Project Report on Database Design for a Social Networking Platform

Submitted by

RITABRATA JOSH 12022002016034
SOMNATH DUTTA 12022002016063

Under the supervision and guidance of

Prof.(Dr.) Deepsubhra Guha Roy

ASSOCIATE PROFESSOR OF CSE(AIML)

In the partial fulfilment of requirements for the award of Degree in ***Bachelors of Computer Science and Engineering(Artificial Intelligence and Machine Learning)***

Batch 2022-2026

Submitted to the

DEPARTMENT OF CSE(AIML)

INSTITUTE OF ENGINEERING AND MANAGEMENT

Table of Content

List of Figure

ABSTRACT

This document outlines the design of a social media platform database, focusing on core features that enable user interaction, content sharing, and community building. The platform facilitates the creation and management of user profiles, which include personal details, bio, and profile pictures. Users can create posts with textual and multimedia content, engage in discussions through comments, and express their appreciation through likes. Additionally, a friendship system allows users to connect with each other, supporting bi-directional relationships and friend request statuses.

The database schema includes primary tables for users, posts, comments, friendships, and likes, each capturing essential data attributes and establishing relational connections between tables. These relationships enable complex social interactions, such as tracking a user's social network, viewing post comments, and managing friendships. Through this structured data model, the platform enables scalable, efficient, and secure user engagement in a digital social space. This design forms the foundation for a fully functional social network, allowing seamless interaction among users and supporting the key features of a modern social media platform.

In conclusion, this social media platform database design establishes a comprehensive framework for user engagement and social interaction. By implementing a well-structured relational schema, the platform effectively supports essential functionalities, including user profiles, content sharing, commenting, liking, and friend connections. The schema's design balances data organization and scalability, enabling efficient data retrieval and a seamless user experience even as the platform grows.

CHAPTER :1

INTRODUCTION

1.1Overview:

Social media platforms have become integral to modern communication, enabling users to connect, share, and interact in a digital environment. These platforms foster a sense of community and engagement, allowing individuals to share personal updates, connect with friends and family, and express themselves through posts, comments, and reactions. At the core of any social media platform is a robust database structure that efficiently handles a large volume of user data, interactions, and media content, ensuring smooth and responsive experiences for users.

This platform is designed with key features that cater to a broad range of social interactions. Users create personal profiles to represent themselves and connect with others, sharing content through posts that can include text, images, or multimedia. Interactions are facilitated through comments and likes, allowing users to engage with one another's content and form connections. Additionally, a friendship system supports user networks by enabling friend requests, managing connections, and an environment where users can exp define relationship statuses.

This design document outlines the foundational database schema for such a platform, aiming to balance user engagement with system performance and scalability. Through efficient data organization and well-defined relationships between data entities, this structure enables a seamless and interactive social experience

1.2 MOTIVATION FOR WORK:

The MERN stack social media app report could have several motivations, depending on the goals of the person or organization creating it. Some possible motivations could include:

1.**Technical evaluation**: A report could be created to evaluate the technical aspects of the MERN stack social media app, including its architecture, design, functionality, performance, and scalability. This information could be useful for developers who want to assess the quality of the app's codebase and identify areas for improvement.

2.**User experience analysis:** A report could also be created to analyze the user experience of the MERN stack social media app, including how easy it is to use, how engaging its features are, and how well it meets the needs of its target audience. This information could be useful for designers who want to improve the app's usability and enhance the user experience.

3.**Business assessment:** A report could be created to assess the business potential of the Business assessment MERN stack social media app, including its market demand, revenue potential, and growth opportunities. This information could be useful for entrepreneurs who want to launch a social media app and investors who want to evaluate the app's viability.

4.Comparative analysis: A report could be created to compare the MERN stack social media app with other social media apps in the market, including their features, performance, user base, and revenue. This information could be useful for marketers who want to identify the app's competitive advantages and weaknesses and develop a marketing strategy to attract users.

1.3 PROBLEM STATEMENT:

As social media apps continue to gain popularity, it is important to understand how they are impacting user behavior and the broader social landscape. There is a need to evaluate the performance, user experience, and business potential of social media apps, such as the MERN stack social media app, in order to identify areas for improvement and growth. Additionally, there is a need to compare the MERN stack social media app with other social media apps in the market to understand its competitive advantages and weaknesses. By addressing these issues, we can gain insights into how social media apps are shaping the way we communicate and interact with one another, and develop strategies to optimize their impact.

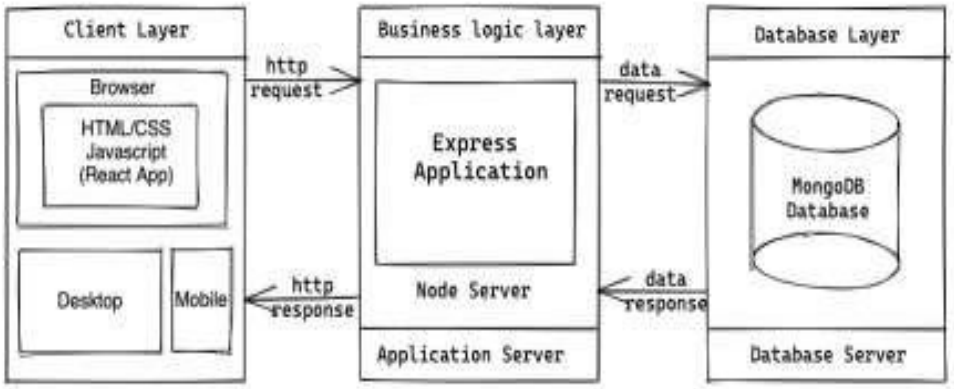
1.4 Scope of the Project:

The main objective of the Project on Social Networking Site is to manage the details of Users, Friends, Posts, Shares, and Photos. It manages all the information about Users, Videos, Photos, and Users. The project is totally built at the administrative end and thus only the administrator is guaranteed the access.

1.5 Aim of Project:

It may help collect perfect management in detail. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the management of the past year perfectly and vividly. It also helps in current work relative to Social Networking Site. It will also reduce the cost of collecting the management & collection procedure will go on smoothly.

CHAPTER:2
PROJECT DESCRIPTION



2.1**BACKEND (SERVER-SIDE):-**

Building websites and web apps has always been done using server-side rendering, also referred to as back-end web development. When we access a page, we send a request for data to the server, which processes it and sends back a response to the browser. All the activities required to build an HTML page that the web browser can understand are carried out on the remote server that houses the website or web application when a website renders server-side. This entails processing any required logic as well as information queries from databases for that web application. While it waits for the distant server to finish processing the request and provide the response, the web browser on the other end sits idle. When a response is sent, web browsers interpret it and show the material on the screen.

2.2**FRONTEND (CLIENT-SIDE):-**

Client-side rendering, often known as front-end development, is a new style of site rendering that is employed in contemporary apps. JavaScript, which is now the de facto standard web language, is used to render the content on your computer as opposed to a distant web server in clientside rendering. In actuality, this indicates that a browser is responsible for generating the HTML output of the web application and that a server is only needed to provide the raw web application. Additionally, it shows that a piece of the presentation logic—the reasoning used to create a web page and display it to the user on the screen—is handled on the client-side. With the introduction of JavaScript libraries like Angular, React, and Vue, client-side rendering became more common.

CHAPTER:3

SYSTEM ANALYSIS

3.1Hardware and Software Requirement

3.1.1 Hardware Configuration

1. Pentium IV Processor
2. 512 MB RAM
3. 40GB HDD
4. 1024 * 768 Resolution Color Monitor

3.1.2 Software Configuration

1. OS: Windows XP,Linux.

2.PHP Triad (PHP5.6, MySQL, Apache, and PhpMyAdmin)

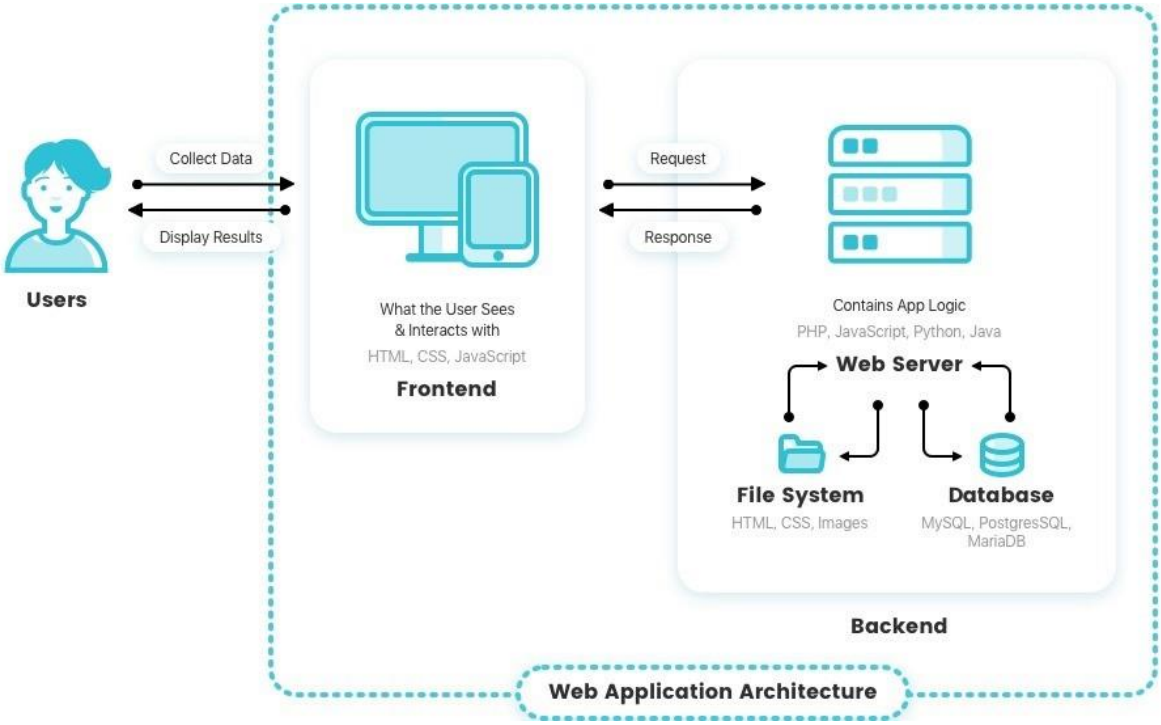
3.Programming Languages: HTML, CSS, Java Script

3.1.3 Server Requirements

1. PHP >= 8.1
2. BCMath PHP Extension
3. Ctype PHP Extension
4. cURL PHP Extension
5. DOM PHP Extension
6. Fileinfo PHP Extension
7. JSON PHP Extension
8. Mbstring PHP Extension
9. OpenSSL PHP Extension
10. PCRE PHP Extension
11. PDO PHP Extension
12. Tokenizer PHP Extension

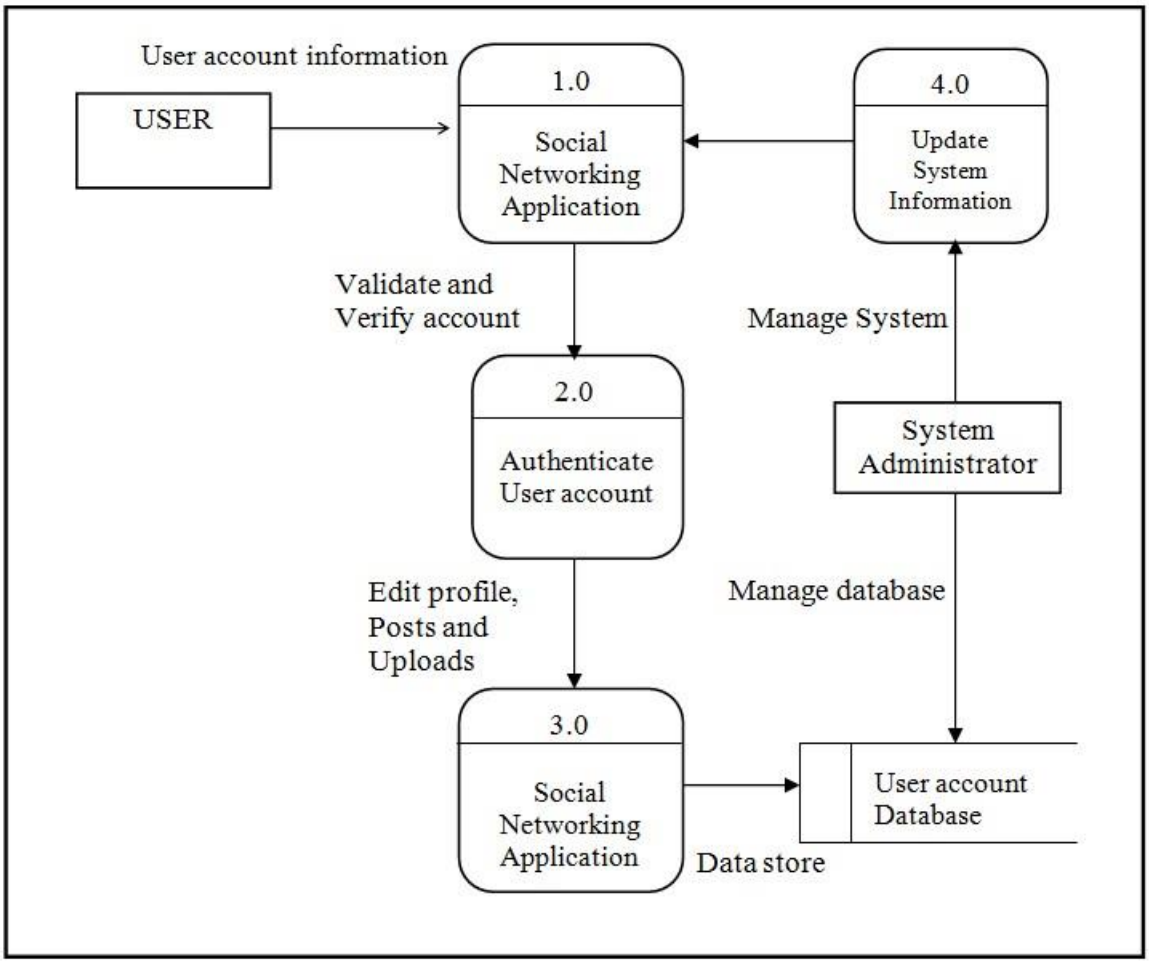
13. XML PHP Extension
14. Intl PHP Extension

3.2.1 Architecture Diagram:



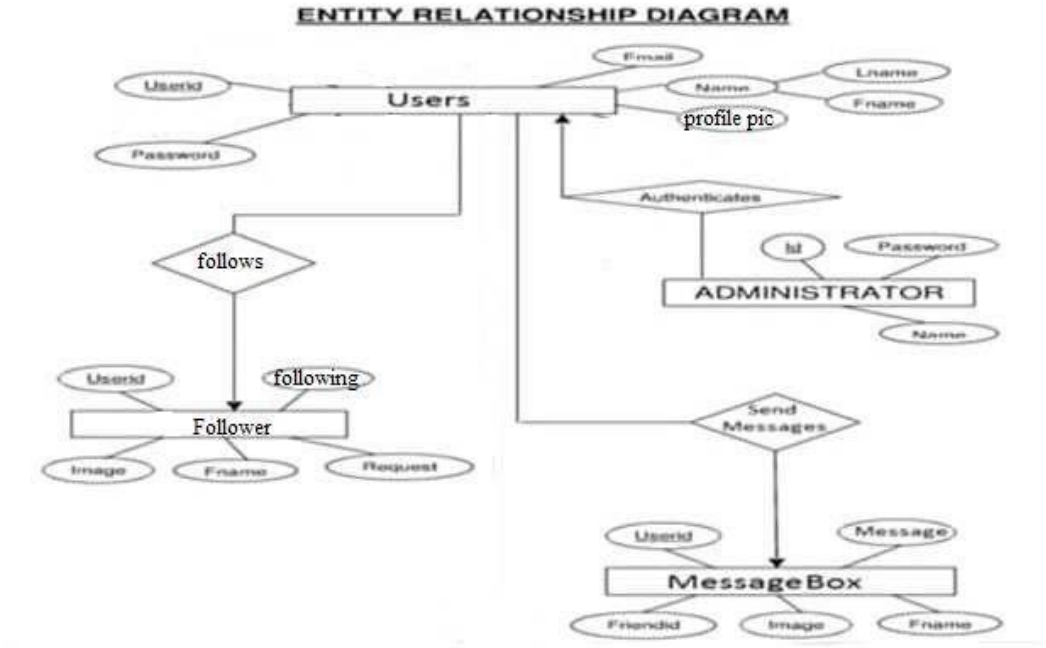
1. A user enters URL into browser.
2. Browser generates an HTTP request to the appropriate web serve.
3. The web server interprets it and forwards it to the corresponding JSP.
4. The JSP processes the request, generates the output and sends it back to theweb server.
5. The web server sends the response back to the client. The browser thendisplays it on the screen.

3.2.2 Data flow Diagram:



A data-flow diagram (DFD) is a graphical representation of the "flow" of data through information system. DFDs can also be used for the visualization of data processing (structured design). On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process. A DFD provides no information about the timing or ordering of processes, or about whether the flow processes will operate in sequence or in parallel. It is therefore quite different from a flowchart, which shows of control through an algorithm, allowing a reader to determine what operations will be performed, in what order, and under what circumstances, but not what kinds of data will be input to and output from the system, nor where the data will come from and go to, nor where the data will be stored (all of which are shown on a DFD).

3.2.3 ER-Diagram:



In software engineering, an Entity-Relationship Model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.

CHAPTER: 4 THEORETICAL ANALYSIS

4.1 Introduction to Tools used in Project

HTML :

HTML means Hypertext mark-up language .HTML is method of describing the format of document, which allows them to be viewed on the computer screen . HTML documents are dis-played by web browser, programs which can navigate across networks and display a wide variety of types of information.HTML page can be developed to be a simple text or to be complex multimedia containing sound, moving ,images, virtual reality, and java applets. The global publishing format is HTML. It allows authors to use not only text but also format that text with heading, lists, and tables .Readers can access the pages of information from anywhere in the world at a click of mouse button.HTML pages can also be used for entering the data as a front end for commercial transactions.

For Example:

This is a sample code of html which is used to make a login form .

```
<!DOCTYPE html>
<html>
<head>
<title>sample </title>
</head>
<body>

<input type="text" placeholder="Email" name="email" id="type">

<input type="password"placeholder="*****" name="password" id="type">

<td align="center"><input type="submit" value="Login" name="submit"
id="btn"> </body>
```

```
</html>
```

4.1.2 Introduction of CSS

Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.

Advantages of CSS

1. Greater consistency in design.
2. Ease of presenting different styles to different viewers.

Friendly Environment

Creating a form, adding controls to form and writing code behind the form are all managed within a friendly Environment.

For Example:

Inside the html message body if we insert the code of CSS inside the head with <style> tag like :

```
.message-body{
    margin-top: 20px;
    width: 30%;
    padding: 5px;
}
.message-window{
    padding: 20px;
    border: 1px solid grey;
    border-radius: 20px;
    width: 100%;
    height: 80%;
```

```
}
li{
    display: inline-block;
    vertical-align: middle;
}
.message-account-profpic-online{
    height: 70px;
    margin-right: 10px;
    border: 2px solid green;
    border-radius: 360px;
}
.message-account-profpic-offline{
    height: 70px;
    margin-right: 10px;
    border: 2px solid gray;
    border-radius: 360px;
}
.message-window-head{
    padding: 10px;
    width: 100%;
    border-bottom: 3px solid #13239ac0;
    margin-bottom: 10px;
}
.message-window-message-box{
    overflow-y: auto;
```

```
    height: 80%;
    max-height: 80%;
}
.message-window-message-display-box{
    padding-top: 5px;
    padding-bottom: 5px;
}
.message-window-message-display-box-message{
    border-left: 2px solid #13239ac0;
    padding-left: 10px;
}
.message-window-input-message{
    position:unset;
    top: 1%;
}
.message-window-input-message input{
    text-align: left;
    width: 90%;
    height:30px;
    margin-left: 5%;
    margin-right: 5%;
    padding: 20px;
    border-radius: 360px;
}
.theme-button{
```

```
    height: 35px;
    width: 35px;
    border: dashed #ffffff3d 0.5px;
    background-color: #2058d100;
    border-radius: 360px;
    padding: 5px;
    margin-top: 22px;
  }
@media(max-width: 720px) {
    .message-body{
        width: 90%;
    }
    .message-window-message-box{
        height: 75%;
        max-height: 75%;
        font-size: 15px;
    }
            .message-window-input-message{
        margin-top: 5px;
    }
}
.menu-icon{
    height: 15px;
    width: 15px;
    vertical-align: middle;
```

```
        margin-right: 5px;
        margin-left: 5px;
    }
```

4.1.3 JAVASCRIPT :

Java Script is a fairly simple language ,which is only suitable for fairly simple tasks. The language is best suited for the task, behind finding the java script is to find the language which could be used to provide client side browser application but which was not as complicated as Java. Java Script is Netscape cross platform object oriented scripting language. Core Java Script contains a core set of objects such as array, date and Math and a core set of language elements such as operators, control structure and statements. It is mainly used here for validation. JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is compliments,
to integrate with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and unlike images, applets are dynamic and interactive. Applets can be used to create animation.

For Example:

This is a sample code of javascript which is used to make a login form .

```
// Check button in mobile view
const menuBTN = document.querySelector('.menu-btn');
const menuItems = document.querySelector('.menu-items');

function toggleBtn() {
   menuBTN.classList.toggle("change");
   menuItems.classList.toggle("active");
}

menuBTN.addEventListener('click', toggleBtn);

// JavaScript code for the lightbox feature
      function openLightbox(imageSrc) {
         var lightboxOverlay = document.getElementById('lightbox-overlay');
         var lightboxImage = document.getElementById('lightbox-image');

         lightboxImage.src = imageSrc;
         lightboxOverlay.style.display = 'block';
```

```
        }

        function closeLightbox() {
            var lightboxOverlay = document.getElementById('lightbox-overlay');
            lightboxOverlay.style.display = 'none';
        }

        document.addEventListener("DOMContentLoaded", function() {
            var images = document.querySelectorAll('.feed-post-display-box-image img');

            images.forEach(function(image) {
                image.addEventListener('click', function() {
                    openLightbox(this.src);
                });
            });
        });

//Dark theme code


function changeTheme() {
    const theme = document.getElementById("theme");
    const themeIcon = document.getElementById("theme-icon");
    //console.log('theme:',theme.getAttribute("href").includes("darktheme_css"));
    if (theme.getAttribute("href").includes("darktheme_css")) {
        theme.setAttribute("href", "style/lighttheme_css/light_feed.css");
        themeIcon.setAttribute("src", "img/dark_img/MoonIcon.png");
    } else {
        theme.setAttribute("href", "style/darktheme_css/dark_feed.css");
        themeIcon.setAttribute("src", "img/dark_img/SunIcon.png");
    }
}

function changeAccountTheme() {
    const theme = document.getElementById("theme");
    const themeIcon = document.getElementById("theme-icon");
    //console.log('theme:',theme.getAttribute("href").includes("darktheme_css"));
    if (theme.getAttribute("href").includes("darktheme_css")) {
        theme.setAttribute("href", "style/lighttheme_css/light_account.css");
        themeIcon.setAttribute("src", "img/dark_img/MoonIcon.png");
```

```
    } else {
        theme.setAttribute("href", "style/darktheme_css/dark_account.css");
        themeIcon.setAttribute("src", "img/dark_img/SunIcon.png");
    }
}

function changeIndexTheme() {
    const theme = document.getElementById("theme");
    const themeIcon = document.getElementById("theme-icon");
    //console.log('theme:',theme.getAttribute("href").includes("darktheme_css"));
    if (theme.getAttribute("href").includes("darktheme_css")) {
        theme.setAttribute("href", "style/lighttheme_css/light_style.css");
        themeIcon.setAttribute("src", "img/dark_img/MoonIcon.png");
    } else {
        theme.setAttribute("href", "style/darktheme_css/dark_style.css");
        themeIcon.setAttribute("src", "img/dark_img/SunIcon.png");
    }
}


function changeMessageTheme() {
    const theme = document.getElementById("theme");
    const themeIcon = document.getElementById("theme-icon");
    //console.log('theme:',theme.getAttribute("href").includes("darktheme_css"));
    if (theme.getAttribute("href").includes("darktheme_css")) {
        theme.setAttribute("href", "style/lighttheme_css/light_message.css");
        themeIcon.setAttribute("src", "img/dark_img/MoonIcon.png");
    } else {
        theme.setAttribute("href", "style/darktheme_css/dark_message.css");
        themeIcon.setAttribute("src", "img/dark_img/SunIcon.png");
    }
}
```

4.1.4 MERN Stack :-

MERN stack is a framework used for creating websites (web app development). MongoDB, ExpressJS, ReactJS, and NodeJS make up its functional components. The specific role of each of these elements while creating a web application are listed below:

- **MongoDB**: The application data is stored in this document- oriented, No-SQL database.
- **NodeJS**: This is the JavaScript runtime environment that is used to run the JavaScript code on the machine itself, instead of a browser.
- **ExpressJS**: It is a framework that sits atop NodeJS and is used to create a website's backend using NodeJS functions and structures. NodeJS was created to run JavaScript on computers, not to create websites, so ExpressJS was created to fill that gap.
- **ReactJS**: It is a library that Facebook built. It is used to build the UI elements that go into a single page web application's user interface. The user interacts with the ReactJS UI components in the front-end of the application, which is situated in the browser. The backend of this application, which is located on a server, is served by ExpressJS, which is built upon NodeJS.

A request to change data is sent to the Express server, which is built on NodeJS, after any interaction. When necessary, Express fetches information from the MongoDB database and sends it to the application's front end, where it is shown to the user.

A single-page web application (SPA) or website interacts with the user and dynamically updates the current web page rewriting the new or modified data from the web server, in contrast to the traditional practices of a web browser loading entirely new pages. The webpage will transition more quickly to boost the appearance of a native app. As opposed to the traditional way, all essential HTML, JavaScript, and CSS code is either fetched by the browser with a single page load or the required resources are dynamically updated and loaded to the webpage as needed, generally in reaction to user activities. A SPA never refreshes the page. Even using the tools mentioned above, it is difficult to build a high-performing app that is fast, responsive, user-friendly by design and secure, maintaining user integrity and security.

4.1.5 PHP TRIAD:

PHP Triad installs a complete working PHP/MySQL server environment on Windows platforms (9x/ NT). Installs PHP, MySQL, Apache, and PHPMyAdmin. PHP is a scripting language originally designed for producing dynamic web pages. It has evolved to include a command line interface capability and can be used in standalone graphical applications. While PHP was originally created by RasmusLerdorf in 1995, the main implementation of PHP is now produced by ThePHP Group and serves as the de facto standard for PHP as there is no formal specification. PHP is free software released under

the PHP License, however it is incompatible with the GNU General Public License.(GPL), due to restrictions on the usage of the term PHP. It is a widely-used general-purpose scripting language that is especially suited for web development and can be embedded into HTML. It generally runs on a web server, taking PHP code as its input and creating web pages as output. It can be deployed on most web servers.

4.1.6 PHP Syntax:

HTML, the PHP code is enclosed within <? Php ?> Tags.

For example:

```
<html>
<head>
<title>php sample</title>
<body
<h2>Hello<
/h2> <?php
echo
“hello”; ?>
</body>
</html>
```

In the above example PHP code is embedded within HTML. In this way the PHP and HTML coding is combined on the same page.

4.1.7 Working with PHP:

When a client requests web page containing PHP code from the server, then the requested PHP pages are passed under PHP environment and interaction with database is made if required. After server side processing, the resulting HTML pages are passed to client and displayed on the browser. In this way the working of PHP is complete.

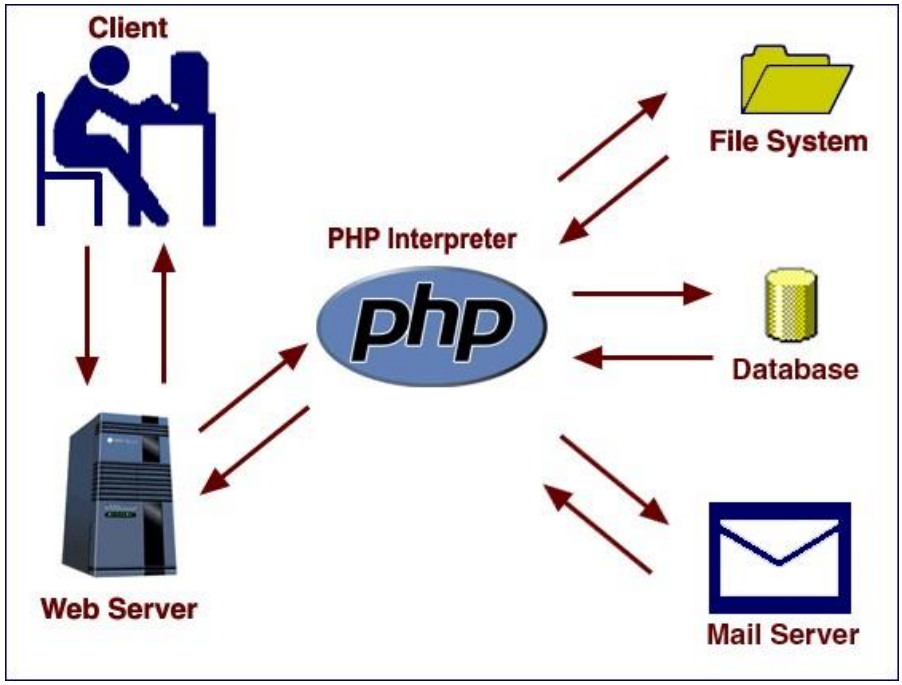
Fig 4.1. Working In PHP

4.1.8 Connecting PHP Application with MySQL Database ▪

```
<?php
// database connection for Social_Media_App
$host = "localhost";
$user = "root";
$pswd = "";
$db = " Social_Media_App ";
// create connect to db
$connection = mysqli_connect($host, $user, $pswd, $db);
?>
```

4.1.9 Introduction to MySQL:

MySQL is a Relational Database Management System (RDBMS) that runs as a server providing multi-user access to a number of databases. MySQL is pronounced (“My S-Q-L”)



MySQL development project has made its source available under the terms of General Public License. MySQL is owned and sponsored by a single for profit firm, the Swedish company MySQL AB, now owned by Sun Microsystem, a subsidiary of Oracle Corporation.
MySQL works on many different system platforms including AIX, BSD i, FreeBSD, HP-UX, i5/OS, Linux, Mac OS X, Net BSD, Novell NetWare, Open BSD, Open Solaris, e com Station , OS/2 Wrap, QNX, IRIX, Solaris, Symbian, SunOS, SCO Open Server, SCO Unix Ware, Sanos, Tru64 and Microsoft Windows. A port of MySQL to Open VMS also exits. All major programming languages with language-specific APIs include Libraries for accessing MySQL database. In addition, an ODBC interface called MYODBC allows additional programming languages that supports the ODBC interface to communicate with a MySQL database, such as ASP or ColdFusion. MySQL server and official libraries are mostly implemented in ANSI C/ ANCI C++.

4.1.10 Introduction to APACHE SERVER

In this project apache server is user to parse and execute PHP pages, before deploying websites on the server, the website should be tested at the developer's side to get a feel of how the website will work on actual server. Therefore apache server is like a local server on the developer side, apache server should be informed about the environment on which it should work. In our project apache server is configured to work with PHP, in this way all the PHP pages are parsed and executed by the server.

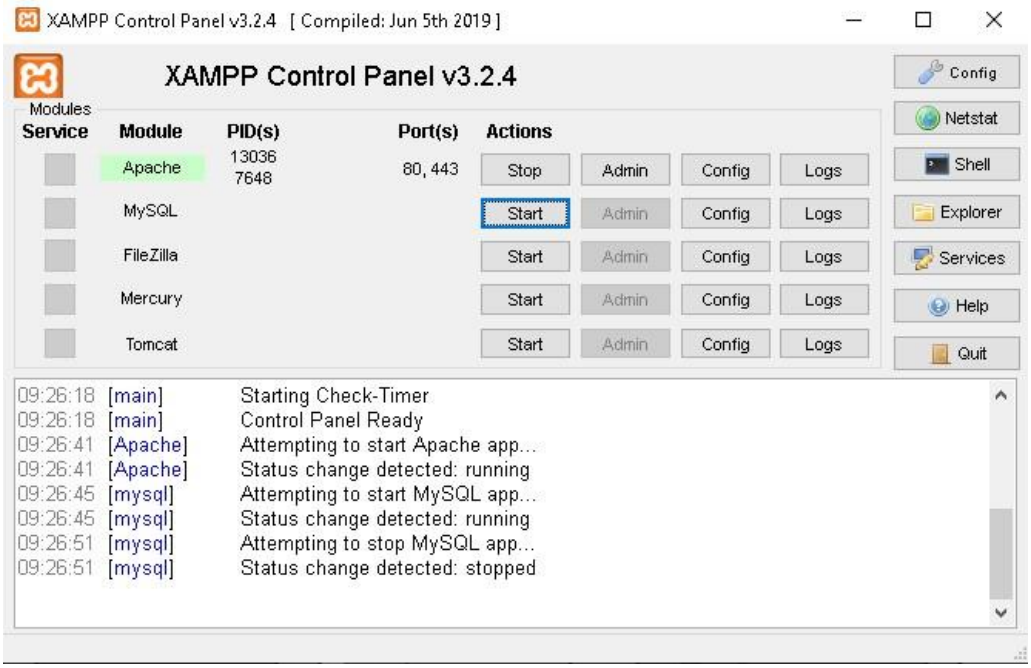


Fig 4.2 Apache Server Monitor

CHAPTER 5:

KEY FEATURES

- **User Registration and Profiles:**

Account creation with email or social media login. User profiles with customizable information, profile pictures.

- **Feed:**

Centralized feed displaying posts from followed users. Support for various types of content, such as images, and videos.

- **Posting and Sharing**:

Ability to create and share posts with images, and videos content. Options to like, comment, and share posts.

- **Following:**

Following and follower system for non-reciprocal connections.

- **Search:**

User can search posts based on it’s caption & tags.

- **User Interactions:**

Like, comment, and share functionality on posts. Emoji for expressing emotions.

- **Settings and Preferences:**

Basic Profile section Theme Mode (Dark / Light) Notification Enable/Disable Clear Cache

- **Reporting:**

User can report posts if it’s not suitable or violation any policy

- **Analytics :**

Basic analytics for tracking user engagement and app usage.

- **User Verification :**

Email authentication to avoid spam users

- **RTL Supported :**

App is supported in Arabic phones

CHAPTER 6:

RESULT AND DISCUSSION

6.1 Discussion of Code segment

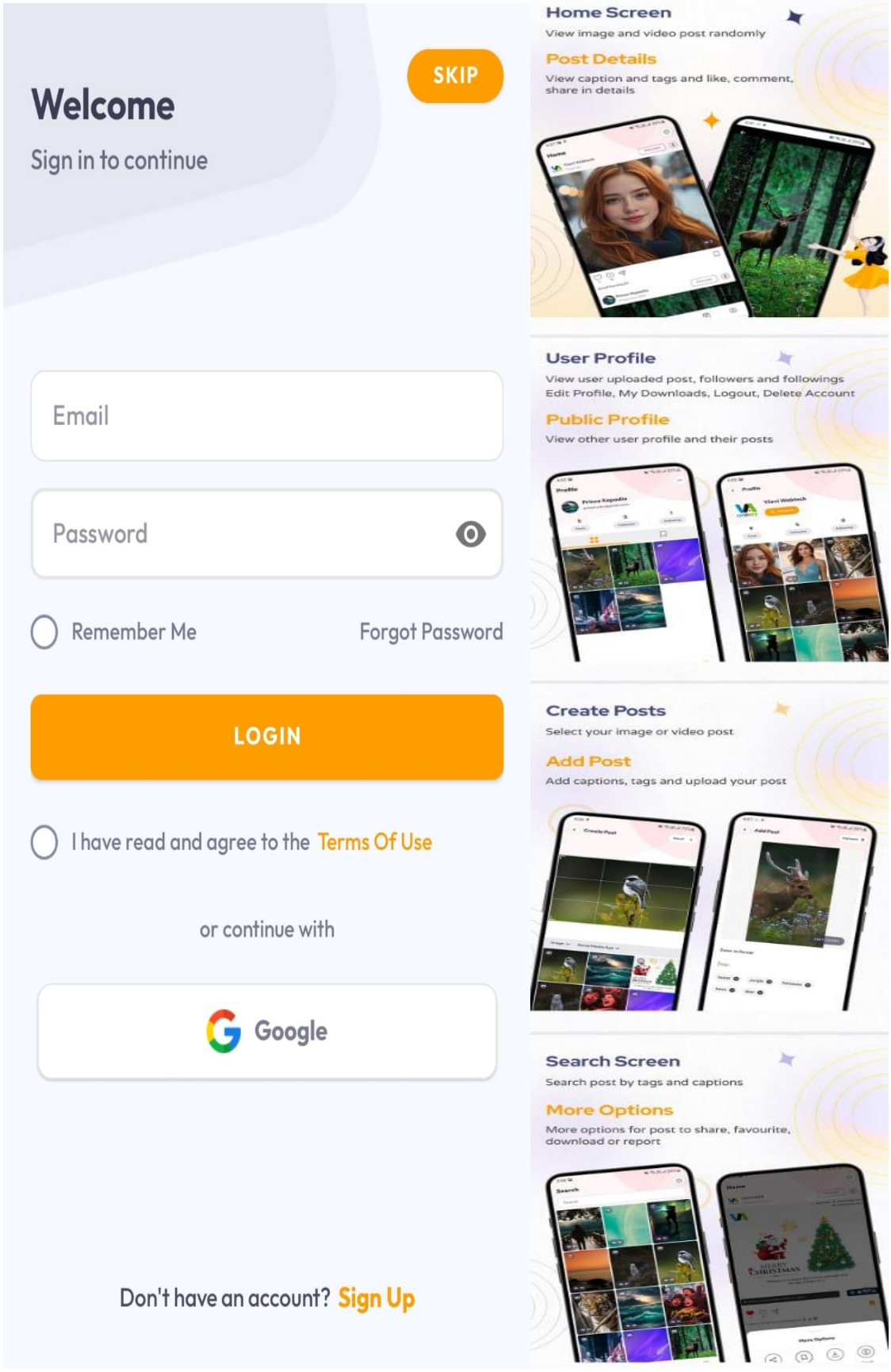
```
 <div class="login">
        <div class="col-lg-4 col-md-8 col-sm-12 bg-white border rounded p-4
shadow-sm">
            <form method="post" action="assets/php/actions.php?signup">
                <div class="d-flex justify-content-center">

                    <img class="mb-4" src="assets/images/pictogram.png" alt=""
height="45">
                </div>
                <h1 class="h5 mb-3 fw-normal">Create new account</h1>
                <div class="d-flex">
                    <div class="form-floating mt-1 col-6 ">
```

```
                        <input type="text" name="first_name" value="<?
=showFormData('first_name')?>" class="form-control rounded-0"
placeholder="username/email">
                        <label for="floatingInput">first name</label>
                    </div>
                    <div class="form-floating mt-1 col-6">
                        <input type="text" name="last_name" value="<?
=showFormData('last_name')?>" class="form-control rounded-0"
placeholder="username/email">
                        <label for="floatingInput">last name</label>
                    </div>
                </div>
                <?=showError('first_name')?>
                <?=showError('last_name')?>

                <div class="d-flex gap-3 my-3">
                    <div class="form-check">
                        <input class="form-check-input" type="radio"
name="gender" id="exampleRadios1"                               value="1"
<? =isset($_SESSION['formdata'])?'':'checked'?><?
=showFormData('gender')==1?'checked':''?>>
                        <label class="form-check-label" for="exampleRadios1">
                            Male
                        </label>
                    </div>
                    <div class="form-check">
                        <input class="form-check-input" type="radio"
name="gender" id="exampleRadios3"                               value="2"
<? =showFormData('gender')==2?'checked':''?>>
                        <label class="form-check-label" for="exampleRadios3">
                            Female
                        </label>
                    </div>
                    <div class="form-check">
                        <input class="form-check-input" type="radio"
name="gender" id="exampleRadios2"                               value="0"
<?
=showFormData('gender')==0?'checked':''?>>
                        <label class="form-check-label" for="exampleRadios2">
                            Other
                        </label>
                    </div>
                </div>
                <div class="form-floating mt-1">
```

```
                    <input type="email" name="email" value="<?
=showFormData('email')?>" class="form-control rounded-0"
placeholder="username/email">
                    <label for="floatingInput">email</label>
                </div>
                <?=showError('email')?>

                <div class="form-floating mt-1">
                    <input type="text" name="username" value="<?
=showFormData('username')?>" class="form-control rounded-0"
placeholder="username/email">
                    <label for="floatingInput">username</label>
                </div>
                <?=showError('username')?>

                <div class="form-floating mt-1">
                    <input type="password" name="password" class="form-control
rounded-0" id="floatingPassword" placeholder="Password">
                    <label for="floatingPassword">password</label>
                </div>
                <?=showError('password')?>


                <div class="mt-3 d-flex justify-content-between align-
itemscenter">
                    <button class="btn btn-primary" type="submit">Sign
Up</button>
                    <a href="?login" class="text-decoration-none">Already have an
account ?</a>
```

DISCUSSION OF THE RESULTS

6.2 Screen Shots


SKIP
Welcome
Sign in to continue
Email
Password
Remember Me
Forgot Password
LOGIN
I have read and agree to the Terms Of Use
or continue with
Google
Don't have an account? Sign Up
Home Screen
View image and video post randomly
Post Details
View caption and tags and like, comment, share in details
User Profile
View user uploaded post, followers and followings
Edit Profile, My Downloads, Logout, Delete Account
Public Profile
View other user profile and their posts
Create Posts
Select your image or video post
Add Post
Add captions, tags and upload your post
Search Screen
Search post by tags and captions
More Options
More options for post to share, favourite, download or report

CHAPTER: 7
Conclusion

In conclusion, the design of the database for a social media platform plays a critical role in ensuring smooth and efficient operation. By focusing on key features such as user registration, profiles, posts, comments, interactions, and the following system, the database architecture has been built to support a dynamic and scalable environment. The relational structure ensures that data is organized in a way that promotes easy retrieval, integrity, and security.

The implementation of essential features like a personalized feed, search functionality, user interactions, and reporting mechanisms ensures that users have a rich, interactive experience while maintaining a secure and responsive platform. Furthermore, the database design allows for growth, enabling the platform to scale as the number of users and posts increases, while also accommodating advanced features like analytics and notifications.

Security is a paramount consideration, with strategies such as email verification, password hashing, and role-based access control embedded into the database design. These measures will help prevent unauthorized access, spam, and misuse of user data, creating a safe environment for all users.

In terms of future development, there is room for enhancing features such as content personalization, better engagement analytics, and the potential to integrate machine learning algorithms to refine content recommendations. Additionally, internationalization support, particularly for RTL languages like Arabic, ensures that the platform is accessible to a broader global audience.

Overall, the designed database not only meets the current needs of the platform but also lays a solid foundation for future expansion, ensuring that the platform remains robust, user-friendly, and adaptable to changing technological and user demands.

CHAPTER: 8
References

[1] Obar, Jonathan A.; Wildman, Steve (2015). ”Social media definition and the governance challenge: An introduction to the special issue”. Telecommunications Policy. 39 (9): 745–750. doi:10.2139/ssrn.2647377. SSRN 2647377.

[2] Hasan, M. and Sohail, M.S., 2021. The influence of social media marketing on consumers' purchase decision: investigating the effects of local and nonlocal brands. Journal of International Consumer Marketing, 33(3), pp.350367.
[3] Mehra, M., Kumar, M., Maurya, A. and Sharma, C., 2021. MERN stack WebDevelopment. Annals of the Romanian Society for Cell Biology, 25(6), pp.11756-11761.
[4] Forbes, L. P., Forbes, L. P. (2013). Does Social Media Influence Consumer Buying Behavior? An Investigation Of Recommendations And Purchases. Journal of Business Economics Research (JBER), 11(2), 107–112. [5] Gonzalez-Padilla, Daniel A., and Leonardo Tortolero-Blanco. "Social ´ media influence in the COVID-19 Pandemic." International braz j urol 46 (2020): 120-124.

Websites:

- https://developer.mozilla.org/en-US/
- https://v4.mui.com/
- https://www.npmjs.com/
- https://www.mongodb.com/
- https://www.wkiipedia.com/
- https://www.w3schools.com/
- https://getbootstrap.com/docs/5.1/getting-started/introduction/
- https://www.codewithharry.com/