and easy-to-use blogging system. This document introduces a modern platform designed to go beyond usual limits and redefine digital expression.

This blogging system is more than just a tool; it's a response to the changing needs of content creators and consumers. We want to make the user experience better, make it easier to create content, and offer cool features. Our goal is to create a space where ideas flow easily, creating a lively online community. At its core, this system is committed to innovation, trying to change how people interact with digital content through simple interfaces, personalization options, and lots of features. It's not just a platform; it's a change in how we communicate and express ourselves online.

## 1.2 Problem Statement

Regular blogging platforms have problems that make it hard for people who write and those who read to connect smoothly. Complicated buttons, not much freedom to make things look how you want, and tools for managing content that aren't so good create problems for saying what you want. Our blogging system is a purpose-driven solution that aims to deal with and overcome these issues.

The real challenge is the gap between what regular platforms offer and what people really need. Our blogging system wants to close that gap by making things simpler, breaking down problems, and giving an easy platform with lots of features. It's not just about fixing current problems; it's about really helping a bunch of different people who make and read stuff. Through smart design, strong features, and a commitment to a different way of doing things, this system aims to bring in a new time of talking online, one that is not only easy to use but also makes things better and works the way you want.

## 1.3 Objectives

The primary objectives of the blogging system are:

1. Information sharing
2. Content marketing and brand building
3. Personal development

## 1.4 Scope and Limitations

### 1.4.1 Scope

The blogging system will include features for:

* User registration and authentication
* Content creation, editing, and publishing
* Social media integration
* Commenting and community engagement
* Search for the post

### 1.4.2 Limitations

The initial version may have limitations such as:

* No customizations
* Can only promote one image per posts
* Limited third-party integrations
* Single-language support

## 1.5 Report Organization

This report is organized in 5 different chapters based on the official documentation as per

Tribhuvan University.

**Chapter 1: Introduction**

In this chapter the introduction about the project and some related topics are illustrated

like why this project, what were the problems in projects and where can it be used, how

this project idea came up and how it is built.

**Chapter 2: Background Study and Literature Review**

In this chapter the history of the ideas used in the project along with the previous research

and implementation of projects similar to this are illustrated. This chapter contains a

study of several other projects with similar ideas, topics and terminologies.

**Chapter 3: System Analysis and Design**

In this chapter the overall system specifications are explained with data in the form of

text as well as pictures. This chapter consists of all the requirements, modeling of the

architecture and system workflow is defined. This chapter tends to explain if the project

is feasible or not along with diagrams like, DFD, ER Diagram, Interface design, database

schema design and architectural modeling.

**Chapter 4: Implementation and Testing**

In this chapter the hardware and software that are in use or used to build this project is

summed up along with modular testing and integration testing which tests different test

scenarios to deliver the principle of the system performance and functioning.

**Chapter 5: Conclusion and Future Recommendation**

In this chapter different information gained with the projects and to which extent it can be

extended or contracted is specified. Different implementations that can be used in future

to make this project more sophisticated and evolving are described along with the final

outcome of the project in a real-world scenario.

# Chapter 2: Background Study and Literature Review

## 2.1 Background Study

### 2.1.1 Fundamental Theories:

**Design for Users:** Our system is all about designing with users in mind. We use a thoughtful process that pays attention to how users behave, what they like, and what they need. By making user experience a top priority, we make sure the system isn't just functional but also easy and enjoyable for creating and consuming content.

**Smart Content Handling:** Our system goes beyond just helping you create content; it's smart about organizing and presenting it too. We consider things like who the audience is, the tone of the content, and keeping things consistent. Every piece of content adds value to the bigger story, fitting into our overall content strategy.

**Responsive Design:** We've designed our system to work smoothly on any device. Whether you're using a computer, tablet, or phone, the system adjusts seamlessly to different screen sizes. This way, everyone gets the best experience, no matter what kind of device they're using. It's about making the platform welcoming and accessible in this age of diverse digital gadgets.

## 2.2 Literature Review

### 2.2.1 Review of Similar Projects:

**WordPress:** We're looking closely at WordPress, a big player in content management, to guide our system's development. By analyzing what works well and what doesn't in WordPress, we learn important lessons about scalability, flexibility, and user-friendly interfaces. Our goal is to build on these lessons, bringing in new features while keeping the reliability that makes WordPress stand out.

**Medium:** We're closely studying Medium's simple yet captivating design to shape the look of our blogging system. Finding a balance between simplicity and usefulness, our platform takes inspiration from Medium's emphasis on presenting content in a compelling way. The goal is to create an engaging reading experience for users.

# Chapter 3: System Analysis and Design

## 3.1 System Analysis

Since this project involves design and implementation of a software system regardless that is well-based, it will be important to mention and consider some models used in software development and deployment, some general models of software development are namely:

* **The Waterfall approach:** It represents activities in requirements, specifications, design, implementation, and testing. All these are separate processes.
* **Incremental / Evolutionary development:** It involves a rapid development of the specifications and then refined later for the customer.
* **Formal transformation:** This approach is based on setting and producing a mathematical specification for the system to meet.
* **Agile methodolog**y: project management approach that prioritizes cross-functional collaboration and continuous improvement.

As we work on making our blogging system, we're using a way of doing things called Agile. It's like a super flexible and adaptable method that helps us quickly adjust to what users want.

****

Figure 1: Agile Methodology

In the world of blogging, Agile means we identify problems and find solutions together with our team. We want our blogging system to run smoothly and work really well. With Agile, we're always getting feedback to make sure our content is efficient and our system is always improving.

Agile planning, a big part of this approach, means we're not sticking to one big plan. Instead, we make small changes as we go, which is great for a blogging system because content needs can change a lot. We listen to feedback from users all the time to make sure our blogging platform keeps up with what people want and what's popular in the blogging world.

### 3.1.1 Requirement Analysis

The requirement analysis phase focuses on understanding and documenting the needs of both bloggers and readers. Through comprehensive use case diagrams and lists, the functional and non-functional requirements are identified and clarified.

**Functional Requirements:**

**Users:**

Registered User (Only type of user, with full access)

**Use Cases:**

* **Create and Publish Blog Post:** Similar to before, this use case describes how a registered user can create and publish blog posts on the platform.
* **Edit and Delete Blog Post:** Registered users can edit and delete their own blog posts.
* **Comment on Blog Post:** Only registered users can read and comment on blog posts.
* **Love Post:** Similar to before, registered users can show appreciation for posts by clicking a "Love" button.
* **Register and Log In:** New users can register to create an account, while existing users can log in with their credentials.
* **Manage Profile:** Registered users can access their profile page to edit personal information, change profile picture, update preferences, and view their published blog posts and comments.
* **Sharing the post:** Users can share the post in social media.
* **Search the Post:** Users can now search for any post on the website.

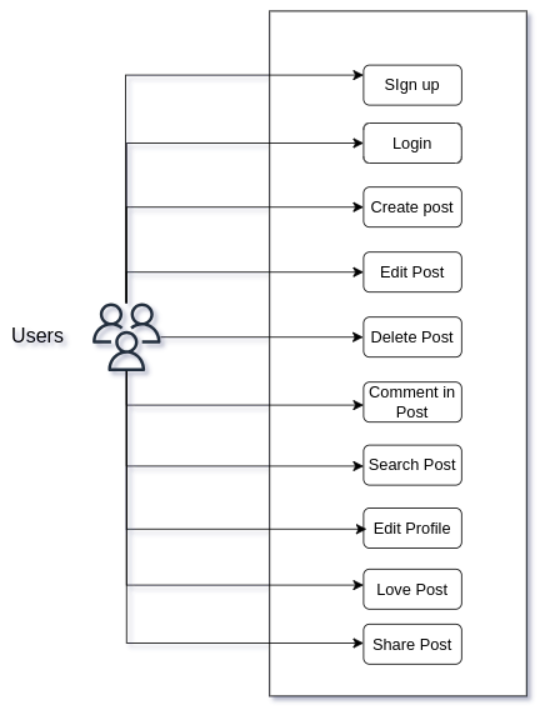


Figure 2: Use Case Diagram

**Non-Functional Requirements:**

* **Performance:** Ensuring fast page loading times for a smooth user experience.
* **Usability:** The user interface should still be intuitive and easy to navigate for registered users who have full access to all functionalities.
* **Security:** User data and blog posts should still be protected with secure authentication, password hashing, and authorization mechanisms.
* **Scalability:** The system should be able to handle a growing number of registered users and blog posts without performance degradation.

### 3.1.2 Feasibility Analysis

**Technical Feasibility:**

* **Hardware:** Check if our servers can handle all the users and visitors. Make sure we can grow without a problem.
* **Software:** List the tools and tech we need (like programming languages and databases) and check if they're good for our project.
* **Skills:** See if our team can do it or if we need to hire experts.
* **Existing Tools:** Look for any free tools we can use to save time and money.
* **Challenges:** Think about problems like hacking or things slowing down and plan how to solve them.

**Operational Feasibility:**

* **System Maintenance:** Figure out what we need to keep things running, like backups and helping users.
* **Hosting:** Choose a good company to keep our website up and running fast.
* **Data Management:** Decide how to keep our information safe, fast, and easy to get.
* **Monitoring:** Watch how everything is working, see what users are doing, and catch problems early.

**Schedule Feasibility:**

* **Project Plan:** Plan with small steps and real deadlines. Think about problems and when we can do things.
* **Risks**: List what could go wrong and plan to stop it or fix it.
* **Resources:** Make sure we have enough people and tools when we need them.
* **Talk Right:** Make a clear way for the team and others to talk and fix problems.
* **Testing:** Try everything many times before showing it to everyone.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task axis** | Requirement and Analysis | **1 Week** |  |  |  |  |  |  |  |
| System Design |  | **4 Weeks** | | |  |  |  |  |
| Implementation |  |  |  |  |  | **2 Weeks** | |  |
| Testing |  |  |  |  |  |  |  | **1 Week** |
| Deployment |  |  |  |  |  |  |  | **1 Week** |
|  |  | W1 | W2 | W3 | W4 | W5 | W6 | W7 | W8 |
| **Task Axis** | | | | | | | | | |

Table 1: Gantt Chart

### 3.1.3. Data Modeling: ER Diagram

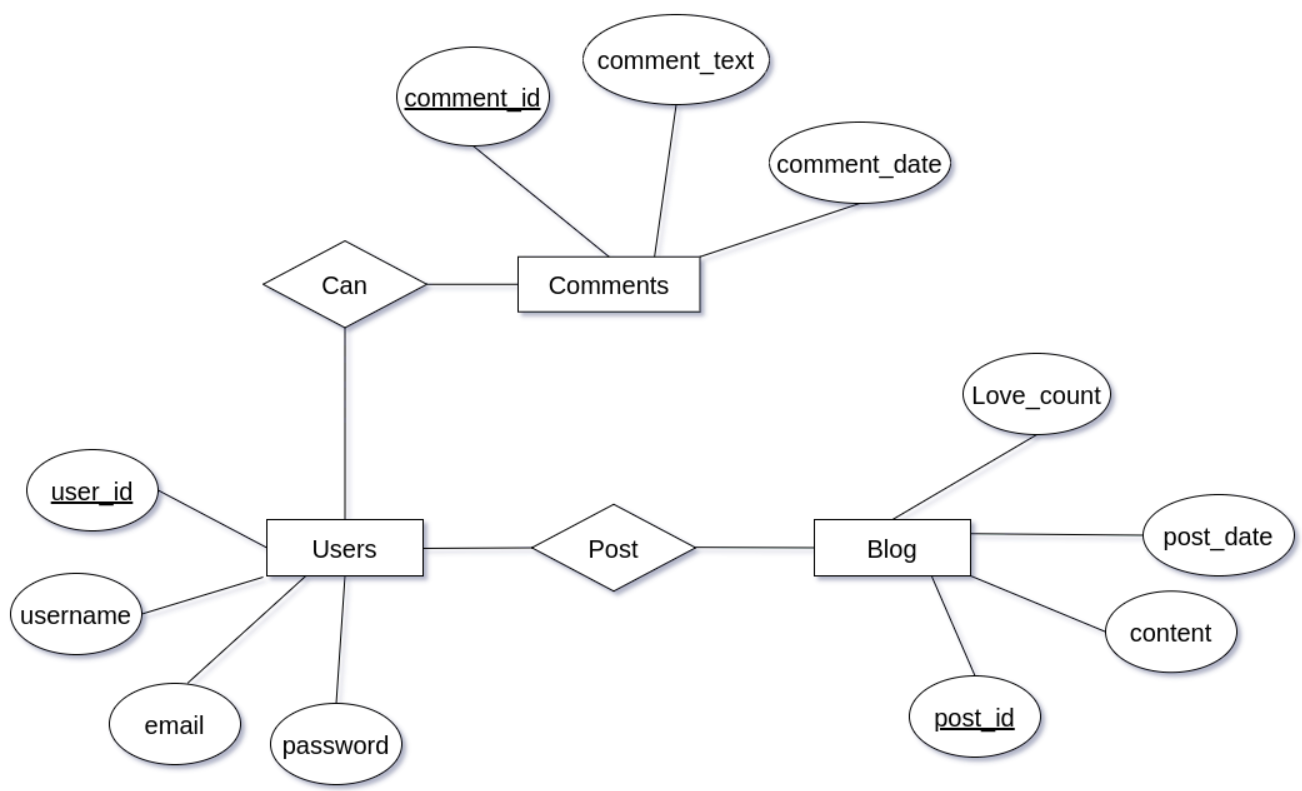


Figure 3: Er Diagram

### 3.1.4. Process Modelling (DFD):

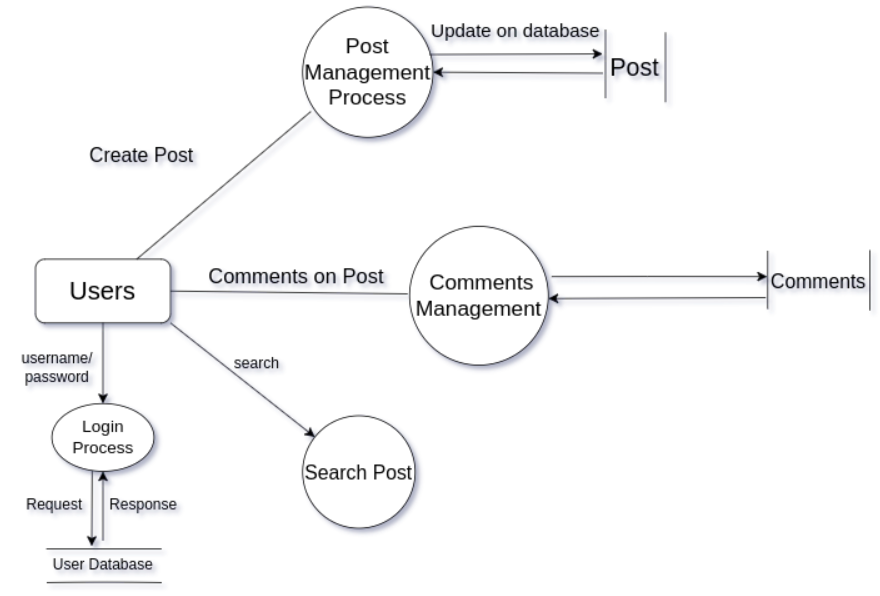


Figure 4: DFD

## 3.2. System Design

### 3.2.1. Architectural Design

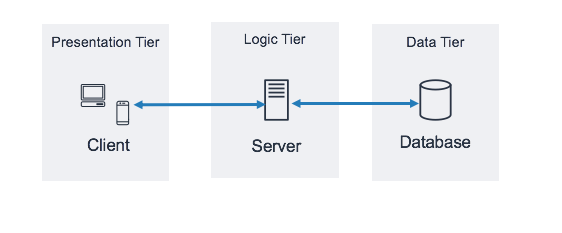
****

Figure 5: Architectural Design

### 3.2.2. Database Schema Design

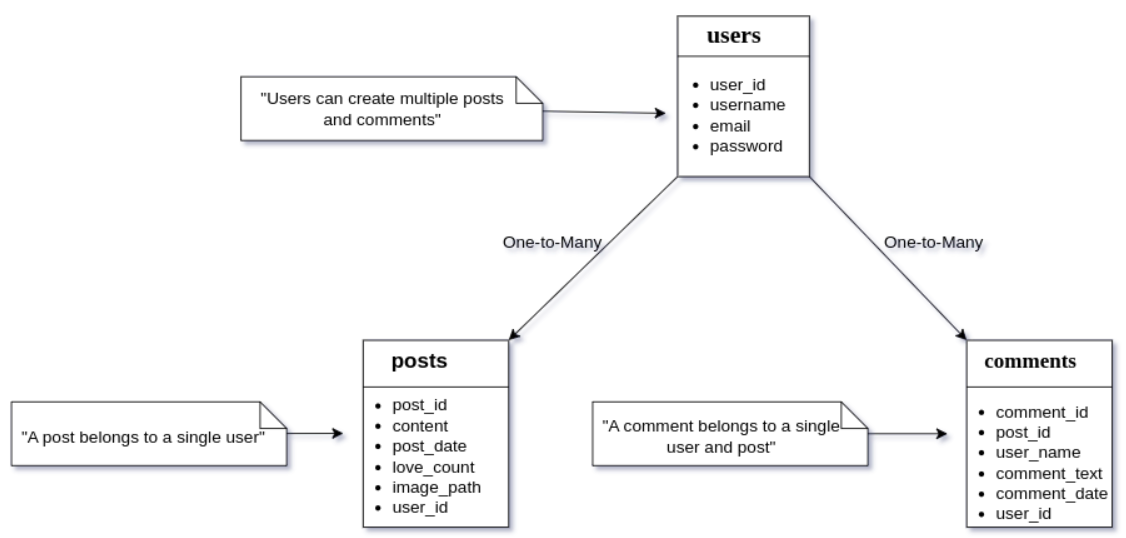
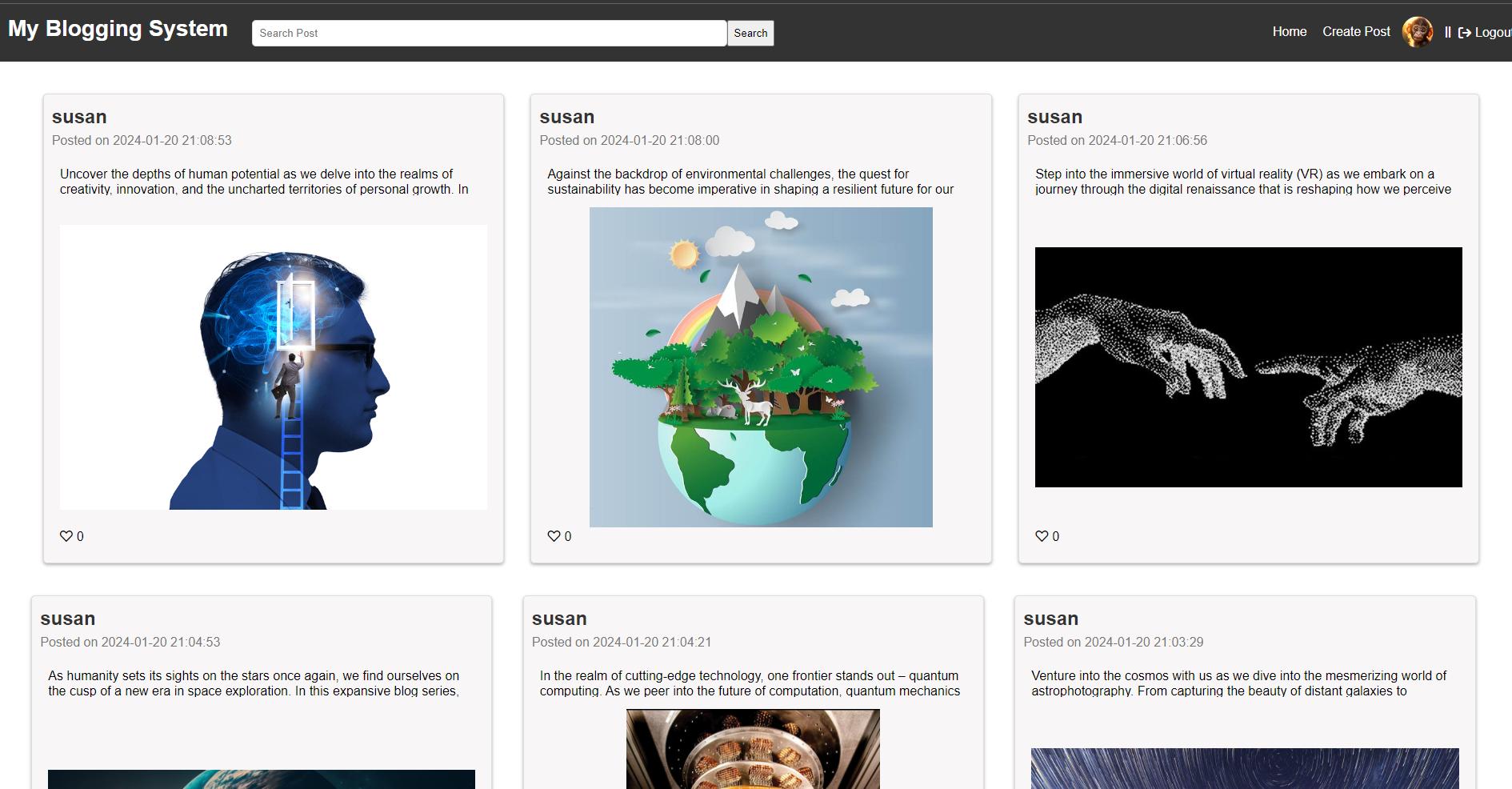
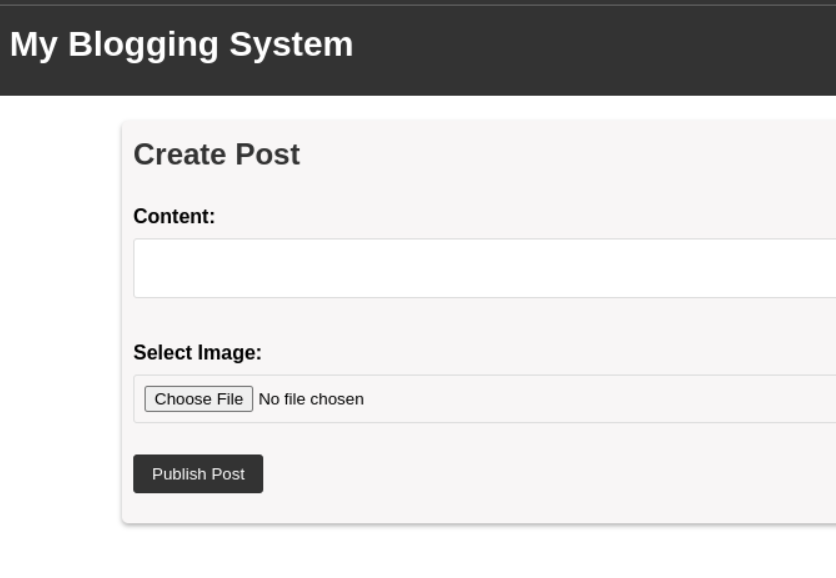
****

Figure 6: Database Schema Design

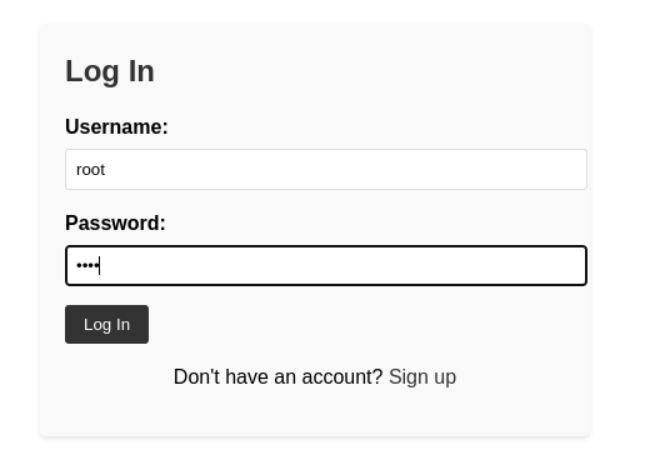
### 3.2.3. Interface Design (UI Interface / Interface Structure Diagrams)



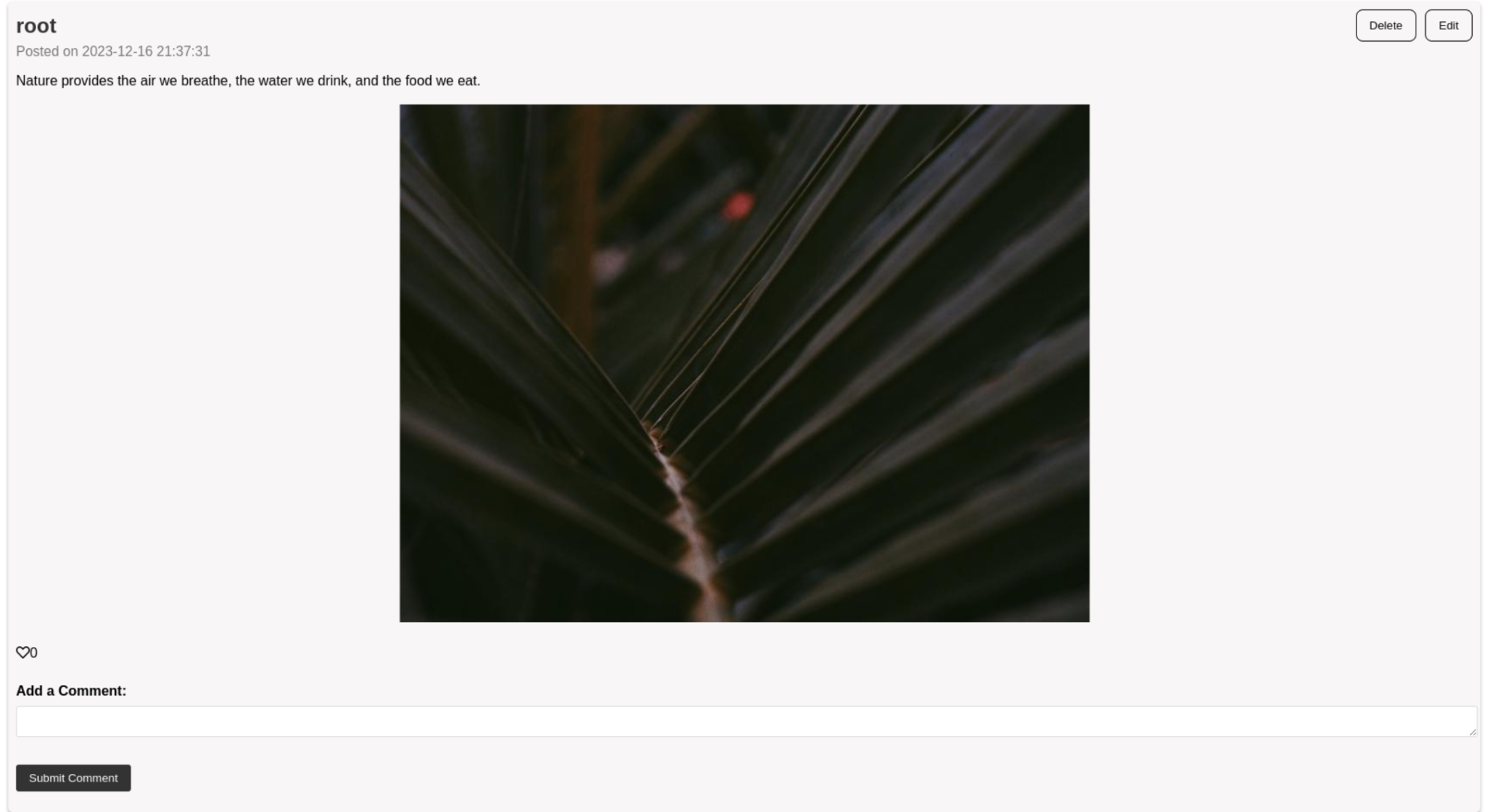
**Home (Main Page For The Post)**

****

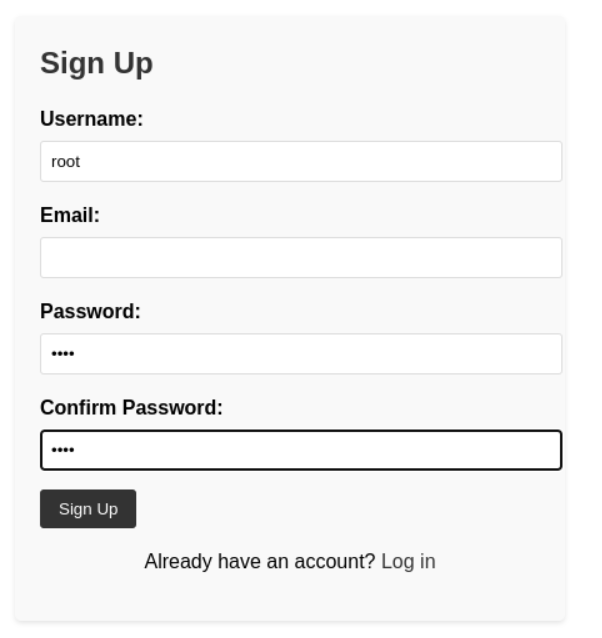
**Create Post Page**

****

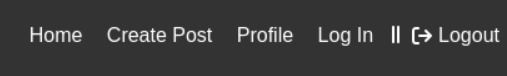
**Login Page**

****

**View Complete Post with Delete, Edit and Comments**

****

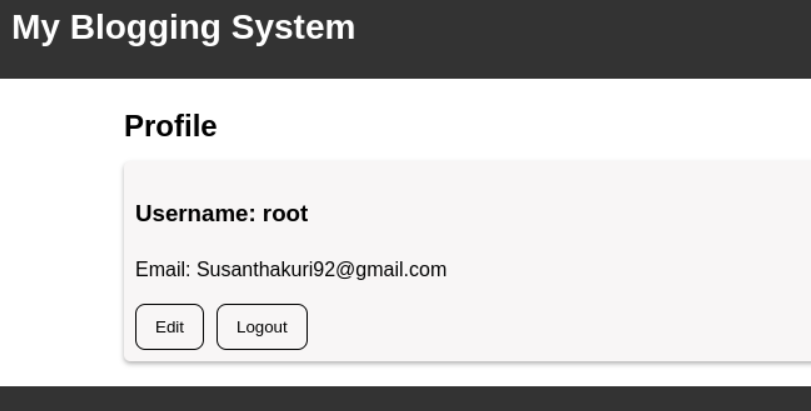
**Signup Page**



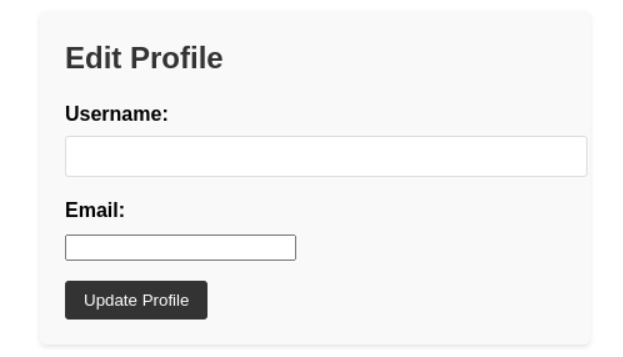
**Nav Bar**

****

**Search Option**

****

**Profile Page**

****

**Profile Edit Page**

# Chapter: 4 Implementation and Testing

## 4.1. Implementation

### 4.1.1. Technologies Used

|  |  |
| --- | --- |
| Used For | Tools and technologies |
| IDE | Visual Studio Code |
| Development Environment | Apache Server |
| Diagram Tools |  |
| Documentation Tool |  |
| Web Testing Tool |  |
| Programming Languages |  |

### 4.1.3. Database Platforms:

**MySQL (MariaDB):** MariaDB, a fork of MySQL, was employed as the relational database management system. It facilitated the organized storage of data, including user information, blog content, and system configurations.

### 4.1.2. Implementation Details of Modules

Each module of the blogging system was meticulously designed to fulfill specific functionalities, contributing to a cohesive and feature-rich platform.

**User Authentication:**  robust user authentication module was implemented using PHP sessions for secure and seamless user access.

**Content Management:** The content management module included procedures for creating, editing, and publishing blog posts. A version control system was implemented to track changes and revisions.

**User login:** Users can send login requests to the server with their email and password and the server will check the user database and if any match id is found and password doesn’t match error message is shown if the email and password match it directs you to homepage and stores.

**Create new user:** User register can be done from the register page after the form is filled the data is checked if the data is duplicate or valid if all the conditions match the user register is successful and the data is sent to the database in users table and directed to the home page.

**Performance and Scalability:** The system was optimized for performance and scalability, utilizing caching mechanisms, asynchronous processing, and load balancing to ensure a smooth experience even with a growing user base and content volume.

The combination of these tools and implementation details contributed to a robust, efficient, and user-friendly blogging system that met the specified objectives and requirements.

## 4.2. Testing

### 4.2.1. Test Cases for Unit Testing

Unit testing is a crucial phase in the software development life cycle, focusing on individual components to ensure their correctness and functionality.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N | Test Description | Input | Output | Result |
| 1 | Login with valid credentials | Enter valid username and password | Successfully logged in | Pass |
| 2 | View blog posts after login | Click on "View Blog" | Display list of blog posts | Pass |
| 3 | Create a new blog post | Enter post details and submit | New blog post created | Pass |
| 4 | Edit existing blog post | Select post, edit, and save changes | Changes reflected in the edited post | Pass |
| 5 | Delete a blog post | Select post, confirm deletion | Post deleted from the system | Pass |
| 6 | Comment on a blog post | Open post, add comment, and submit | Comment appears in the post | Pass |
| 7 | Manage user profiles | Edit profile details and save changes | User profile updated successfully | Pass |
| 8 | Logout from the system | Successfully logged out | Successfully logged out | Pass |
| 9 | Love React | Clicked Loved Icon | Loved and Counted | Pass |

Table 2: Test Case For Unit Testing

### 4.2.2. Test Cases for System Testing

System testing involves assessing the entire system's functionality and performance in a simulated environment.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N | Test Description | Input | Output | Status |
| 1 | Register a new user | Enter valid registration details | New user registered | Pass |
| 2 | Verify user authentication | Login with registered credentials | Successful authentication | Pass |
| 3 | Post and view multimedia content | Upload images in a post | Multimedia content visible | Pass |
| 4 | Test comment section functionality | Add and view comments on a post | Comments displayed | Pass |
| 5 | Test system performance under load | Simulate concurrent user interactions | System handles load well | Pass |

Table 3: Test Case For System Testing

# Chapter: 5 Conclusion and Future Recommendations

## 5.1. Outcome:

In reviewing the implemented blogging system, it becomes evident that the project has successfully achieved its primary objectives. The user-friendly interface facilitates seamless user interaction, allowing users to log in, create, and engage with blog posts effortlessly. The implementation of features such as commenting and reacting to posts enhances user engagement, contributing to a vibrant online community.

## 5.2 Conclusion:

In conclusion, the development and deployment of the blogging system have been fruitful. The system provides an intuitive platform for users to share their thoughts and connect with others through blog posts and comments. The responsive design and feature-rich environment contribute to a positive user experience. As the project moves forward, continuous improvements and enhancements can be made based on user feedback and emerging technologies to ensure the system remains relevant and valuable in the future.

## 5.3. Future Recommendations

1. Make it easier for users to connect by adding real-time notifications and a messaging system.
2. Use smart technology to keep content safe and respectful with an automatic moderation system.
3. Create a mobile app so users can blog on it.
4. Learn more about what users like and don't like by adding tools that measure post engagement and user behavior.
5. Add more ways for users to connect, like forums and events, and make it more fun with game-like features.
6. Make sure everyone can use the system by adding features for people with different abilities and languages.
7. Keep everything safe and secure by checking for problems and fixing them regularly.
8. Let users share their blog posts on social media and other platforms.
9. Ask users what they think regularly so you can make the system even better.
10. Plan for the future by making sure the system can grow and handle more users.

# Appendices

<?php

session\_start();

ini\_set('display\_errors', 1);

error\_reporting(E\_ALL);

// ------check if login or not---------

if (!isset($\_SESSION["user\_id"])) {

header("Location: login.php");

exit;

}

include 'connect.php';

include 'comments.php';

// ------love count update---------

if (isset($\_GET['love'])) {

$lovePostID = $\_GET['love'];

$sqlUpdateLove = "UPDATE posts SET love\_count = love\_count + 1 WHERE post\_id = ?";

$stmtUpdateLove = $conn->prepare($sqlUpdateLove);

$stmtUpdateLove->bind\_param("i", $lovePostID);

if ($stmtUpdateLove->execute()) {

header("Location: {$\_SERVER['PHP\_SELF']}");

exit();

} else {

echo "Error updating love count: " . $stmtUpdateLove->error;

}

$stmtUpdateLove->close();

}

if (isset($\_GET['query'])) {

$query = $\_GET['query'];

$sql = "SELECT posts.\*, users.username FROM posts JOIN users ON posts.user\_id = users.user\_id

WHERE content LIKE ? OR username LIKE ? ORDER BY posts.post\_date DESC";

$stmt = $conn->prepare($sql);

$searchKeyword = "%" . $query . "%";

$stmt->bind\_param("ss", $searchKeyword, $searchKeyword);

$stmt->execute();

$result = $stmt->get\_result();

if ($result->num\_rows > 0) {

$uniquePostIDs = [];

while ($row = $result->fetch\_assoc()) {

$postID = $row['post\_id'];

if (in\_array($postID, $uniquePostIDs)) {

continue;

}

$uniquePostIDs[] = $postID;

$username = $row['username'];

$content = $row['content'];

$postDate = $row['post\_date'];

$loveCount = $row['love\_count'];

echo "<a href='post\_details.php?post\_id=$postID' class='post-link'>";

echo "<section class='post'>";

echo "<div class='post-header'>";

echo "<h2 class='username'>$username</h2>";

echo "</div>";

echo "<p class='meta'>Posted on $postDate</p>";

echo "<div class='post-content'>";

echo "<p id='limitedText'>$content</p>";

echo "</div>";

if (!empty($row['image\_path'])) {

$imagePath = htmlspecialchars($row['image\_path']);

$timestamp = time();

echo "<div style='width: 100%; height: 400px; overflow: hidden;'>";

echo "<img src='{$imagePath}?t={$timestamp}' alt='Post Image' class='fill-container'>";

echo "</div>";

}

echo "<div class='love-section' style='display: flex; align-items: center;'>";

echo "<a href='javascript:void(0);' onclick='updateLoveCount($postID)' class='love-button'><i class='fa-regular fa-heart'></i></a>";

echo "<span class='love-count' id='love-count-$postID'>$loveCount</span>";

echo "</div>";

echo "</div>";

echo "</section>";

echo "</a>";

}

} else {

echo "No results found for '$query'";

}

$stmt->close();

} else {

$sql = "SELECT posts.\*, users.username FROM posts JOIN users ON posts.user\_id = users.user\_id ORDER BY posts.post\_date DESC";

$result = $conn->query($sql);

$uniquePostIDs = [];

if ($result === false) {

echo "Error: " . $conn->error;

} elseif ($result->num\_rows > 0) {

while ($row = $result->fetch\_assoc()) {

$postID = $row['post\_id'];

if (in\_array($postID, $uniquePostIDs)) {

continue;

}

$uniquePostIDs[] = $postID;

$username = $row['username'];

$content = $row['content'];

$postDate = $row['post\_date'];

$loveCount = $row['love\_count'];

echo "<a href='post\_details.php?post\_id=$postID' class='post-link'>";

echo "<section class='post'>";

echo "<div class='post-header'>";

echo "<h2 class='username'>$username</h2>";

echo "</div>";

echo "<p class='meta'>Posted on $postDate</p>";

echo "<div class='post-content'>";

echo "<p class='limitedText'>$content</p>";

if (!empty($row['image\_path'])) {

$imagePath = htmlspecialchars($row['image\_path']);

$timestamp = time();

echo "<div style='width: 100%; height: 400px; overflow: hidden;'>";

echo "<img src='{$imagePath}?t={$timestamp}' alt='Post Image' class='fill-container'>";

echo "</div>";

}

echo "<div class='love-section' style='display: flex; align-items: center;'>";

echo "<a href='javascript:void(0);' onclick='updateLoveCount($postID)' class='love-button'><i class='fa-regular fa-heart'></i></a>";

echo "<span class='love-count' id='love-count-$postID'>$loveCount</span>";

echo "</div>";

echo "</div>";

echo "</section>";

echo "</a>";

}

} else {

echo "No posts found.";

}

}

$conn->close();

?>

**Index.php**

<?php

$host = "localhost";

$username = "root";

$password = "0092";

$database = "canteenmate";

// Create a database connection

$conn = new mysqli($host, $username, $password, $database);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

**Connect.php**

<?php

ini\_set('display\_errors', 1);

error\_reporting(E\_ALL);

include 'connect.php';

function getComments($postID)

{

global $conn;

$sql = "SELECT comments.\*, users.username AS user\_name FROM comments JOIN users ON comments.user\_id = users.user\_id WHERE post\_id = ? ORDER BY comment\_date ASC";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("i", $postID);

$stmt->execute();

$result = $stmt->get\_result();

return $result->fetch\_all(MYSQLI\_ASSOC);

}

function addComment($postID, $userID, $commentText)

{

global $conn;

$sql = "INSERT INTO comments (post\_id, user\_id, comment\_text) VALUES (?, ?, ?)";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("iis", $postID, $userID, $commentText);

return $stmt->execute();

}

if (isset($\_POST['submitComment'])) {

$postID = $\_POST['postID'];

$commentText = trim($\_POST['comment']);

if (!empty($commentText)) {

session\_start();

if (isset($\_SESSION['user\_id'])) {

$userID = $\_SESSION['user\_id'];

} else {

header("Location: login.php");

exit;

}

$success = addComment($postID, $userID, $commentText);

if ($success) {

header("Location: post\_details.php?post\_id=$postID");

exit;

} else {

echo "<p>Failed to add comment. Please try again later.</p>";

}

} else {

echo "<p>Please enter a comment.</p>";

}

}

**Comments.php**

<?php

ini\_set('display\_errors', 1);

error\_reporting(E\_ALL);

session\_start();

include 'connect.php';

if (!isset($\_SESSION["user\_id"])) {

header("Location: login.php");

exit;

}

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$content = trim($\_POST['content']);

if (empty($content)) {

die("Please enter content for your post.");

}

$target\_dir = "uploads/";

if (!file\_exists($target\_dir)) {

mkdir($target\_dir, 0777, true);

}

$target\_file = $target\_dir . basename($\_FILES["image"]["name"]);

$uploadOk = 1;

$imageFileType = strtolower(pathinfo($target\_file, PATHINFO\_EXTENSION));

$imageInfo = @getimagesize($\_FILES["image"]["tmp\_name"]);

if ($imageInfo === false) {

die("File is not a valid image.");

}

if ($\_FILES["image"]["size"] > 500000) {

die("Sorry, your file is too large.");

}

$allowedFormats = ["jpg", "jpeg", "png", "gif"];

if (!in\_array($imageFileType, $allowedFormats)) {

die("Sorry, only JPG, JPEG, PNG & GIF files are allowed.");

}

if ($uploadOk == 0) {

die("Sorry, your file was not uploaded.");

} else {

if (move\_uploaded\_file($\_FILES["image"]["tmp\_name"], $target\_file)) {

echo "The file " . htmlspecialchars(basename($\_FILES["image"]["name"])) . " has been uploaded.";

$sql = "INSERT INTO posts (user\_id, content, image\_path, post\_date) VALUES (?, ?, ?, NOW())";

$stmt = $conn->prepare($sql);

$stmt->execute([$\_SESSION["user\_id"], $content, $target\_file]);

if ($stmt->affected\_rows > 0) {

header('Location: index.php');

exit();

} else {

error\_log("Error saving post: " . $stmt->error);

die("Error saving post. Please try again later.");

}

} else {

die("Sorry, there was an error uploading your file.<br>Error details: " . $\_FILES["image"]["error"]);

}

}

}

$conn->close();

?>

**Create\_Post.php**

<?php

session\_start();

include 'connect.php';

if (isset($\_SESSION["user\_id"])) {

header("Location: profile.php");

exit;

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$username = $\_POST["username"];

$password = $\_POST["password"];

$sql = "SELECT \* FROM users WHERE username = ?";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("s", $username);

$stmt->execute();

$result = $stmt->get\_result();

if ($result->num\_rows == 1) {

$row = $result->fetch\_assoc();

if (password\_verify($password, $row["password"])) {

$\_SESSION["user\_id"] = $row["user\_id"];

header("Location: profile.php");

exit;

} else {

echo "Invalid password.";

}

} else {

echo "User not found.";

}

$stmt->close();

}

$conn->close();

?>

**Login.php**

<?php

error\_reporting(E\_ALL);

ini\_set('display\_errors', 1);

session\_start();

include 'connect.php';

if (!isset($\_SESSION["user\_id"])) {

exit("User not logged in.");

}

$user\_id = $\_SESSION["user\_id"];

$sql = "SELECT \* FROM users WHERE user\_id = ?";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("i", $user\_id);

if (!$stmt->execute()) {

echo "SQL Error: " . $stmt->error;

exit;

}

$result = $stmt->get\_result();

if ($result->num\_rows == 1) {

$row = $result->fetch\_assoc();

$username = $row["username"];

$email = $row["email"];

$profile\_image = $row["profile\_image"];

echo '<div class="profile-details">';

echo '<h3>Username: ' . $username . '</h3>';

echo '<p>Email: ' . $email . '</p>';

echo '<div class="form-group">';

echo '<div class="button-container">';

echo '<a href="editprofile.html"><button class="button">Edit</button></a>';

echo '<a href="logout.php"><button class="button">Logout</button></a>';

echo '</div>';

echo '</div>';

echo '</div>';

echo '<div class="profile-image-container">';

if (isset($profile\_image) && !empty($profile\_image) && file\_exists($profile\_image)) {

echo '<img src="' . $profile\_image . '" alt="Profile Image" class="profile-image" onclick="document.getElementById(\'image\').click();">';

} else {

echo '<img src="uploads/default\_profile.jpg" alt="Default Profile Image" class="profile-image" onclick="document.getElementById(\'image\').click();">';

}

echo '<form action="profile.php" method="post" enctype="multipart/form-data" id="imageForm">';

echo '<input type="file" name="image" id="image" accept="image/\*" style="display: none;" onchange="readURL(this);">';

echo '<input type="submit" value="Upload Image" name="submitImage">';

echo '</form>';

echo '<button type="button" onclick="document.getElementById(\'image\').click();">Upload Image</button>';

echo '</div>';

} else {

exit("User not found.");

}

$stmt->close();

if (isset($\_FILES['image'])) {

$target\_dir = "uploads/";

$target\_file = $target\_dir . basename($\_FILES["image"]["name"]);

$uploadOk = 1;

$imageFileType = strtolower(pathinfo($target\_file, PATHINFO\_EXTENSION));

if (file\_exists($target\_file)) {

$filename = pathinfo($target\_file, PATHINFO\_FILENAME);

$extension = pathinfo($target\_file, PATHINFO\_EXTENSION);

$timestamp = time();

$target\_file = $target\_dir . $filename . '\_' . $timestamp . '.' . $extension;

}

if ($uploadOk == 0) {

echo "Sorry, your file was not uploaded.";

} else {

if (move\_uploaded\_file($\_FILES["image"]["tmp\_name"], $target\_file)) {

$updateSql = "UPDATE users SET profile\_image = ? WHERE user\_id = ?";

$updateStmt = $conn->prepare($updateSql);

$updateStmt->bind\_param("si", $target\_file, $user\_id);

if ($updateStmt->execute()) {

header("Location: profile.php");

exit();

} else {

echo "Error updating profile image.";

}

}

}

}

$conn->close();

?>

**Profile.php**

<?php

session\_start();

include 'connect.php';

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST["signup"])) {

$username = $\_POST["username"];

$email = $\_POST["email"];

$password = $\_POST["password"];

$confirmPassword = $\_POST["confirm\_password"];

if ($password !== $confirmPassword) {

echo "Passwords do not match.";

exit;

}

$hashed\_password = password\_hash($password, PASSWORD\_DEFAULT);

$sql = "INSERT INTO users (username, email, password) VALUES (?, ?, ?)";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("sss", $username, $email, $hashed\_password);

$stmt->execute();

if ($stmt->affected\_rows == 1) {

header("Location: login.php");

exit;

} else {

echo "Error occurred while registering the user.";

}

$stmt->close();

}

$conn->close();

?>

**Signup.php**

# Bibliography

1. [**https://www.w3schools.com/html**](https://www.w3schools.com/html/) - for Html, CSS, Php and Js.
2. [**https://www.drawio.com**](https://www.drawio.com/) and [**https://www.lucidchart.com**](https://www.lucidchart.com/pages/) - for Diagram.
3. [**https://openai.com/chatgpt**](https://openai.com/chatgpt) - for help and error handling.
4. [**https://www.sqltutorial.org**](https://www.sqltutorial.org/) for sql
5. [**https://www.youtube.com/**](https://www.youtube.com/) - for error handling