

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

**Updates:**

- **Update a word mistake in the website theme.**
- **You have to implement search in all search page. Not in navbar**

# Assignment 11

---

Welcome to Job assessment

We're currently in search of a talented junior MERN stack developer to join our team. As a junior member, you'll have the opportunity to work closely with our experienced developers on exciting projects utilizing MongoDB, Express.js, React, and Node.js. This role offers an excellent chance for growth, where you'll be able to enhance your skills while contributing to the development of innovative web applications. We're seeking someone with a solid foundation in web development and a passion for learning new technologies. If you're enthusiastic about building dynamic and user-friendly applications and eager to kick-start your career in the MERN stack environment, we'd love to hear from you. We Invited you to complete this job assessment to prove your skill and

## Read This Following details carefully

**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.)

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
- Projects shown in our course.
- Any of your previous assignments

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

## Let's understand the website theme

You have to create a **simple service-sharing web application**. Where users can add , update and delete his own services , watch others' services , watch service details, book any specific service , and update the current status of his service .

### Possible Pages and Routes

1. **Login:** Simple Login system with a form and social login (google only )
2. **Registration:** Simple Registration system with a form & social login (google only).
3. **Home:** This will be the main landing page and by clicking logo and Home you have to take users to this page.
4. **All Services:** This public Route will show all the services from your system service collection
5. **Single Services:** For specific service , It will show the service details with a book now button. Where users can book specific services. .
6. **Add-A-Service Page:** Users will successfully add service to the system from this page
7. **Manage-Services Page:** This Private Route will show all the services he has added in the system with 2 actionable buttons [ update and delete ] . he can update or delete his service from this route.
8. **Booked Service:** This Private Route will show the services you have booked from the system and current status of the services.
9. **Service To Do:** This Private Route will show all your services booked by other users. User can update the status of booked services with a dropdown. Status are
  - pending( default value )
  - working
  - completed

For This Assessment please do not add any validation for booking his own service. It will help us to check the assignment properly. After getting the result you can work on it.

### Service Ideas

You can select service ideas in different categories. Create the system based on a single category,

|                                  |                      |
|----------------------------------|----------------------|
| Consultation Services by Doctors | Home Repair Services |
|----------------------------------|----------------------|

|  |   |
|--|---|
| Consultation Services by Lawyers       | Electronic Item Repairing Services                            |
| Consultation Services by psychologists | Spiritual activity services<br>(milad, khatna, puja , Bibaho) |
| Educational Services by Teachers       | Beauty and Grooming Services                                  |
| Event Management Services              | Fitness and Therapeutic Services                              |

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

### Key Rules:

- Include a minimum of 18 notable GitHub commits on the client side
- Include a minimum of 8 notable GitHub commits on the server side
- Add a meaningful readme.md file with the name of your website and a live site URL. Include a minimum of five bullet points to feature your website. Write your selected category's name here.
- Make it responsive for all devices. You need to make it responsive for mobile, tablet and desktop views.
- After reloading the page of a private route, the user should not be redirected to the login page.
- Use the Environment variable to hide the Firebase config keys and MongoDB credentials.
- Don't use any Lorem ipsum text; you can not use the default alert to show any error or success message.

## Requirements



## 1. Layout

|                                  |
|----------------------------------|
| Header                           |
| Dynamic sections based on Routes |
| Footer                           |

- Create A Beautiful Header with a Navbar. Follow [2nd section](#) for more details about navbar.

- Create a meaningful footer. (including website logo+name, copyright information , social media links (facebook and Linked-In)

- create a 404 Error-Page also. It will show for invalid routes or errors with a button. By clicking the button it will take the user to the homepage.

## 2. Navbar Design

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 | Dashboard | Logout Button with user Image/Name |
|------|----------|-----------|------------------------------------|

**The Dashboard is a dropdown menu.** On Clicking dashboard user will show following Routes

- Add Service
- Manage Service
- Booked-Services
- Service-To-Do

### 3. 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 Based Login
- 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.

### 4. Add a Services (PRIVATE)

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

- Image URL of the Service
- Service Name,
- Price,
- Service Area,
- Description
- Add button

```
Service Information Dummy Structure
{
  imgUrl,
  serviceName,
  Price,
  serviceArea,
  description,
  providerEmail, (firebase -> user.email )
  providerImage, (firebase -> user.photoURL)
  providerName [ firebase -> user.displayName)
}
```

**The person adding the service is a service Provider .** You have to store service provider information with the service information.

Service provider image , Service provider name, Service provider email will be taken from firebase . On clicking the add service button service information will be added in the database.

## 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 maximum 6 service-card with one-column layout or 2 column layout having the following information:

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

The section will have 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.

## 6. All Services Page

### Services Section:

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

On this page, you will have to show all services 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

## 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 Provider Image
- Service Location/Area

### Single Service Section:

- Service Image
- Service Name
- Service Description
- Service Provider Image & Name
- Service Price
- Book Now Button
- On clicking the book now button, a modal will open with the following input fields or take the user to another route which will have a form with the following input field.
  - ServiceId (Not editable)
  - Service Name ( Not editable )
  - Service Image ( Not editable )
  - Provider email ( Not editable )
  - Provider Name (Not editable )
  - Current User email ( Not editable )
  - Current User Name( Not editable)
  - **Service Taking Date (editable)**
  - **Special instruction (editable )**  
// anything like address , area, customized service plan.
  - Price ( Not editable )
  - Purchase Button

On clicking the Purchase Button will add all the information with a serviceStatus (pending) into a new collection in the database.

#### Booking Information Dummy Structure

```
{
  serviceId, serviceName, serviceImage , providerEmail, providerName, price

  userEmail, userName , serviceDate, instructions,status
}
```

## 8. Manage Services (PRIVATE)

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

**Hint:** `providerEmail==user.email`

## 9. Booked Service (PRIVATE)

The page will contain The services from the booking collection which you have booked/purchased have to show in this section.

It means that show all the booked services from the booking collection where current userEmail is equal to userEmail in each document.

If there is no service, show a relevant message.

## 10. Dynamic Titling

Make your website title Dynamic. For every Route change, The Website Title will be changed based on that route.

# Challenges

### 1. Service To Do (PRIVATE)

The page will contain all the booked services information where the serviceProvider is you. That If there is no service, show a relevant message.

You have to show service status also.



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

- pending( default value )
- working
- completed

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

## 2. Search System

Create a Search System based on the service name at all service pages .

- When a user writes anything. It will show you all the services matched with the search text.
  - On clicking the detail button of any service will take you to the service detail page.
- For example , you may take inspiration from the facebook search system.

## 3. Theme Customization

Add a Theme Toggling Button for changing themes from light to dark / dark to light. By changing the theme, it will make the full system dark / light based on user interaction.

### What to submit:

1. Your Assignment ID/Variant
2. Your client-side code GitHub repository
3. Your server-side code GitHub repository
4. Your live website link

## 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. Add one extra feature of your own. This will help you in the future to differentiate your project from others.
4. Implement Pagination in the all services page. Show 6-9 services per page.

## 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.

## 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.