

Task Description

Website Theme: Career Counseling Website

You are here to develop a robust career counseling website that provides services to users seeking career guidance.

For this project, will develop a **Career Counseling Website** designed to guide users on their career paths. The goal is to create a seamless, intuitive platform that offers personalized career advice, guidance, and resources to users seeking to make informed decisions about their professional futures. The website will be built using **React**, **React Router**, and **Firebase** for handling routing, user authentication, and data management.

Key Rules:

- **Github Commits:** 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.
- **Environment Variables:** Use environment variables for Firebase config keys.
- **Responsiveness:** Make it responsive for all devices. You must make it responsive for mobile, tablet, and desktop views.
- **Host your Application:** You can choose deployment systems like Netlify, Surge, and Firebase for hosting. As you are developing a single-page application
 - ensure that the page doesn't throw any error on reloading from any routes.
 - Add your domain for authorization to Firebase if you use Netlify / surge
 - Logged in User must not redirect to Login on reloading any private route
- **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 in our modules or conceptual sessions.

★ You can also look for free resources on [blogs](#) to help with your website.

Main Requirements:

Layout Structure

1. **Navbar:** The home page will have a navbar, and active routes will be implemented on the navbar. The navbar will contain the Website name, Home, and My Profile (challenges requirements 1).
 - 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.
2. **Main Section:** Main Section will show different pages based on routes.
3. **Footer:** A Footer with all relevant information and eye-catching design.

💡 **Make sure the Navbar and Footer are showing on all pages without an Error/404 page.**

JSON Data Generation:

- Create your JSON data and host the image on imgbb
 - JSON :
 - a relevant Image,
 - Service name -> ("Career Counseling Sessions", "Resume Review"), etc.
 - Category (service providing type ex: online/ offline/ group/)
 - Brief description
 - pricing

- Duration with date (for example: 05-12-24 5-6pm)
- Counselor
- Rating
- and a “Learn More” button.
- You can add other information for each data if you want

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

💡 The home page will have these sections Slider, Services, & 2 extra section

Home Page:

5. **Slider:** Add a slider (Daisy UI slider or Swiper slider) with a minimum of 3 slides.
6. **Services Section:** In the services section you need to create a JSON minimum of 6 data. You need to show some data on the card layout on the home page.

6.1. Each service card will contain - JSON Data

- a relevant Image,
- Service name
- Category
- pricing
- Counselor Name
- and a “Learn More” button.

(and other info what you want to show on the card is totally up to you)

7. Service Details Page: (Protected)

When the user clicks the “Learn More” button it takes the user to the Service 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. You need to show all the information about the selected service.

In this page there will be an input field and a “Comment/Feedback button”. Users can add any comment or give feedback regarding the service by

clicking on the button. Show submitted comments on this page.
You don't need to save comments using localStorage. Don't worry .

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

Authentication:

8. **Login Page:** When you click the login button on the navbar /protected routes it redirects to the login page.
You have to use a password and email-based authentication to log in. The login page will have-

- Email
- Password
- Google login,
- A link that will redirect to the Register page

9. **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
- Google Login

★ 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

⚠️ **Show all the error and success messages by using toast/sweet alert.**

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

Others:

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 and protected, and its contents 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 Firebase's `onAuthStateChanged` method.

Challenges Requirements:

1. **My Profile:** Create a protected route where, after logging in, users can see their information like- name, email, and photoURL and there will be a form with 2 input fields for name and photoURL. Users can edit the 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.
2. Implement show and hide passwords on the registration page. By default, it won't show the password, but when you click on the "Eye" icon, it will. In this way, you need to toggle.
3. Implement any of 1 packages from the following list:
 - a. [Swiper slider](#)
 - b. [Aos animation](#)

4. Forget Password

Make your forgot password feature functional. Clicking "forgot password" redirects the user to the forget password page, containing a form with an email field and a reset password button.

- Suppose a user enters an email address on the login page. Show the email also in the forget password form.
- On clicking the reset button. Redirect the user to Gmail.

You need To Submit:

⚠ Before you submit, check carefully-> you did the things mentioned in the '[Key Things You Must Do](#)'

- Assignment Category
- Your GitHub repository
- Your Live site link