

Documentations of Real Estate management system (php Laravel)

Project Name: Real Estate Management System

Project Type: Web Application

Platform: PHP Laravel

Used Programming Languages: HTML, HTML5, CSS, CSS3, Bootstrap, JavaScript

Database: MySQL

Software Tools: Laravel, Composer, XAMPP, Visual Studio Code

Supported Browsers: Internet Explorer, Opera Mini, Google Chrome, Mozilla Firefox

Supported Operating Systems: Windows 7, Windows 8, Windows 10, Linux, Mac

Project Overview

The **Real Estate Management System** is a web application developed using Laravel designed to facilitate the buying, selling, and renting of properties. It connects buyers, sellers, and real estate agents, allowing users to search for properties, manage listings, and communicate effectively.

Features

User Interface:

- 1. User Registration and Login**
 - Buyers, sellers, and agents can register and create accounts.
 - Secure login for returning users.
- 2. Property Listings**
 - Users can browse through available properties with filtering options (e.g., location, price, type).
 - Detailed views for each property, including images, descriptions, and pricing.
- 3. Property Management (for Agents)**
 - Real estate agents can add, edit, and remove property listings.
 - Manage inquiries and view application statistics.
- 4. Search and Filter Options**
 - Advanced search functionality to find properties based on various criteria.
- 5. Contact Form**
 - Interested buyers can contact sellers or agents directly through the platform.

Admin Panel:

1. **Admin Login**
 - Secure access for administrators to manage the system.
 2. **Manage Users**
 - View and manage buyer, seller, and agent accounts.
 - Approve or deactivate user registrations.
 3. **Manage Properties**
 - Add, edit, or delete property listings.
 - Approve property listings before they go live.
 4. **Inquiries Management**
 - View and respond to inquiries from potential buyers.
 5. **Generate Reports**
 - Generate reports on property activity, user statistics, and inquiries.
-

Software Requirements

- **PHP:** 7.2 or higher
 - **MySQL:** 5.7 or higher
 - **Laravel:** 8.x or higher
 - **Web Server:** Apache or Nginx
 - **Database Management:** phpMyAdmin or similar tool
-

How to Configure (Run) the Project

1. **Download the Project Zip File**
 - Download the zip file containing the source code of the Real Estate Management System.
2. **Unzip and Place in Root Directory**
 - Unzip the project file and place the extracted files in the root directory of your web server (e.g., `htdocs` in XAMPP).
3. **Install Dependencies**
 - Navigate to the project directory in the terminal and run:

```
bash
Copy code
composer install
```
4. **Create and Import Database**
 - Open phpMyAdmin (<http://localhost/phpmyadmin>).
 - Create a new database (e.g., `real_estate_db`).
 - Import the SQL file into the database.
5. **Configure Environment Settings**
 - Rename `.env.example` to `.env` and update the database credentials.
6. **Generate Application Key**

- In the terminal, run:

```
bash
Copy code
php artisan key:generate
```

7. Access the Application

- User Interface URL: `http://localhost/real_estate/public/`
- Admin Panel URL: `http://localhost/real_estate/public/admin`

8. Login Credentials

- Admin Login:
 - **Username:** Admin
 - **Password:** Admin

UI/UX Wireframes

- **Homepage:** Overview of available properties, featured listings, and search functionality.
- **Property Listing Page:** Display of properties with filtering options.
- **Property Details Page:** Detailed view of the property with images, descriptions, and inquiry forms.
- **User Dashboard:** Overview of saved properties, inquiries, and account management.
- **Admin Dashboard:** Overview of property listings, user management, and key metrics.

Error Handling Guidelines

1. **Validation Errors**
 - Display user-friendly messages for form validation errors (e.g., "Please enter a valid email address").
2. **Database Errors**
 - Show generic error messages and log detailed errors for debugging purposes.
3. **404 Errors**
 - Display a custom 404 error page indicating that the requested resource was not found.
4. **500 Internal Server Errors**
 - Display a generic error message and log detailed errors for further analysis.

Business Logic

1. **User Registration**
 - Validate user input and securely store user credentials in the database.

2. **Property Management**
 - Allow administrators and agents to create, update, and delete property listings.
 3. **Inquiry Management**
 - Handle inquiries from interested buyers, allowing communication between users.
 4. **Reporting**
 - Generate and display reports based on property listings and user activity.
-

Session Management

1. **Session Initialization**
 - Use Laravel's built-in session management to handle user logins.
 2. **Session Protection**
 - Ensure that only authenticated users can access admin functionalities.
 3. **Access Control**
 - Define user roles and permissions to restrict access to specific features.
-

PHP Functions

1. **OnLoad Functions**
 - Implement functions to optimize the performance of Laravel routes and controllers.
 2. **Database Connectivity**
 - Use Laravel's Eloquent ORM for secure database interactions.
 3. **Utility Functions**
 - Implement utility functions for common tasks such as input validation and data processing.
-