

Project Structure

1. User Roles

- **Admin (Landlord)**
 - Upload, update, and delete house listings with images.
 - Respond to user comments/questions.
 - View analytics on user engagement (likes, comments).
 - **Users (Tenants/Comrades)**
 - View available houses.
 - Like, comment, and ask questions on listings.
 - Receive replies from landlords.
-

2. System Modules & Components

A. Frontend (User Interface)

1. **Homepage (House Listings Page)**
 - Displays houses with images, descriptions, price, and location.
 - Shows likes and comments per listing.
 - Search and filter options (e.g., price range, location).
 2. **House Details Page**
 - Full details of a specific house.
 - Like and comment functionality.
 - Comments section with landlord responses.
 3. **Admin Login Page**
 - Secure login for landlords.
 - Password recovery option.
 4. **Admin Dashboard**
 - Overview of houses listed.
 - User engagement analytics (likes, comments).
 - Manage listings (add/update/delete).
 - Respond to user comments.
-

B. Backend (Server & Database Structure)

1. User Authentication System

- Secure login for landlords.
- Registration (if needed).

2. Database Schema (Tables/Collections)

- **Users Table:** ID, name, email, password, role (landlord/user).
- **Houses Table:** ID, landlord_id, images, description, price, location, status.
- **Comments Table:** ID, house_id, user_id, comment, timestamp.
- **Likes Table:** ID, house_id, user_id.

3. API Endpoints (Backend Logic)

- **User Authentication APIs**
 - Login, logout, password reset.
- **House Management APIs**
 - Add, update, delete houses.
- **Interaction APIs**
 - Like a house.
 - Post comments.
 - Landlord replies to comments.

3. Tech Stack Recommendation

Frontend:

- HTML, CSS, JavaScript (React or Vue.js for dynamic UI)

Backend:

- Python (Flask/Django) or Node.js (Express.js)

Database:

- MySQL or MongoDB (for structured storage)

Hosting & Deployment:

- Firebase/Auth0 for authentication
 - Cloudinary for image hosting
 - DigitalOcean, AWS, or Firebase for deployment
-