Prototype Criteria

- The prototype must include a **Login Page** with fields for username and password and a working login button.
    - The login page will also be the basis for the **Registration Layout**.

- The prototype must include a **Home Page** that displays core navigation options
    - Includes a **Search function** …
    - On the Home Page the following elements must be present:
        - Navigation bar along the top banner
        - Accessibility buttons on the bottom row
        - Profile page access on the top left
        - Timeline feed as the main focus
            - In the Timeline will be the posts that feature various media that the user takes interest in.

- Within the home page the prototype must include a **User Profile Page** that shows user details, editable information, and user activity/posts.
    - Even if the app aims to have anonymity for its users, there shall be a system that measures the validity of the user's posts.

- The prototype must include a **Discover Page** that allows users to browse or search posts/shops.

- The prototype must include a **Support Page** that provides help or contact options for users.
    - The support page will act as a menu for the following pages:
        - Technical Support
        - Account Status
        - Sending a Ticket
        - Terms and Policies
        - About us

- The prototype must include a **Navigation bar** that is present all throughout the pages for easier access.

ERD Criteria

1. Entities

The ERD must include the following entities: **User_Account, Role, Post, Report**.

- **User_Account**
  - PK: `user_id`
  - Attributes: `username`, `email`, `password`, `birthdate`, `phone_number`, `join_date`, `profile_picture`, `preferences`, `favorites`,upvote_ratio, `role_id` (FK).
  - Users manage their own accounts.
  - Admins can manage and timeout other accounts. Additionally, admins can take down flagged posts after reviewing them.

- **Role**
  - PK: `role_id`
  - Attributes: `admin`, `member`.
  - Only admins can create other admin accounts.

- **Post**
  - PK: `post_id`
  - Attributes: media_type, media_title, description, image_url, created_at, shop, location, user_id (FK)
  - Created by users.
  - Can include `shop` and `location` as metadata.
  - Only the post owner or admins can edit/delete.
  - Viewable by all users.

- **Report**
  - PK: `report_id`
  - Attributes: `reason`, `status`, `post_id` (FK), `user_id` (FK).
  - Created by users to flag posts.
  - Status managed by admins

2. Relationships

- **User_Account–Role**: One-to-many (one role can apply to many users).

- **User_Account–Post**: One-to-many (a user can create multiple posts).

- **User_Account–Report**: One-to-many (a user can file multiple reports).

- **Post–Report**: One-to-many (a post can be reported multiple times).

3. Keys and Constraints

- Each entity has a **unique primary key** (`user_id`, `role_id`, `post_id`, `report_id`).

- **Foreign keys** enforce referential integrity:
    - `role_id` in `User_Account`
    - `user_id` in `Post`
    - `user_id` in `Report`
    - `post_id` in `Report`

- Attributes `shop` and `location` in `Post` are simple fields, not separate entities.