

# Mi-12 Assignment Requirement

assignment12\_category\_0010

🚩 : 0 [ If we have any update we will mention it here ]. Do check frequently to see if any updates have been made.

## Objective

You must develop a **Real estate Platform** (an online platform where people can add and buy real estate properties) using the MERN stack.

## Main Tasks

1. Make sure your design and website idea is unique. First, finalize your idea (what type of website you want to build). Then google the site design or visit `themeforest` to get your website idea. [\[You can visit this blog to collect free resources for your website\]](#). However, your website `can not be related to your previous assignments or any demo project displayed in the course or in our conceptual sessions`

1. Home page will have a navbar, banner/slider, Advertisement section, Latest User review section and a footer.

- **Advertisement section:** In this section there will be at least 4 cards which will contain

- property image.
- Property location.
- price range.
- verification status.
- Details button.(clicking on the details button will redirect the user to the property details page. See requirement 6)

- **Latest User review:** In this section there will be at least 3 latest (recently added) user reviews which will contain:

- Reviewer name.
- Reviewer Image.
- Review description.
- property title.

2. The navbar will have a website name with the logo, Home, All properties, Dashboard and Login. Your website will have these routes. Except for the Home route, other routes will be private.

3. After successfully performing any CRUD - Create, Read, Update, Delete operation, please, show a relevant toast/sweet alert. (Don't use the browser alert() function. otherwise, marks will be reduced)

4. Add two extra sections to the home page in addition to the sections mentioned above.

5. **All properties page:** This page will contain all the admin verified properties which are added by all the real estate agents. Each card will contain the following informations:

- Property image.
- Property title.
- property location.
- Agent name.
- Agent image.
- verification status.
- Price range.
- Details button. (clicking on the details button will redirect the user to the property details

page. See requirement 6)

-This will be a private/protected route.

6. **Details page:** This page should display detailed information about the property. What you will include in the detailed information is entirely up to you but make sure to add property title, description, price range and agent name.

-There will be a button on the details page called "Add to wishlist". Clicking this button will add the property on the wishlist page and also the information will be saved to the database.

-There will be a review section at the bottom of the details page where you have to show all the reviews for that specific property which are made by the users.

-There will also be a review button on this page. Clicking this button a user can add a review for this specific property through a modal.

-This will be a private/protected route.

## 7. User Dashboard (Private Route):

-When a user clicks on the Dashboard, he/she will be redirected to a page where there will be the following routes:

- A. My Profile.
- B. Wishlist.
- C. Property bought.
- D. My reviews.

**My Profile page:** On this page there will be some information about the logged in user. The information must include user name, user image, role (if the user is a regular user then don't show anything). You can add other relevant information if you wish.

**Wishlist:** This page will contain all the properties which the user has wishlisted by clicking the "Add to wishlist" button on the details page. Each card will contain the following information:

- Property image.
- Property title.
- property location.
- Agent name.
- Agent image.
- verification status.
- Price range.
- Make an offer button.
- remove button.

- Clicking the "Make an offer" button will navigate the user to a new page where there will be a form which will have some input fields.

- Property title (readonly)
- Property location (readonly)
- Agent name (readonly)
- Offered amount. (User can not add an amount which is not in range of the price which was specified by the agent of that property)
- buyer email (readonly)
- buyer name (readonly)
- buying date.
- offer button

- By clicking the offer button the user can offer an amount (This amount must be in between the agent specified price range. If the price is not in between the price range then the offer will not be completed and an error message will be shown) to the agent for that specific property. This information will be saved to the database and also it will be added to users "Property bought" page with the status "pending".

**Property bought:** All the properties which the user has offered for will be shown here. Each card will have the following information.

- property location.
- property title.
- property image
- Agent name
- offered amount.
- status

- There will be a status on each card. Initially the status will be "pending". If the offer is accepted by the agent then this status will be "accepted" and a "Pay" button will appear on the card. Clicking the "Pay" button will redirect user to the payment page where the user can pay the offered amount through stripe or any other payment methods for that property. If the payment is completed then the status will be "bought" also payment transaction id will be shown in place of the pay button.

**My reviews:** In this page a logged in user will see only his/her reviews which he/she has given for any property. Each review card will contain

- property title
- agent name
- review time
- review description
- delete button.

- If the user clicks the delete button then the review will be deleted from the my reviews page and also from the database.

## **8. Agent Dashboard (Private Route and only the users who has the agent role will be able to see these routes):**

- When an agent clicks on the Dashboard, he/she will be redirected to a page where there will be the following routes:

- A. Agent Profile .
- B. My added properties.
- C. My sold properties.
- D. Requested properties.

**Agent Profile:** On this page there will be some information about the logged in user. The information must include user name, user image, role (if the user is a regular user then don't show anything). You can add other relevant information if you wish.

**My added properties:** On this page an agent will see all his/her added properties. each card will contain

- Property image.

- Property title.

- property location.

- Agent name.

- Agent image.

- verification status. (verification status will be modified by the admin. if the admin verifies the property then the verification status will be "verified" and if the admin rejects then the verification status will be "rejected")

- Price range.

- update button (if the verification status is "rejected" then the update button will disappear, which means the agent won't be able to update the property).

- > delete button

-Clicking on the `Update button` will redirect the user to a form page where the form will have the following fields:

- > Property image.

- > Property title.

- > property location.

- > Agent name. (readonly)

- > Agent email. (readonly)

- > Price range.

-Initially these fields will be filled up by previous values but the agent can modify these values. If the user clicks the submit button then the data will be updated in the database.

-clicking the delete button will remove the property from the "My added properties" and also from the database.

**My sold properties:** On this page all the sold properties of the specific agent will be shown. Only the properties for which the buyer has paid for will be shown here. Show these data in a tabular format. Each data will contain->

- property title.
- Property location.
- buyer email
- buyer name
- sold price.

**Requested/offered properties:** On this page all the offers which are made by the users for the properties that the agent has added will be shown. These data will be shown in a tabular format and each data will contain

- > property title.
- > Property location.
- > buyer email
- > buyer name
- > offered price.
- > accept button
- > reject button

-by default status will be pending. After clicking the accept button this status will be changed to "accepted" and both the accept and reject button will disappear and the status will be shown there. This will also happen for the reject button and "rejected" status will be shown.

-If the Agent accepts an offer for a specific property then other offers for that property will be rejected automatically and "rejected" status will be shown there.

## **9.Admin Dashboard (Private Route and only the users who has the admin role will be able to see these routes):**

-When an admin clicks on the Dashboard, he/she will be redirected to a page where there will be the following routes:

- A. Admin Profile.
- B. Manage Properties.
- C. Manage Users.
- D. Manage reviews.

**Admin Profile:** in this page there will be some information about the logged in user. The information must include user name, user image, role (if the user is a regular user then don't show anything). You can add other relevant information if you wish.

## Manage properties:

All the properties which the agents have added will be shown here in a tabular format. Each data will have

- property title.
- Property location.
- Agent name
- agent email
- price range.
- verify button
- reject button.

-if the admin clicks the verify button then that property will be added to the "All properties" page with the verified status also verify button and reject button will disappear and a status "verified" will be shown there. If the admin clicks the reject button then the property will not be added to the all properties page and "rejected" status will be shown in place of verify and reject button.

**Manage users:** Show all the users in a tabular form where each row will have:

- User name
- User email
- Make admin button
- Make agent button
- Mark as fraud button (This button will only appear if the user is an agent)
- Delete user

-The admin can make a user admin by clicking on the **Make Admin** button, admin can make a user "agent" by clicking the make agent button also the admin can delete a user from the database by clicking the delete user button.

-If the admin clicks the "Mark as fraud" button for any agent then that specific agent will be marked as fraud and a "Fraud" status will be shown in place of "Make admin", "Make agent" and "Mark as Fraud" button also all the properties that the agent has added will be removed from the "All properties" page and advertisement section on the homepage and that agent will not be able to add any properties in the future.

**Manage reviews:** All the user reviews for all the properties will be shown here. Each card will contain reviewer image, reviewer email, reviewer name, review and a delete button. If the admin clicks the delete button of any review card then that review will be removed from this page and also from the users page who has made this review.

10. You Must implement Email and password-based Authentication. This means you will have to implement the Registration and the login page. Users should be able to toggle between Login and Registration view.

> **Note:** Do not enforce the `forget or reset password feature` and the `email verification method`, as it will inconvenience the examiner. If you want, you can add email verification/forget the password after receiving the assignment result.

On the Registration page, display errors when:

The password

- is less than 6 characters
- don't have a capital letter
- don't have a special character

On the Login page, display errors when:

- password doesn't match
- email doesn't match

You can take the error message from Firebase. You can show the error below the input fields or via alert/toast. If you use alert/toast, do not implement the browser alert.

11. Use Tanstack query with mutation for data fetching and posting.

## You must do the following:

1. Once logged in, the user name, profile picture and the logout button should appear on the navbar. If the user clicks on the logout button, make sure to log him/her out.

2. Add a 404 page (not found page)

3. **\*\*Commits & readme:\*\***

- Minimum 20 meaningful git commits on the client-side repository.
- Minimum 10 meaningful commits on the server-side repository.
- Create a readme for client-side and write about your project (at least 5 bullet points). **\*\* Remember to add your client-side live link to your website here.\*\***

4. Also, implement at least `one extra login` which could be (facebook, github, google, etc).

5. After reloading the page of a private route, the user should not be redirected to the login page.



6. Make the website responsive. Make sure the site looks different on desktop and mobile responsive. Tablet responsiveness is optional.

## **Bonus Tasks**

1. Add a search functionality on the "All properties" page based on the property title.
2. Implement a sort functionality based on the price range on the "All properties" page.
3. **Implement JWT on login (Email/Password and social) and store the token.**
4. **Add a section on My sold properties page of the Agent dashboard where the total property sold amount of a specific agent will be shown.**
5. Add another route on the admin dashboard called "Advertise property". In this route all the admin verified properties will be shown in a tabular format. Each data will contain property image, property title, price range, Agent name "Advertise" and "Remove Advertise" button.

-If a property is already advertised then the "Advertise" button will be disabled and "Remove Advertise" button will be enabled.

-If a property is not already advertised then the "Remove Advertise" button will be disabled and "Advertise" button will be enabled.

-On clicking the "Advertise" button an admin can advertise a specific property. This property will be shown in the advertisement section on the homepage also the "Advertise" button will be disabled and "Remove Advertise" button will be enabled.

-On clicking the "Remove Advertise" button an admin can remove a property from the advertisement section on the homepage. After this "Remove Advertise" button will be disabled and "Advertise" button will be enabled.

-Admin can advertise at most 6 properties. If admin tries to advertise more properties then an error message will be shown.

## **Optional Tasks**

You have to implement **two tasks** from the following:

1. There will be an edit button on the profile page of any user (regular user, admin and agent). By clicking the edit button a modal will open where there will be a form that will have some input

fields to update that users information. Don't change user email because it may cause some problems in your project.

2. Add a button on the property details page called "Report this property". By clicking this button an user can report that specific property (a report may include reporter name, reporter email, report description. You can take these inputs through a modal). There will be a route on the admin dashboard called "Reported property" and all the reported properties will be shown here. Each report will have reporter name, reporter email, report property title, reported property agent name and report description. There will also be a "Remove this property" button. By clicking this button an admin can remove that property from "All properties page". Also the review for that property will be removed.
3. Add a route called "selling statistics" on the Agent dashboard where you have to visualize a chart about an agent's selling record. You can show the chart based on the property title and sold price. This chart can be of any type. [recharts](#).
4. Implement Axios interceptor.
5. Implement a price based filtering system on "All properties page".
6. Use swiper.js for any slider.
7. Use React hook form for any form related works.

## What to Submit

1. **Assignment Category:** assignment12\_category\_0010
2. **Admin email:**
3. **Admin password:**
4. **Agent email:**
5. **Agent password:**
6. **Front-end Live Site Link:**
7. **Client Side Github Repository Link:**
8. **Server Side Github Repository Link:**