Development Milestone 2 – Final Report

Database and Web Application Development

DIG5127

Nishan Shrestha

Student Id: 24128429

Bsc (HONS) Computer Science with Artificial Intelligence

(Faculty of Computing, Engineering and the Buit Environment)

Birmingham City University

Sunway College Kathmandu

Kathmandu, Nepal

Nishan.shrestha@mail.bcu.ac.uk

Contents

I. Introduction

Social Sphere is an online platform to share valuable memories in the form of pictures. It is a place where people connect and oversee the moments of other people add comment to it sharing thoughts and many more. It has a user-friendly interface and responsive UI for easy accessibility in smaller devices also.

The website uses HTML, CSS and JS as front-end while PHP and MYSQL are used to handle the backend of the project. As it handles the user authentications, user registration, adding post, comments, and many more functionality. The website is designed by using Entity Relationship Model (ERM) to ensure efficient data management.

II. Platform Overview

Social Sphere is commonplace to share the memorable images add own description and posting it online and make it visible to all its followers. While others can add their post my adding a comment and appreciating it with a like. Social Sphere is a hub to gather all the digital memories of users and keep it safe for future generations.

III. Motivation behind this Project

The main purpose of developing Social Sphere is to build a visual platform that allows people to convey themselves through images. This platform offers users the chance to share photographs alongside art and images with minimal interruptions from textual content. I aim to construct a community that prioritizes both aesthetic and inspirational content across all domains including travel and fashion besides food and additional visual contents. A straightforward design philosophy guides the creation of this social media platform which replaces typical social networks through its emphasis on creative content sharing with user-friendly interface.

IV. Target Audience

Social Sphere accommodates users of diverse backgrounds because it exists as a unified platform for visual sharing of memorable content. Users at Social Sphere can document and share all types of stories through visual content whether for important events or mundane happenings or artistic ventures in a supportive community environment. Users perceive Social Sphere as a professional yet engaging platform because it delivers accessibility together with visual expression capabilities.

V. Objective of this Project

Social Sphere exists to serve users from different social backgrounds with an image-sharing environment for preserving meaningful moments. The app centers visual media distribution to build a unified platform which welcomes general users together with media creators. Below are the key objectives of the Social Sphere:

1. **Provide a user-friendly platform** for sharing memorable moments through images.
2. **Foster an inclusive community** where users from all backgrounds can participate.

3. **Encourage creativity and self-expression**, allowing users to share personal stories through images.
4. **Serve as a platform for both casual users and content creators** to showcase their memorable moments.
5. **Inspire exploration and discovery** of diverse content across various interests and themes.

VI. Technologies Used

Social Sphere implements popular technology standards for its development to provide users with an uninterrupted responsive and dependable interface experience. The application development uses key technological elements including:

1. **HTML:** Used to structure its content because this technology organizes content into proper layouts that remain accessible to every user.
2. **CSS:** Used to create an attractive design with responsive capabilities which improves user interface appearance across all devices.
3. **JavaScript:** Used to enable platform users to experience interactive features and smooth operations which include image uploading capabilities and real-time systems updates.
4. **PHP:** Used for server-side operations by receiving requests and handling user data which allows it to perform effectively with the database.
5. **MySQL:** Used to securely manage all user data and images as well as other content through its reliable and scalable database approach.
6. **Bootstrap**: Used to streamline and speed up the development process, providing pre-built, responsive design components that ensure the platform is visually appealing and optimized for all screen sizes and devices.

VII. Choice of Colors and Fonts

A. Colors:

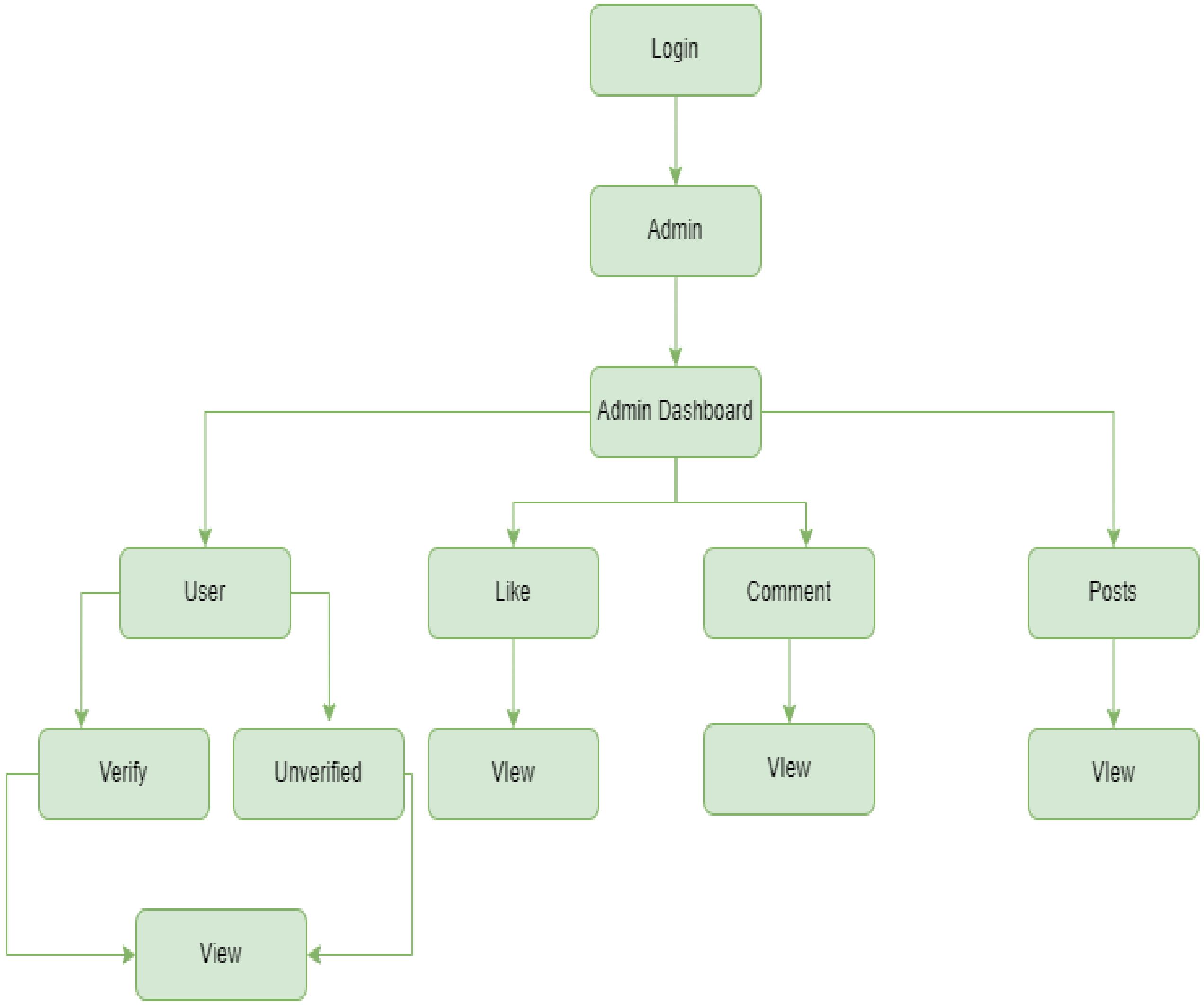
Social Sphere features a friendly user interface design through its color scheme selection. The main back section displays a light neutral hue of #F8F9FA which produces ample space, yet the sidebar maintains a subtle contrast through its soft gray shade #F2F2F2. Users notice the side bar buttons because they showcase a strong red hue #FF2F40 but the black text (#000000) gives them clear readability. The implemented design style featuring light canvas combined with bold accent elements creates an appealing interface that makes use both pleasing and practical for users.

B. Fonts:

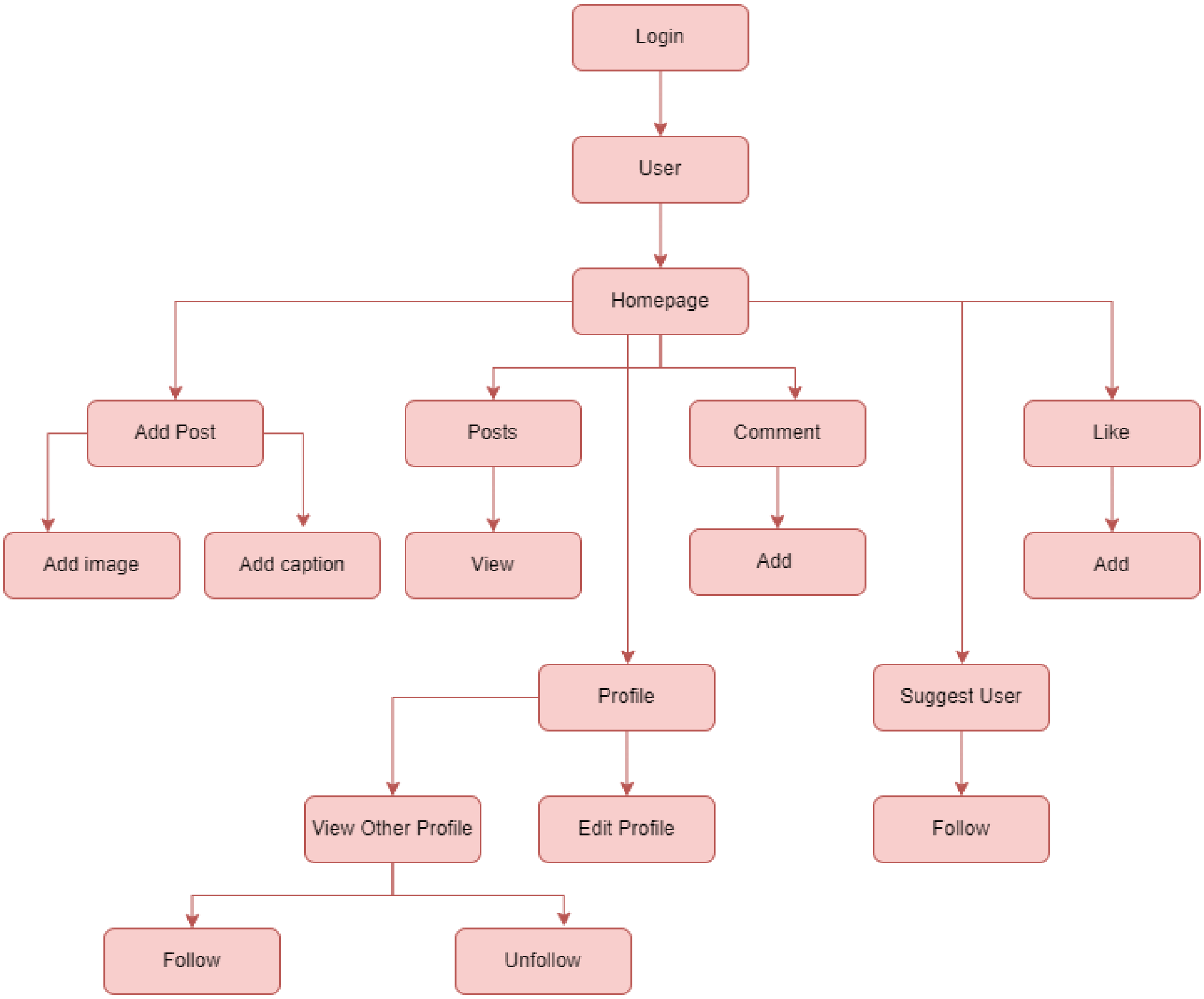
The Quicksand font serves Social Sphere as its primary choice to create a contemporary design as well as a neat and welcoming appearance. The rounded geometric style of the font works well across different devices since it offers easy readability and visual appeal. The friendly inviting nature of Quicksand can be attributed to its soft curves and balanced proportions which match the minimalist design approach of the app. Both headers and main texts benefit from this font because it provides clear visibility and contempory charm to the interface design.

VIII. Site Map

A. Admin



B. User



IX. Entity Relationship Model (ERM)

Entity Relationship Model is a model to know what entities to be captured in the database and how these capture entities are related to each other. ER data model defines enterprise schema that represents entire logical layout of database graphically.

In Social Sphere, ERM is a medium to create the database structure which will store and manage the data of the user, the content they post and the engagement the post gets.

Let's dive into the primary elements of Social Sphere ERM:

A. **Users entity:** The **users** entity is the one that will contain information on individual users of the platform. It consists of numerous features that define the user, his/her uniqueness, and the record of the person's actions. The key attributes of the **user's** entity are as follows:

1. **id:** This attribute is used to give a unique number of each natural person record within the entity.

2. **full_name:** This attribute contains the full name of the user to create a friendly approach for the user on the platform.

3. **email:** The email address of the user is saved here, which is used widely for interaction with the user and to revert previous account details.

4. **username:** This attribute stores the alias which the user selects while joining the platform and which is used while interacting on the platform, the alias is different from the user's full name.

5. **password:** This attribute has a hashed or encrypted value of a password as this is the best way to authenticate users.

6. **profile_pic:** This attribute holds the user profile image within the platform and presents the user icon within the platform.

7. **created_at:** This attribute captures the time the user account was created so that if there is any suspicious activity, his account history may be traced.

8. **updated_at:** This attribute is used to store the time that the user's profile data was last updated.

9. **account_status:** This attribute defines the status of the user's account at present, whether the account is active, suspended or deactivated, which helps in managing the users and security aspects well.

B. **Posts Entity:** The **posts** datastore is the one that oversees preserving and organizing information disclosed by the clients of the website. It captures features that allow the user to write, read and post content on the page. The key attributes of the **post's** entity are as follows:

1. **id:** This attribute provides a specific value to every post record allowing one to identify a specific record in an efficient manner.

2. **user_id:** This attribute relates each post to a particular user, using the foreign key which is the identification number of the user that has posted. It creates dependency between the posts entity and the users entity so that the post is linked to the appropriate user.

3. **post_img:** This attribute involves the image the user wants to post together with the textual content a user wants to post.

4. **caption:** This attribute contains the text shown as a description of the post content, a message, or context that can go with the shared picture or the post content.

5. **created_at:** This attribute documents the time the post was created to give information about the time the content was posted and used to sort or filter out the content based on time.

C. **Likes Entity:** The **likes** entity is implemented in order to represent the relation between the users and the posts which is primarily represented by the action of users that like a certain post. The attributes of **likes** are:

1. **id:** This attribute allocates a code to the like, making each like record to be different and easy to sort within the system.

2. **post_id:** This attribute connects the like to a particular post by an identifier of the post being liked and displayed. It creates an association between the Like entity and the Post entity thus enabling the platform to identify which posts have been liked.

3. **user_id:** This attribute records the number of likes of this post gives information on the user ID of the user who liked this post.

D. **Comment Entity:** In the **Comments** entity, we record the user generated response to the posts, which enable interactions on the platform. Below is a more detailed explanation of its attributes:

1. **id:** Comment identification number that is applied to every comment. This makes it possible for each comment to be given a unique identification so that in the management of comments they are easily manipulated in the database.

2. **post_id:** This attribute connects a comment to a specific post by leaving the comment with ID of the specific post. It organizes the comment and the post to which it belongs, thereby allowing the platform to show the right comment under the right post.

3. **user_id:** To accomplish this attribute, one must invoke the identifier of the user who made the comment from the Users entity. It makes it easy for the platform to show which of the users made a comment, to further give context to the interaction.

4. **comment:** This attribute sessions to the content of the comment but the actual comment where the user wrote a message in reply to a publication. It allows users to share their sentiments, response, opinion or comment.

5. **created_at:** This attribute captures the time at which this comment was made

E. **Follower List Entity:** The **follower_list** table is used to identify which user is in a particular list, that means status of a user, who is following or being followed by. It aids in social networking by creating and promoting social relations on the site. The key attributes of the **follower_list** entity are:

1. **id:** A more helpful definition of follow relation to identify the follow relation between two users uniquely. It also helps to avoid any overlapping elements in the **follower_list** providing each record as a separate manageable entity.

2. **follower_id:** This Vista attribute distinguishes the person who is a follower of another person in the Vista System. It also refers to the application unique ID of the user who performs the following action.

3. **user_id:** It identifies the user that is being followed by this attribute. It uses the ID of the user to whom the following is being directed and the connection between the users is established.

F. **Admin Entity:** The **admin** entity is to control administrative rights to the platform and perform system operations while observing platform functionality. The key attributes of the **admin** entity are as follows:

1. **id:** A serial number given to each of the administrators.

2. **admin_email:** This attribute records the email of the admin.

3. **password:** This attribute stores the admin to authenticate the features accessible in the administration of the platform.

X. Relationship

The relationships between the entities are structured as follows:

A. **Users and Posts:**

- **One-to-Many:** It was also discovered that a user can make many posts though a post can only belong to one user.

B. **Users and Comments:**

- **One-to-Many:** A user can write several comments, while a comment can be from only one user at a time.

C. **Posts and Comments:**

- **One-to-Many:** One post can have other posts, that is, one comment is posted and was relevant to one post only.

D. **Users and Likes:**

- **One-to-Many:** One may like several posts while a like can only pertain to one user at a time.

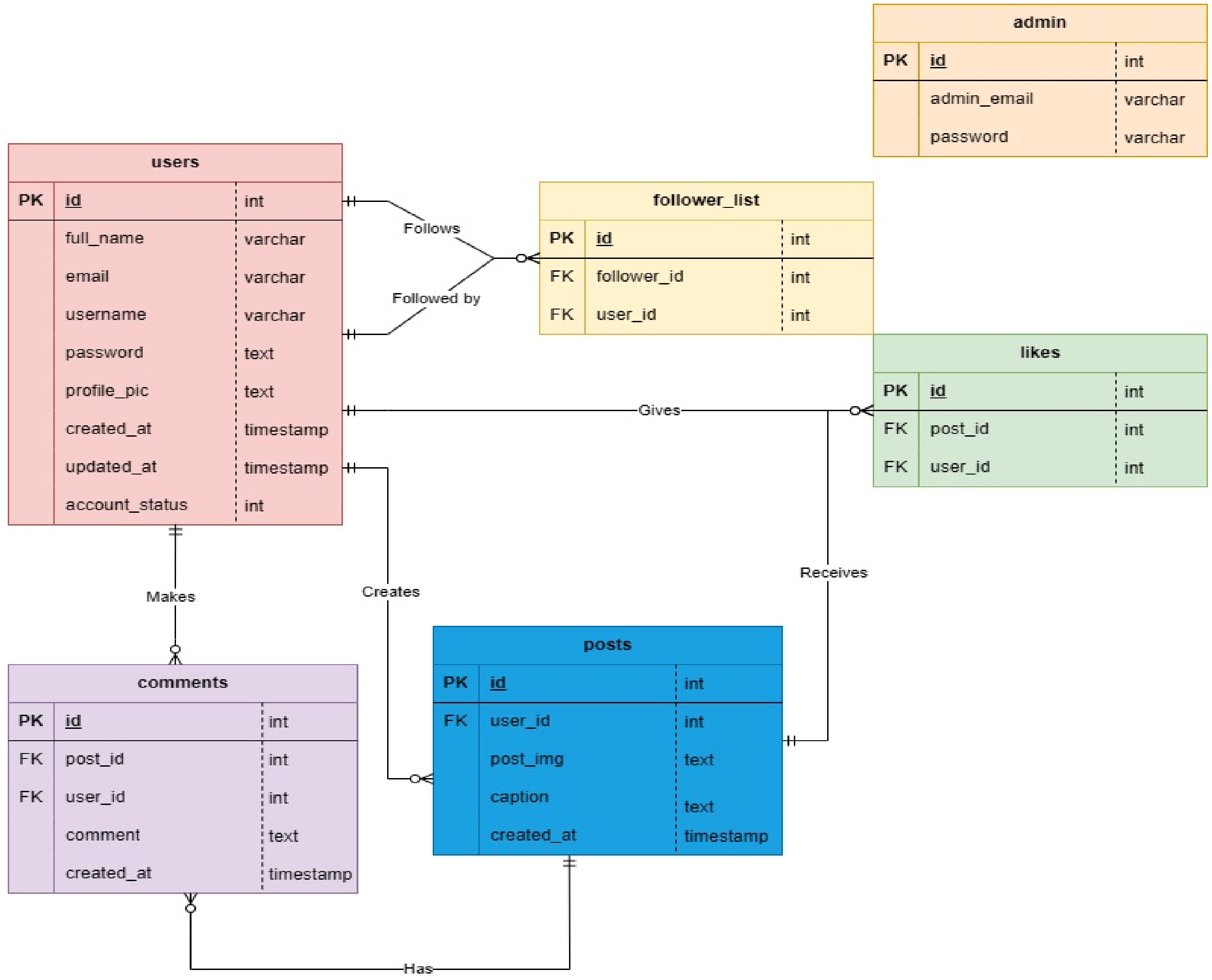
E. Posts and Likes:

- **One-to-Many:** A post can receive more than one like while a like is given to a post and cannot be given to anything else.

F. Users and Follower List:

- **Many-to-Many:** A user is free to follow as many other users as possible while, at the same time, many users can follow a certain user.

ERM Diagram of Social Sphere



XI. Code Organization Structure

Social Sphere follows a modular structure for its code organization to enable both scalability and maintainability and produce clear programming logic. The following detailed segment explains the purpose of each folder in the structure:

1. **admin/** directory includes the files that operate the administrative panel according to an organized system to run the platform.
   - **adminBootstrap/** directory specifically handles Bootstrap assets, CSS, JS and keeps an orderly responsive design for administrative implementation.
   - **images/** directory maintains all media files pertaining to administrator functions.
   - **php/** subfolder comprises PHP backend database scripts which handle the administrative capabilities like user authentication and data operations.
   - **views/** contains view files which separate administration interface logic from presentation to keep them distinct.
   - **index.php** – Serves as the main entry point for the admin panel, likely functioning as the dashboard or login page.

2. **bootstrap/** directory contains Bootstrap framework files, CSS files, and JS files which provide responsive user interface and visual consistency for the entire platform.
3. **images/** A designated directory for storing user-uploaded images and media assets, maintaining an organized file structure.
4. **php/** Houses backend PHP scripts responsible for core functionalities such as user authentication, database interactions, and image processing.
5. **views/** store's front-end view files, separating presentation logic from backend operations to improve code maintainability.
6. **index.php** The main entry point of the application, likely serving as the homepage or primary landing page for users.

XII. Features

Users can enjoy a smooth and interactive approach for sharing images through Social Sphere. The main capabilities inherent to the platform include these features:

A. ***User Features***
   - **Image Uploading:** Users can upload and share high-quality images with their followers.
   - **User Profiles:** Each user has a personalized profile displaying their uploaded images and basic details.
   - **Like & Comment System**: Users can engage with posts by liking and commenting on shared images.
   - **Follow & Unfollow:** Users can follow others to stay updated with their latest posts.

B. ***Admin Features***
   - **Dashboard Management:** Admins can access a dashboard to monitor user activity, content, and statistics.
   - **User Management**: Admins verify the unverified accounts.

C. ***Security Features***
   - **Secure Authentication:** User data is protected with secure login and session management.

- **Optimized Database:** The platform uses a well-structured MySQL database for efficient data retrieval.
- **Error Handling & Validation**: Input validation and error handling ensure a smooth and bug-free user experience.

XIII. Challenges Faced and Solved

While developing Social Sphere there were multiple challenges on the way, such as handling follow and unfollow effectively, adding like to the post, preview images while uploading, and many more. It was a tough journey but with proper guidance from my teacher and some research I have found the way to solve the issue.

It was not an easy task but with implementation of "fetch" method for updating only the certain elements on the web page helped me a lot and it was an efficient way to update data and a fast way. Using proper JavaScript function and method to show the images and it was fun while debugging it and building the project.

XIV. Future Enhancements

There are many features that can be implemented but due to lack of my knowledge and time the features cannot be added now. The improved version will have features like personal chatting, adding videos, and many more. These features will help user to interact more with others, make new friends across the globe.

XV. Conclusion

Social Sphere functions as an interactive user-friendly platform for image-sharing activities that impact users with convenient social interactions. The platform establishes efficient performance through its well-structured codebase and combines this with responsive design and secure backend development for delivering scalable and user-friendly execution. Users have full control over their image sharing activities because they can both upload content and share photos and participate with interactive options that include liking and commenting and following other members.

The strategic use of web technologies including HTML, CSS, JavaScript, Bootstrap and PHP and MySQL allows for the creation of an attractive functional platform on Social Sphere. The platform operates at high speeds and remains easy to maintain through its optimized database structure and security measures combined with its modular design.

Social Sphere provides users with both an exciting and protective space to easily exchange their noteworthy life experiences regardless of their background. The image sharing social media platform targets development of a leading platform through ongoing enhancements and new features.