Assignment - 9 Requirements

Task Description

Website Categories & Segments

You need to make a website dedicated to real estate, but not just any real estate! We're focusing on one specific type Here, we have given you a list of 8 different categories of real estate properties. Choose one of these categories from the following list to create your website. Your website should have at most one real estate property category type. For example, if you choose a Residential property category, your website should only have properties related to the Residential category; you must not add other categories like Commercial, Industrial etc.

- **Residential:** Single-family homes, townhouses, apartments, student housing, senior living communities, vacation rentals etc.
- **Commercial:** Office buildings, retail spaces, warehouses, industrial facilities, restaurants, entertainment venues etc.
- **Land:** agricultural land, ranches, farms, vineyards, forests, Plantations etc.
- **Industrial:** Warehouses, Factories, Production plants, Industrial parks, Cold storage facilities, Power plants, etc.
- **Speciality:** Hospitals, schools, religious facilities, transportation terminals, convention centers, art galleries etc.
- **Luxury:** Penthouse, Beachfront properties, resorts, private islands, Villas, mansions etc.
- **Government & Public:** Military bases, courthouses, libraries, public parks, and community centers etc.
- **Hospitality:** Hotels, motels, resorts, vacation rentals, Lodges, Guesthouses etc.

Main Requirements:

Key Rules:

- Include a minimum of 10 notable Github commits. Add a meaningful readme.md file with the name of your website and a live site URL. Include a minimum of five bullet points features and characteristics of your website. Add the npm package name which you have used for the challenges task.
- Make it responsive for all devices. You need to make it responsive for mobile, tablet and desktop views.
- Use environment variables for Firebase config keys. **(In the Assignment video you will see this point is on additional information but remember it is mandatory to implement)**

---

1. **Make sure your design is unique.** First, decide what kind of website you want to make. Then, search online or check out websites like ThemeForest to get ideas for the design. But remember, your website idea shouldn't be similar to any projects you've done before, and it shouldn't be like any examples shown in our modules or conceptual sessions.

★ You can also look for free resources on blogs to help with your website. https://bootcamp.uxdesign.cc/free-images-and-resources-collection-for-website-c77f2fc46ce5

2. Give a website name and it will show on the website title.

💡 **The home page will have these sections - Navbar, Slider, Estates, Footer & 2 extra section**

🎯🎯🎯 **Here Creating 2 Extra sections is fully optional.**🎯🎯🎯

3. **Navbar:** The home page will have a navbar and Implement active routes on the Navbar. The navbar will contain the Website name, Home, Update Profile (challenges requirements -3) and user profile.
    - But the user profile is conditional, If a user is logged in, the user image will show on the Navbar and When you hover over the user image it will show the user name.
    - If the user is not logged in it will show a login button.
    - So if a user is logged in you will show the user image and a logout button. If you click on the logout button make sure you have logged out.

4. **Login Page:** When you click the login button on the navbar it redirects to the login page. You have to use a password and email-based authentication to Login. The login page will have-
    - Email
    - Password
    - Google login,
    - GitHub/Facebook/Twitter login - implement any of one
    - A link that will redirect to the Register page

    💡**Here the email and password should match with the registered email and password. If it doesn't match, show an error. You can show an error by using toast/sweet alert if you want.**

5. **Register Page:** You have to use a password and email-based authentication to register. The Register page will have the following -
    - Name
    - Email
    - photoURL
    - password
    - A Link that will redirect to the login page

    ★ For password verification you need to follow this -

- Must have an Uppercase letter in the password
- Must have a Lowercase letter in the password
- Length must be at least 6 character

★ If any of this isn't fulfilled it will show an error/toast

★ After successful login or Register you need to show toast/sweet alert

💡Don't implement email verification or forget password method as it will inconvenience the examiner. If you want, you can add these after receiving the assignment result.

6. **Banner:** Add a slider (Daisy UI slider or Swiper slider) with a minimum of 3 slides.

7. **Footer:** A Footer with all relevant information and eye-catching design.

   💡**Make sure the Navbar and Footer are showing on all pages**

8. **Estate Section:** In the Estates section you will choose a category for example: "**Residential**" and for this category you need to create a JSON minimum of 4 and a maximum of 9 data. Each estate card will contain -
   - a relevant Image,
   - estate_title,
   - id,
   - segment_name
   - description,
   - price,
   - Status => It can be "sale", "rent"

- Area => for example- 2000 sq ft
- location
- facilities => in an array => [“living room”, “swimming pool”, “Kitchen”]
- and a “View Property” button.
- You can add other information for each data if you want

9. **Estate Details:** When the user clicks the “View Property” button it takes the user to the Estate Details page. This page will be a protected route and ensure that the private route moves to the login page if the user is not logged in.
    - **Banner:** you need to show detailed information about the selected estate.

💡**The user shouldn’t be redirected to the login page after you reload a page of the private route.**

10. **404 page:** Create a 404 page/ not found page
11. **Extra route:** Add 1 more extra route to your website. This route should be private/protected and the contents for that route should be meaningful and relevant.
12. **Dynamic Title:** You need to implement a dynamic title for each page.

13. Once a user is logged in, user information will show on the navbar even if he/she reloads the website, this info will not disappear. You can show a loader when the logged-in user info is in a loading state. You can implement this using the onAuthStateChanged method of Firebase.

Challenges Requirements:

1. Implement show and hide passwords on the registration page. By default it won't show the password when you click on the "Eye" icon it will show the password. In this way, you need to toggle.
2. Implement any of 3 packages from the following list:
   a. AOS Package,
   b. Animate.css,
   c. React-leaflet,
   d. React Hook form,
   e. Swiper slider

3. Create a protected route where, after logging in, users can see their information like- name, email, photoURL displayed in a form. They can edit name and photoURL and save the changes. This "saving" process uses Firebase's updateProfile() method to update the user's information stored securely in Firebase.

Additional Information:

- Create your JSON data and host the image on imgbb
- Deploy your website on Firebase.

You need To Submit:

- Your GitHub repository
- Your Live site link