# Assignment 11 Requirements

---

## Video Link:

https://drive.google.com/file/d/1Z3NnGTbwmCBwNZWigrpDhkpYMdFQlR7-/view?usp=sharing

**Assignment_ID:** assignment_category_0007

## 1. Website Theme 🎖️

1. **Project Description: Community Food Sharing and Surplus Reduction Platform**

   Our project focuses on building a Community Food Sharing Platform, and we're seeking a front-end web developer. Using technologies like React, Firebase, Node.js (Express js) , and MongoDB, we aim to connect those with surplus food to those in need, reducing food waste and addressing hunger. Join us to create a user-friendly, socially impactful platform that promotes sustainability and community cooperation.

## 2.Layout

Ensure that the navbar and footer are present on all pages except for the 404-page. Create a meaningful and informative footer that includes the website logo and name, copyright notice, contact information, social media links, address, and more.

**Optional but recommended :**

Try some other Tailwind CSS library for design like

**flowbite** , **Mamba UI**, **Preline**,Chakra UI ,

## 3.Navbar

The navbar will have a website name with logo, Home, Available Foods, Add Food, Manage My Foods and My Food Request, Login/Signup. Your website will have these routes. Except for the ***Home*** ,***Available Foods*** and ***Login/Signup*** other routes will be private.

## 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 /Github 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.

**Featured Foods :** In this section, you will have to show at least 6 Food items.
the following information:

- Food Image
- Food Name
- Donator Image & Name
- Food Quantity (no. of person to be served.)
- Pickup Location
- Expired Date/Time
- Additional Notes
- View Detail Button

**Note :** Featured Foods represent the highest quantity of food options available.

**A Show All button** that will redirect you to the Available Foods page.

**Extra Section:** Add 2 relevant 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 food details page.

## 6. Available Foods Page

**Filter Section:**
You have to implement search functionality by the Food name.

**Sorting Section:**
You have to implement Sorting functionality by the Food Expire Date.

**Foods Section:**
You have to show all the available foods in this section having the following information:

- Food Image

- Food Name
- Donator Image & Name
- Food Quantity
- Pickup Location
- Expired Date/Time
- Additional Notes
- View Details Button

Note: If you apply any filter or search parameters then show filteredFoods/Sorted Foods instead of all foods.
On clicking the View Details button will redirect to the Food details page.

## 7. Single Food details

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

**Donor Information:**

- Donar Name
- Food Pickup Location

**Single Food Section:**

- Food Image
- Food Name
- Food Quantity
- Expired Date/Time
- Request Button
- On clicking Request Button button a modal will open with following input field
    - Food Name( Not editable )
    - Food Image ( Not editable )
    - Food Id (Not editable)
    - Food Donator email ( Not editable )
    - Food Donator Name(not editable)
    - User email ( LoggedIn user , Not editable )
    - Request Date(current time not editable)

- Pickup Location(not editable)
- Expire Date(not editable)
- **Additional Notes** (editable
- **Donation Money**(editable)
- Request Button

On clicking the Request Button will add the requested food into the database food request collection.

## 8. Add a Food (PRIVATE)

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

- Food Name
- Food Image
- Food Quantity
- Pickup Location
- Expired Date/Time
- Additional Notes
- Add Button
- Donator Image , Name, & email (From logged in user)
- Food Status (By default keep it ”available”)

Clicking on the add button the data will be saved on a collection (food collection) and the added food will be shown in the Available Food Page.

## 9. Manage My Foods (PRIVATE)

This page will be private routes:  If a user logs in, they will see the Manage Food page, which will show all the foods in tabular format they have added from the Add Food page.  (React-Table - must use)
Each card will have an Edit  and delete button.

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

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

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

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

**Manage Button:** If the user clicks the manage button it will redirect to manage a single food Page (/manage/:id).

## 10. Manage Single Food (PRIVATE)

The page will contain the food request information. A Food doner can update the request status from pending to delivered.
***Clue:*** *Maintain a different collection for food requests. Fetch the requests for a specific food by querying with the food id/donator info/relevant info from this collection.*

This section will contain the following informations:

- Requester Name
- Requester Image
- Requester Email
- Request Time and Date
- Status ( there will be a button to change the status to Delivered)

**If the status is changed to “Delivered” then no user will be able to see/request for this food again.**

## 11. My Food Request (PRIVATE)

The page will contain All Food Requests made by the logged in user. Every request will have the following information.

- Donar Name
- Pickup Location
- Expire Date

- Request Date
- Your Donation Amount
- Status (available/delivered)
- Cancel Request Button**.**

OnClicking the cancel request button, the request will be removed from the food request collection if the status is “available”.

## 11. Show The Toast:

For all the CRUD operations, show relevant toast/ notification/ anything with a meaningful message.Don’t use Browser Alert!

## 12. 404 page:

Create a 404 page. Add any interesting jpg/ gif on the 404 page. Do not add header & footer on this page. Just add a jpg/ gif & a Back to home button. TheBack 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 of the following packages:

- react-elastic-carousel
- Lottie-react
- Framer-motion
- React Hook Form
- React Helmet

# 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. 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 Foods route, your website title will be 'PHero | All Foods.

## 5. 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.
5.Use Axios Custom Hook. [AxiosSecure]

## 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.