

SocialHive Planning Document

Elisea Mizzi

1. Description of the Web Application

1.1 Description

SocialHive is a dynamic social networking platform that I have built using PHP and MySQL. It is designed for users to connect, share content, and create communities. The platform focuses on user interaction, allowing people to share posts, comment, message each other, join groups, and follow other users. My goal is to create a user-friendly, engaging, and interactive social media environment.

1.2 Target Audience

SocialHive is designed for individuals of all ages who wish to interact with others based on common interests. It's a platform for sharing ideas, connecting with like-minded people, and exploring various topics in different interest groups.

2. Features

2.1 Core Features

1. User Authentication

- **Sign Up:** Users can register by providing a username, email, and password.
- **Log In:** Registered users can log in using their credentials.
- **Log Out:** Users can securely log out.
- **Password Recovery:** Users can reset their password via email.

2. Profile Management

- **View Profile:** Display user information, including bio, profile picture.
- **Edit Profile:** Update the bio and profile picture.

3. Content Posting

- **Create Posts:** Users can post text-based content.
- **View Posts:** Display posts made by others in the feed.
- **Delete Posts:** Users can delete their own posts.

4. Messaging System

- **Send Messages:** Users can send private messages to each other.
- **View Messages:** Display a list of received messages.
- **Delete Messages:** Users can delete messages.

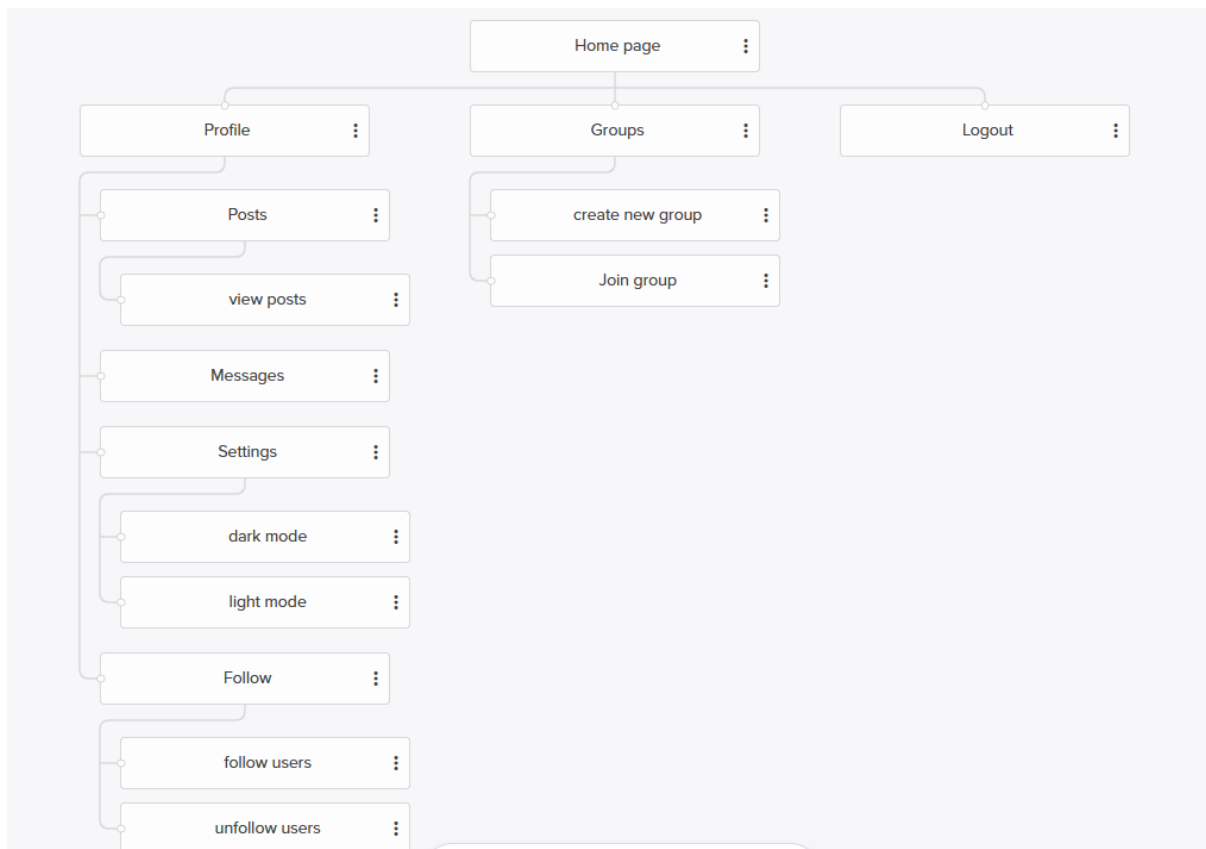
5. Follow System

- **Follow Users:** Users can follow other users to see their posts.
- **Unfollow Users:** Users can unfollow users they no longer wish to follow.
- **View Followers:** Users can see a list of people following them.

6. Basic Admin Panel

- **Admin Login:** Admin can log in to manage users and posts.
- **Manage Users:** Admin can view, delete, or ban users.
- **Manage Posts:** Admin can view and delete posts.
- **View Notifications:** Admin can view user activities and content.

3. Sitemap



3.1 Frontend Pages (User-Facing)

1. **Home Page:** Displays recent posts, user activities, and notifications.
2. **User Authentication:**
 - login.php: User login page.
 - signup.php: New user registration page.

- logout.php: Logs out the current user.

3. Profile Management:

- profile.php: Displays user profile, posts, and followers.
- edit_profile.php: Update profile details (e.g., bio, profile picture).
- settings.php: Modify account settings (privacy, notifications).

4. Content Posting:

- create_post.php: Form to create a new post.
- view_post.php: View individual posts and comments.
- edit_post.php: Edit existing posts.
- delete_post.php: Delete a post.

5. Group Management:

- create_group.php: Create a new group.
- view_group.php: View group details and members.
- edit_group.php: Edit group settings.
- delete_group.php: Delete a group.

6. Messaging System:

- send_message.php: Send private messages.
- view_message.php: View conversation threads.
- edit_message.php: Edit sent messages.
- delete_message.php: Delete messages.

7. Follow System:

- follow_user.php: Follow another user.
- unfollow_user.php: Unfollow a user.
- followers_list.php: View list of followers and following.

8. Notifications:

- notifications.php: View notifications.
- update_notifications.php: Change notification preferences.

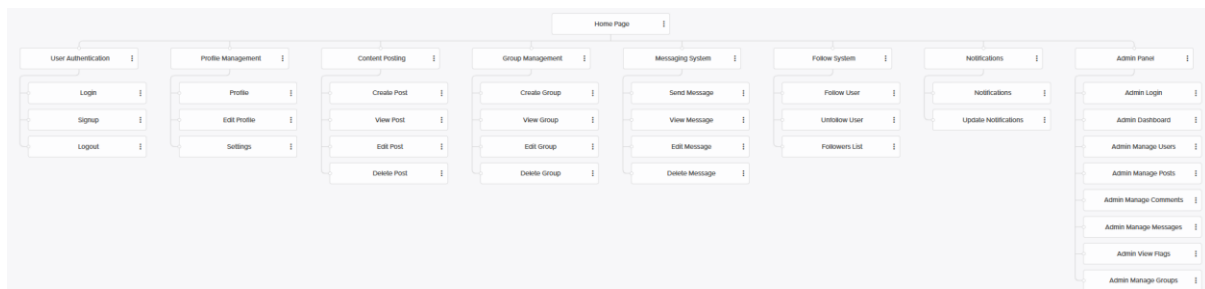
3.2 Admin Panel Pages:

1. Admin Login:

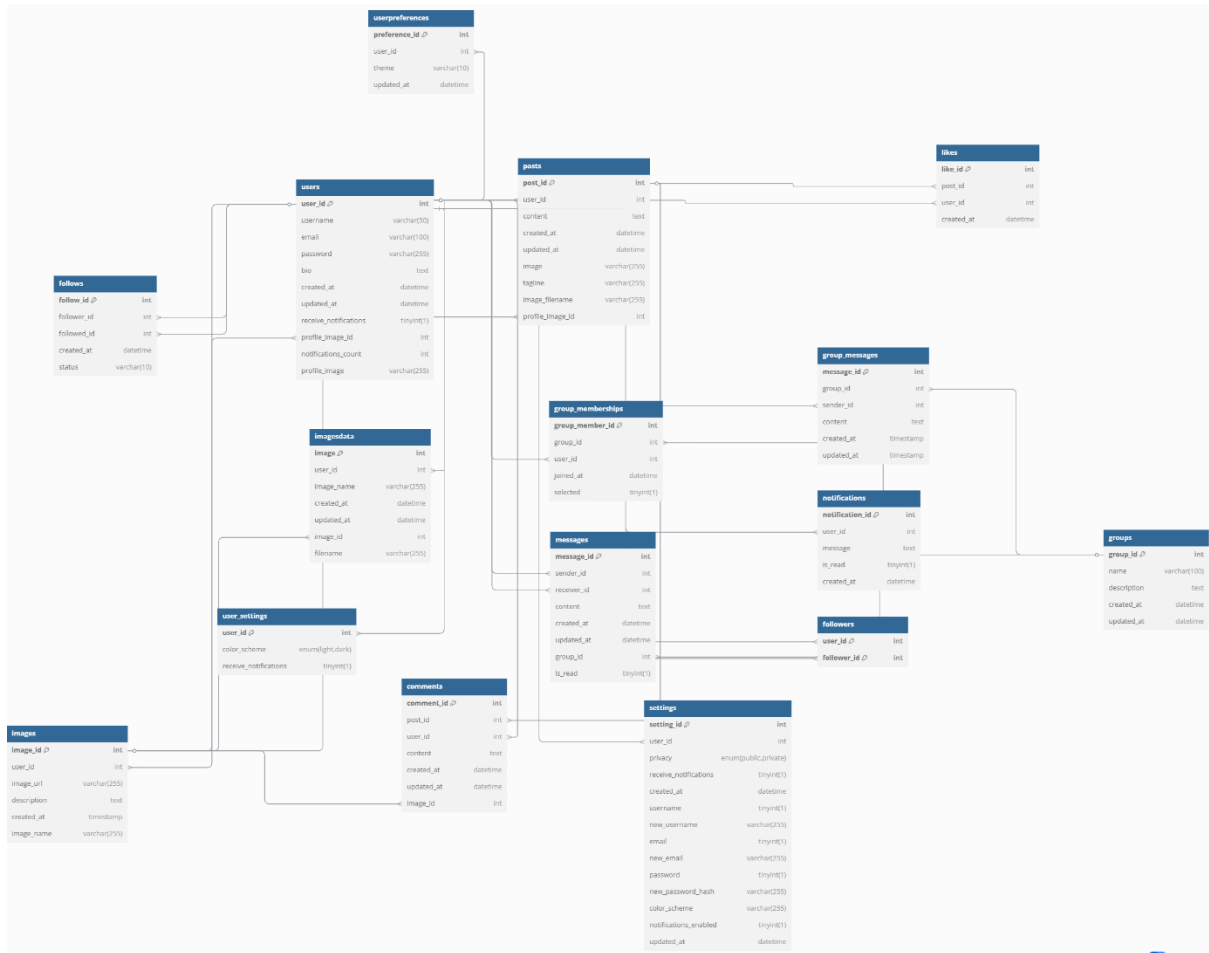
- admin_login.php: Login page for admins.

2. Admin Dashboard:

- admin_dashboard.php: Overview of platform stats (total users, posts, flagged content).
3. **Manage Users:**
 - admin_manage_users.php: View, delete, or ban users.
 4. **Manage Posts:**
 - admin_manage_posts.php: View and remove posts.
 5. **Manage Comments:**
 - admin_manage_comments.php: View and delete comments.
 6. **Manage Messages:**
 - admin_manage_messages.php: View and delete private or group messages.
 7. **Flagged Content:**
 - admin_view_flags.php: View and resolve flagged posts, comments, or messages.
 8. **Manage Groups:**
 - admin_manage_groups.php: View and delete groups.



4. ERD Diagram for Database Structure



Explanation of Relationships:

- Users and Posts:** A user_id in the **posts** table refers to the user_id in the **users** table (foreign key).
- Users and Comments:** A user_id in the **comments** table refers to the user_id in the **users** table.
- Users and Likes:** A user_id in the **likes** table refers to the user_id in the **users** table.
- Users and Messages:** A sender_id and receiver_id in the **messages** table refer to the user_id in the **users** table.
- Users and Images:** A user_id in the **images** table refers to the user_id in the **users** table.
- Users and Group Membership:** A user_id in the **group_memberships** table refers to the user_id in the **users** table.
- Users and Notifications:** A user_id in the **notifications** table refers to the user_id in the **users** table.
- Users and Settings:** A user_id in the **settings** table refers to the user_id in the **users** table.

9. **Users and User Preferences:** A user_id in the **userpreferences** table refers to the user_id in the **users** table.

5. Database Structure

5.1 Database Tables

- **Users Table:**
 - id (Primary Key)
 - username (VARCHAR)
 - email (VARCHAR)
 - password (VARCHAR)
 - profile_picture (VARCHAR)
 - bio (TEXT)
- **Posts Table:**
 - id (Primary Key)
 - user_id (Foreign Key to users.id)
 - content (TEXT)
 - created_at (DATETIME)
- **Messages Table:**
 - id (Primary Key)
 - sender_id (Foreign Key to users.id)
 - receiver_id (Foreign Key to users.id)
 - content (TEXT)
 - created_at (DATETIME)
- **Followers Table:**
 - id (Primary Key)
 - follower_id (Foreign Key to users.id)
 - followed_id (Foreign Key to users.id)
- **Groups Table:**
 - id (Primary Key)
 - creator_id (Foreign Key to users.id)
 - group_name (VARCHAR)

- description (TEXT)

6. File Structure

/socialhive

- assets
 - css
 - js
 - images
- includes
 - header.php
 - footer.php
- config
 - db.php
- pages
 - login.php
 - signup.php
 - profile.php
 - home.php
 - create_post.php
 - view_post.php
 - settings.php
- admin
 - admin_login.php
 - admin_dashboard.php
 - admin_manage_users.php
 - admin_manage_posts.php
- functions
 - user_functions.php
 - post_functions.php
 - message_functions.php
- gitignore
- README.md

7. Test Strategy

- **Unit Testing:** Test individual functionalities such as user sign-up, login, post creation, and messaging.
- **Integration Testing:** Ensure that various components interact smoothly (e.g., a post shows up in the user feed, messages appear in the inbox).
- **System Testing:** Test the application as a whole to ensure everything works together seamlessly.
- **Acceptance Testing:** Verify that the platform meets user requirements and works as expected.

- **Performance Testing:** Ensure the application can handle many users and posts without degrading performance.

8. Implementation Details

8.1 Technologies Used

- **Frontend:** HTML, CSS, JavaScript (for client-side behaviour)
- **Backend:** PHP (for server-side logic)
- **Database:** MySQL (to store data for users, posts, messages, etc.)

8.2 User Authentication Logic

- **Sign Up:** When a user submits the registration form, their data is validated and stored in the users table. The password is hashed for security before being saved.
- **Log In:** The system checks the submitted username and password against the users table. If valid, the user is logged in.
- **Password Recovery:** A simple password reset form is provided where users can request a password reset link (email functionality could be added).

8.3 Profile Management

- **Edit Profile:** Allows users to update their bio and upload a new profile picture.
- **Profile Display:** A user's profile page displays their posts, bio, and profile picture.

8.4 Content Posting

- **Post Creation:** Users can create text-based posts through a form that submits to the posts table.
- **Post Deletion:** Users can delete their own posts by sending a delete request to the backend.

8.5 Messaging System

- **Send and View Messages:** Users can send and view messages between each other. Messages are stored in the messages table, with a reference to the sender and receiver.

8.6 Follow System

- **Follow and Unfollow Users:** When a user follows another user, a record is created in the followers table. When they unfollow, the corresponding record is deleted.
- **View Followers:** A user can view their followers by querying the followers table.

9. Next Steps

- **Integrate Email for Notifications and Password Recovery**
- **UI/UX Design Enhancements**
- **Scalability Enhancements** (e.g., adding caching for performance)

10. IPO Chart

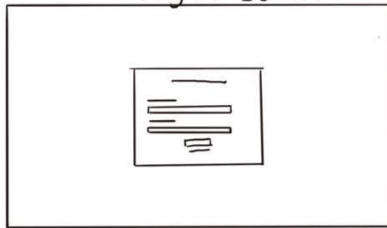
FEATURE	INPUT	PROCESS	OUTPUT
USER REGISTRATION	User provides a username, email, password, and confirms email	System validates inputs (e.g., email format, unique username), hashes password, stores user data in the database	New user account is created, user is redirected to the login page
USER LOGIN	Username and password input by the user	System checks the database for matching credentials, verifies password hash	User is authenticated and redirected to the home page, or error message
POST CREATION	User inputs text content and optional media (images, videos)	System validates input (content length, file size), stores post in the database with user info	New post is displayed in the feed, confirmation message is shown
POST DELETION	User selects "Delete" option for a post	System checks if the user is the post owner, deletes the post from the database	Post is removed from the feed, and the post data is deleted from the DB
FOLLOW USER	User clicks "Follow" on another user's profile	System creates a record in the followers table linking the follower and the followed user	Followed user's posts are displayed in the follower's feed
UNFOLLOW USER	User clicks "Unfollow" on a followed user's profile	System deletes the record in the followers table linking the follower and the followed user	Followed user's posts are no longer visible in the follower's feed
SEND MESSAGE	User inputs a recipient and message content	System checks if the recipient exists, stores the message with sender and receiver IDs in the messages table	Message appears in the recipient's inbox, and a confirmation is shown to the sender
VIEW POST DETAILS	User clicks on a post to view details	System retrieves post details (content, comments, likes) from the database	Post details are displayed with options to comment, like, or share

COMMENT ON POST	User types a comment and submits it	System stores the comment in the comments table, associates it with the post and user	Comment is displayed under the post, and the post's comment count is updated
LIKE POST	User clicks "Like" on a post	System creates a record in the likes table linking the user and the post	Like count on the post is updated, and the like is reflected in the UI
VIEW PROFILE	User clicks on their own or another user's profile	System retrieves and displays user details (bio, posts, followers) from the database	User profile is displayed with posts and follower/following info
EDIT PROFILE	User edits bio, profile picture, and other details	System updates the profile fields in the database with new values	Updated profile is displayed on the user's page
ADMIN MANAGE USERS	Admin selects a user to manage	System retrieves the user's data, admin can ban or delete the user from the platform	User is either banned or deleted, and the platform is updated
ADMIN MANAGE POSTS	Admin selects a post to manage	System retrieves post details, admin can delete flagged or inappropriate posts	Post is removed from the platform, and the content is updated
ADMIN MANAGE COMMENTS	Admin selects a comment to manage	Admin reviews, deletes inappropriate comments from the database	Comment is removed from the platform, and the comment count is updated
ADMIN MANAGE MESSAGES	Admin selects a message to manage	Admin reviews, deletes inappropriate messages from the database	Message is removed from the platform, and the message data is updated
VIEW NOTIFICATIONS	User clicks on the notifications icon	System retrieves notifications from the database for the user	Notifications are displayed, and the user can mark them as read

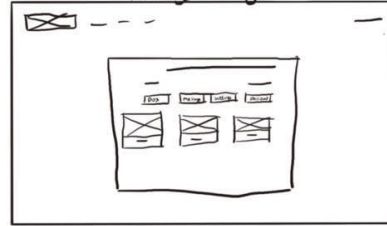
11. Wireframes

Front end design

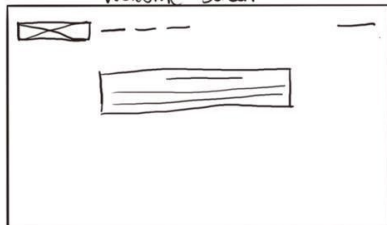
Log in screen



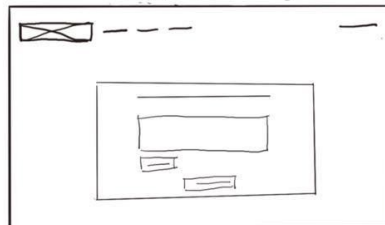
Profile Page



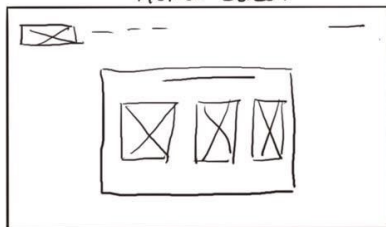
Welcome screen



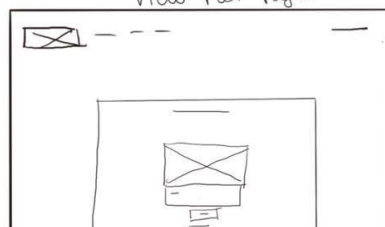
Add Post Page



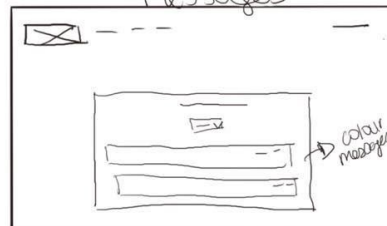
Home Screen



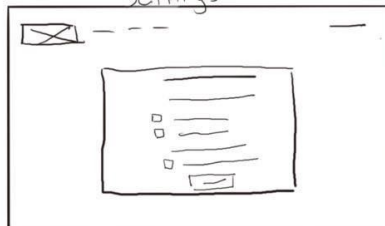
View Post Page



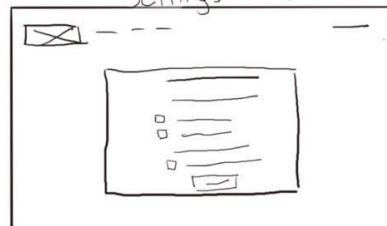
Messages



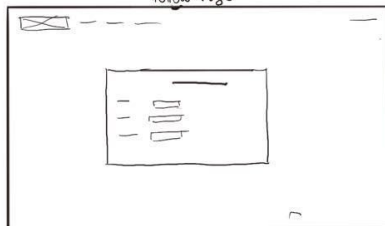
Settings



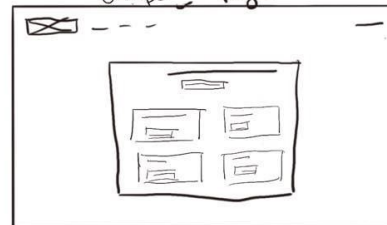
Settings



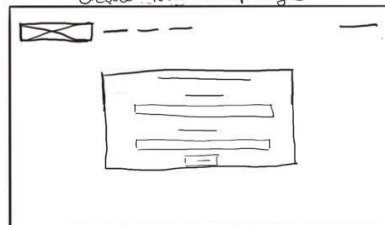
Follow Page



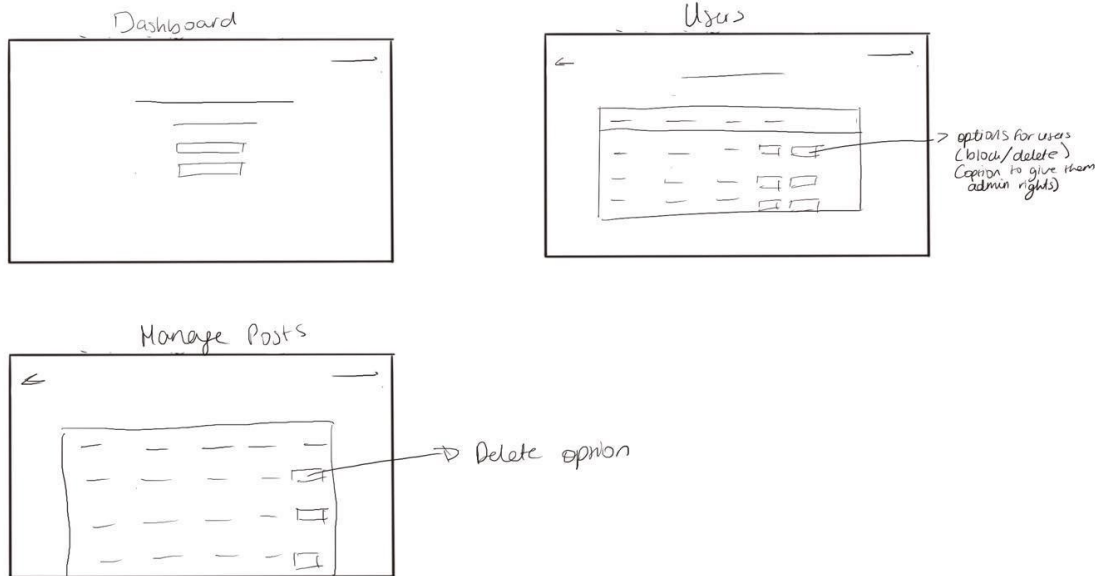
Groups Page



Create New Group Page



Admin side design



Admin side will be able to view all the users posts simultaneously allowing them to delete posts, block or delete users as well as give them and remove their admin rights.