

Task description

[1 update](search the word update)

Main Requirements

Create a website that shows the recipes of all available chefs. The website must be exclusively dedicated to one

cuisine(Bangladeshi, Chinese, Indian, Mediterranean, Thai, Japanese, Italian, African, Vietnamese, Srilankan

, Nepalese, American, Korean, Mexican, Australian, Afgan, French, etc.). For example, if you give Chinese Chef

and Chinese food recipes, it will only have Chinese Chef and Chinese food recipes. You cannot add recipes from

other countries—only one country chef and recipe.

1. Make sure your design is unique.

> First, select the cuisine type. Google the site design or visit ThemeForest to get your website idea.

However, your website cannot relate to your previous assignments or any demo project displayed in the course

or our conceptual sessions. (=====okey=====)

2. Give your website a name. The name should appear on the website and be displayed on the website's title.(=====okey=====)

2. Your website should have a navbar.

> with the **Website name, Home, Blog** and **User profile picture**. The **user profile picture** on the

navbar is conditional. If the user is signed in, the navbar will show the profile picture; otherwise, it will

show the **Login button**. **If the user name is available, the user's name will be visible when the mouse hovers**

over the profile picture.* (=====okey=====)

3. Create a reasonable and meaningful footer.

> Make sure that the navbar and footer are present on all pages: okey=====

3. When a user clicks on the login button, they will be redirected to the login page having the following:

- Email/Password

- Google Sign-in

- GitHub Sign-in

- A link that will redirect to the registration page (=====okey=====)

1. The Registration page will have the Email/Password form having the following fields:

- Name

- Email


- Password

- Photo URL (=====okey=====)

> **Note:** Do not enforce the email verification method, as it will inconvenience the examiner. If you want,

you can add email verification after receiving the assignment result.

1. On the Registration page, display errors when:

>  update: If a user's email address or password doesn't match, display error. This error will be displayed

on the Login page.

- The password is less than 6 characters
- A user cannot submit empty email and password fields (=====ok=====)

6. On the Homepage: There will be a banner section. You can make it a reasonable banner. That be text on one side,

and picture in another side. Or it could be text with background pictures. It could be a slider or a carousel.

You can make it the way you want.(=====ok=====)

7. The home page will have a Chef section with at least six cards. Each card will have the following information:

- Chef Picture
- Chef Name
- Years of experience
- Numbers of recipes

- Likes

- View Recipes Button (=====ok=====)

1. Add at least **two** extra sections on the home page. Make sure the sections are meaningful for the website

you are building. (=====okey=====)

1. When a user clicks on a chef's **View Recipes** button, he/she will be redirected to the **Chef Recipes** page.

The **Chef Recipes** page will have the following :

- **Banner** - chef picture, chef name, a short bio/description, likes, number of recipes, and years of experience

. (=====okey=====)

- Think about a section displaying the information of the selected chef's recipes in a tabular form or in a card group.

- > Each recipe info should contain the recipe name, ingredients (at least 5), cooking method, rating, and a

Favorite button. **Show at least 3 recipes.** for each chef. **If needed, you can have duplicate recipes for each chef**

(=====ok=====)

1. When clicking the Favorite button, show a toast message showing the recipe is your favorite and disable the button.

(=====ok=====)

2. Show a spinner when the data is in a loading state (=====ok=====)

2. Create a 404 page. Add a picture on the 404 page.(=====ok=====)

3. Use the Environment variable to hide the Firebase config keys. (=====ok=====)

4. Add "active route" to indicate the route which you are visiting.(=====ok=====)

Bonus

1. Minimum 12 meaningful git commits on the client-side repository and minimum of 5 meaningful commits on the

server-side repository. (=====okey=====)

1. Create a readme for **client-side** and write about your project (at least 5 bullet points) .Do not forget

to add your live site link of your website here.

2. Make your website home page mobile & desktop responsive (tablet responsive is optional)

(=====okey=====)

3. Create a `blogs` page route where you will have to answer the following questions

(=====okey)

1. Tell us the differences between uncontrolled and controlled components.

2. How to validate React props using PropTypes

3. Tell us the difference between nodejs and express js.

4. What is a custom hook, and why will you create a custom hook?

1. The Chef Recipe page will be a private/protected page. If you reload the protected/private route (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.

(-----okey-----)

2. Apply lazy load to the chef picture either on the banner of `Chef Recipes` or on the chef pictures of the six cards

on the Homepage.()=====

> Hint: Google `React Awesome components`, go to the GitHub repo, find `react-lazy-load`, and try it. Alternatively,

you can try any other package as well. (=====okey=====)

2. FUN: The blog page, Create a heading and add an icon/ button. When you click on that button, it will create a pdf

and will be downloaded (This will give you some fun. Try out this after completing all of your tasks.

Hint: Explore

`react-to-pdf` package). Your pdf should contain some information about the blog page.(=====)

2. Make sure your site looks reasonable. The design and color selection is meaningful.(=====okey=====)

3. Clean and organized Code (folder structure). Organize components with meaningful names, and add comments when needed

.(=====okey=====)

Optional (But Highly Recommended):

1. When clicking on the Favorite button, store the recipes in the Local Storage.

> Create a My Favorite Recipes page where you will show all your favorite recipes. Data will be in tabular form,

having a **recipe picture**, **chef name**, **recipe name**, and a **remove button**. The recipe will be removed from

the table by clicking on the Remove button.

2. When a user clicks on their profile image, it will redirect to the user details page. This page will show the user's

detailed information (name, email & profile picture). The logged-in user can update their profile(image, displayName)

by clicking on a button. (Alternatively, The profile edit form could be in a modal)

1. Add a link for Password reset (use toast/ alert), (Don't worry if the email goes to the spam folder.)

1. Explore & use these react packages in your website : react-icons, react-image-magnify, react-awesome-slider, react-loader etc.

1. Add an about us page

1. If you have time, try implementing a dark/light theme toggle.

3. Add something extra of your own. This will help you in the future to differentiate your project from others.

4. If you want to take a challenge. Consider using `react-leaflet` anywhere in your application

Additional information:

1. You can use a local image or host image anywhere or use pictures from the internet. And it's ok if you have the

image url, but the image link doesn't work.

2. You can use vanilla CSS or any CSS library (Bootstrap, tailwind) you want. If you wish, you can use both Bootstrap

and react-bootstrap. Also, if you want, you can use any tailwind component library such as DaisyUI, etc.

3. Try to host your site on Firebase (Netlify hosting will need some extra configurations)

4. Host your server-side application on Vercel. If needed, you can host somewhere else as well.

4. 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 `chatgpt` to generate JSON data. You can use chatGPT for Yother purposes as well.

No Pain, No Gain:

`The best things of life are on the other side of the pain.`