

**Assignment\_ID:** assignment\_category\_0002

**Video Link ::** <https://www.youtube.com/watch?v=mkrckyVCVJQ>

**Changes on req 07:**

**Single Service details will be a private Route**

**No need to show Short description of the user.**

**Show Service Area Not location**

# Assignment 11 : THE CAREER MAKER

---

## Main Requirements

### 0. Focus on making the website visually appealing. Ensure that

- Color contrast is pleasing to the eye
- The website does not have a gobindo design
- The website has proper alignment and space`
- If needed, customize the design of any component you are taking from any component library. (For example, you are using daisy ui & have taken a card component from Daisy, if needed, customize the styling of the card to make it reasonable rather than just copy & paste it.)

*Note: Your website can not be related to your previous assignments' layout/ design or any practice project shown in the course modules or our conceptual sessions. Ex: You can't copy any design or similar functionality/ layout of*

- *Dragon news*
- *Espresso Emporium*
- *Genius Car*
- *Career Hub Website, any conceptual session projects, or projects showed in our course.*
- *Any of your previous assignments*

*If any similarities are found, you will get zero(0) as a penalty.*

---

## 1. Website Theme 🏆

You have to create a **offline service sharing web application**. Where users can do below functionalities.

1. On Home-Page user can watch available services and the information of service provider
2. Users can watch all services in the all services page.
3. Users can book services after reading the detailed information from the service detail page of a specific service.
4. Users can see their booked services on my schedule page
5. If a user want, he can add his own services from add a service page .
6. If other user booked his /her services then it will also show in my schedule page.
7. Users can delete his service / update his service .
8. If someone booked a service for the user , then the service owner can update the service status with a dropdown menu.

### Possible Pages and Routes

1. Home (/)
2. Services
3. Single Services
4. Add-A-Service Page
5. Manage-Services Page
6. My-schedule Page

### Service Ideas

- Carpooling and Ride-Sharing
- Tool Sharing
- Home Services Exchange
- Language Exchange
- Pet Sitting and Walking
- Book or Media Swap
- Childcare Co-op
- Clothing Swap
- Homegrown Produce Exchange
- Fitness and Sports Partners
- Ride-Share for Seniors
- Local Tours and Guide
- Community Repair Workshops
- Paste control

- Many More, any kind of unique service sharing idea will be accepted.

Note:: You can choose only a category. User can create services relevant to the category.

## 2.Layout

Make sure to keep the navbar and footer on all the pages including/excluding the 404-page.

Create a reasonable and meaningful footer. (including website logo+name , copyright, some contact information, social media links, address, etc.)

### Optional but recommended :

Try some other Tailwind CSS library for design like

[flowbite](#), [Mamba UI](#), [Preline](#),

Try to ignore daisy-UI

## 3.Navbar

Your website should have a navbar with the Website name+logo, You have to do some conditional rendering on navigation. Please follow the below requirement.

Every non-logged user will show following nav menu

Home , Services, log-in

Every logged in user will show following Nav Menu

Home , Services, logout, Dashboard.

The Dashboard is a dropdown menu. On Clicking dashboard user will show following menus, ( My-services , Add-services , My-schedules )

## 4. Login & Registration systems

On the Registration and Login pages, display relevant error messages when necessary.

**Login Page:** When a user clicks on the login button, they will be redirected to the login page having the following:

- Email/Password
- Google Sign-in
- A link that will redirect to the registration page

**Registration Page:** The Registration page will have the Email/Password form having the following fields:

- Name
- Email
- Password
- Photo URL

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

## 5.Home page:

**Banner/Slider:** Add a beautiful banner/ slider . Try to make it catchy and impressive.

**Popular Services :** On this section, you will have to show at least 4 services with one-column layout or 2 column layout having the following information:

- service Image
- Service Name
- Service Description ( max 100 character)
- View Detail Button
- Service Provider Image & Name
- Service Price

**a Show All button** that will redirect you to the all services page.

**Extra Section:** Add 3 relevant and unique extra sections in the homepage in addition to the Featured Section. Try to make them attractive.

Note: When a user is not logged in and if he/ she clicks on the View Details button, redirect him/ her to the login page. Without a login, you can not visit the single service details page.

### Extra section ideas

*Value Proposition or Unique Selling Points, Features or Benefits, Call to Action (CTA) , Customer Testimonials, Pricing or Plans, Product or Service Showcase, About Us, Company Information, Team or Founders' Profiles, FAQ Section, Blog or Content Highlights, Contact Information, Newsletter Signup, Social Media Links, Trust Symbols or Badges, Client Logos or Partnerships, Video Introduction, Portfolio or Case Studies, Countdown Timer (for limited-time offers), User-generated Content (e.g., reviews, user-submitted photos), Awards and Recognitions,*

*Resources or Downloadable Content, Customer Support Chat or Help Center, Test Drive or Demo Option, Security and Privacy Assurance, Exit-Intent Popup, Mobile App Download Links, Location or Service Area Map, Latest News or Updates and many more.....*

## 6. All Services Page

### **Filter Section:**

You have to implement search functionality by the service name.

### **Services Section:**

You have to show the services/filtered services in this section.

On this page, you will have to show at least 6 products with one-column layout having the following information:

- Service Image
- Service Name
- Service Description ( max 100 character)
- View Detail Button
- Service Provider Image , Name
- Service Area
- Service Price

There will be more buttons. On click the button will show all services and hide the button after loading all services.

## 7. Single Service details ( PRIVATE )

After clicking on the View Details button, he/ she will be redirected to the Service Details route ( /services/:id ) containing the information

### **Service Provider Information:**

- Service Provider Name
- Service Location

### **Single Service Section:**

- Service Image
- Service Name
- Service Description

- Service Provider Image & Name
- Service Price
- Book Now Button
- On clicking book now button a modal will open with following input field
  - Service Name( Not editable )
  - Service Image ( Not editable )
  - Service Provider email ( Not editable )
  - User email ( Not editable )
  - **Service Taking Date (editable)**
  - **Special instruction (editable )**  
// anything like address , area, customized service plan.
  - Price ( Not editable )
  - [Purchase this Service] Button

On clicking the Purchase Button will add the booking into the database.

## 8. Add a Services (PRIVATE)

Create an Add Product page where there will be a form having the following fields:

- Picture URL of the Service
- Service Name,
- Your name, (from Firebase user , not editable )
- Your email, (from Firebase user , not editable )
- Price,
- Service Area,
- Description
- Add button

## 9. Manage Services (PRIVATE)

This page will be private routes: If a user logs in, they will see the Manage Service page, which will show all the services they have added from the Add Product page.

Each card will have an Edit and delete button.

**Update Action** - If they click the edit button, they can update the service information

Note: you can show the update form in a modal or another route.

**Delete Action** - If they click the delete button, the service will be removed.  
Before the delete, ask for a delete confirmation.

Note: If a user logs in they will only see the products they have added. The user cannot see the products other users added.

## 10. My Schedules (PRIVATE)

The page will contain 2 section

### **My Bookings ::**

The services which you have booked will be loaded in this section. If there is no service ,show a relevant message.

### **My Pending works ::**

The services of yours which other users have booked will be loaded in this section. If there is no service ,show a relevant message.

There will be a dropdown button for each booked service card. The dropdown will have 3 options.

1. Pending (default)
2. In Progress
3. Completed

On changing the Dropdown from a specific booking service. It will update the service status of that service from the booking collection

## 11. Show The Toast

For all the CRUD operations, show relevant toast/ notification/ anything with a meaningful message.

## 12. 404 page:

Create a 404 page. Add any interesting jpg/ gif on the 404 page. add a jpg/ gif & a Back to home button. The Back to home button will redirect the user to the home page.

## 13. Environment Variable

Use the Environment variable to hide the Firebase config keys and MongoDB Credentials.

## 14. Explore New Packages

Implement any two packages from the following list & try to make your website more attractive:

- React- Leaflet
- React-simple-typewriter
- react-elastic-carousel
- React-counter
- Lottie-react
- Framer-motion

## Bonus Requirements

### 1. Commits & readme:

- Minimum 18 meaningful git commits on the client-side repository.
- Minimum 8 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.\*\***

### 2. Fix your Reload Issue

If you reload the protected/private routes (after login), this page will not redirect the user to the login page. Instead, it will keep the logged-in user on the protected route.

### 3. Make HomePage responsive

Make the Homepage of your website mobile, tablet & desktop responsive.

### 4. Show Other Services on Single Service Page

On the single service page you have to show the other service card of this service provider. (hint: You can filter from service collection by service-provider-email). If anyone clicks on any one of the other service cards of that service provider, the user will be redirected to the single service page showing the details of that service card. If there are no other services of that service provider then don't show the section.

**Note:** If you cannot exclude the service card you are currently viewing in the other service card section, you can show all the service cards including the one you are currently viewing.



## 5. Website Naming

Give your website name. The website title will be changed according to the route you are clicking. Suppose your website name is PHero. Then, on the 'All Products' route, your website title will be 'PHero | All Products'.

## 6. Do Some Security with JWT

Upon login, you will create a jwt token and store it on the client-side and you will send the token with the call and verify the user. Implementing 401 and 403 is optional.

Ensure you have implemented jwt token and create a token and store it on the client-side for both email/password-based authentication and social login. You must implement JWT on your private routes.

---

## Optional (But Highly Recommended):

Implement any two tasks from the following optional list:

1. Add a spinner when the data is in a loading state. You can add a gif/jpg, use any package or customize it using CSS.
2. Explore and implement any of the animations from the Framer Motion
3. Explore and implement Tanstack query mutations in your api.
4. Add one extra feature of your own. This will help you in the future to differentiate your project from others.

## Additional information:

1. You can host images anywhere.
2. You can use vanilla CSS or any library.
3. Try to host your site on Firebase (Netlify hosting will need some extra configurations)
  - [Firebase Hosting Setup Complete Issue](#)
4. Host your server-side application on Vercel. If needed, you can host somewhere else as well.
  - [How to deploy a Node/Express server using Vercel CLI](#)
  - [Some Common Vercel Errors](#)
5. Make Sure you deploy server-side and client-side on the first day. If you have any issues with hosting or GitHub push, please join the "Github and deploy"

related support session.

## **What to submit:**

- 1. Your client-side code GitHub repository**
- 2. Your server-side code GitHub repository**
- 3. Your live website link**

## **Some Guidelines:**

1. Do not waste much time on the website idea. Just spend 15-20 minutes deciding, find a sample website, and start working on it.
2. Do not waste much time finding the right image. You can always start with a simple idea. Make the website and then add different images.
3. Don't look at the overall task list. Just take one task at a time and do it. Once it's done, pick the next task. If you get stuck on a particular task, move on to the next Task.
4. Stay calm, think before coding, and work sequentially. You will make it.
5. Be strategic about the electricity issue.
6. use chat gpt to generate JSON data. You can use chatGPT for Other purposes as Well.