Assignment Category: Tulip

🚩🚩update on main requirement 6: By Clicking on the “See More” button it will redirect you to the details page.

Update on main requirement 12: Clicking the See More button will redirect you to the Details Page.

Crowdcube: A Crowd Funding Application

Project Theme

A crowdfunding website is a platform where people can raise money for different projects, ideas, or causes by inviting others to contribute financially. These projects can include personal needs (like medical expenses), creative ideas (like making a film or app), and startups (like launching a new product).

Key Rules:

- **GitHub Commits:**
  - Include a minimum of 15 notable GitHub commits on the client side.
  - Include a minimum of 8 notable GitHub commits on the server side
- **Readme.md:** 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.
- **Responsiveness:** Make it responsive for all devices, including mobile, tablet, and desktop views.
- **Environment Variables:** The Environment variable hides the Firebase config keys and Mongodb credentials.
- **Lorem Text:** Don’t use any Lorem ipsum text; you can not use the default alert to show any error or success message.

- **Unique Design:** Create a unique Design but remember, your website idea shouldn't be similar to any projects you've done before or to any examples in our modules or conceptual sessions.

★ You can also look for free resources on blogs to help with your website.

- **Host your Application:** You can choose deployment systems like Netlify, Surge, and Firebase for client-side hosting and vercel for server-side hosting. As you develop a single-page application
    - Ensure the page doesn't throw any error when 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

---

Main Requirements:

- ### Layout Structure

1. **Navbar:** The navbar will contain the
    - Website name/logo,
    - Home,
    - All Campaign,
    - Add New Campaign**(It will be a private/protected route)**,
    - My Campaign**(It will be a private/protected route)**,
    - My Donations**(It will be a private/protected route)**,
    - "Log in", and "Register" **(Conditional)**
        - The "Login" and "Register" buttons is conditional, if the user is not logged in it will show the "Login" and "Register" Button. If a user clicks on "Login" it will redirect to the login page. If a user clicks on "Register" it will redirect to the Register page.
        - If the user is logged in here you will show the user photoURL, when you hover over the image it will show the displayName. And it will show the "Log out" button.

2. **Main Section:** Main Section will show different pages based on routes.
3. **Footer:** A Footer with all relevant information and eye-catching design.

- **Home Page:**

4. **Home Page:** Besides the navbar and footer the **Home Page** will contain a banner/Slider, a Running Campaign Section, 2 Extra Sections,
🎯 **Make sure to keep the navbar and footer on all the pages except the 404 page.**

5. **Banner:** Add a slider (you can use any type of slider/carousel) with a minimum of 3 slides and meaningful information

6. **Running Campaign Section:** You need to show at least 6 cards of Running Campaign. Running campaigns are the ones where the deadline has **not passed** yet. These campaigns are still active, and people can contribute to them. Here you will show the data from the database. Which data you want to show on the card is up to you. Each card will contain a "See More" button.

   update: By Clicking on the "See More" button it will redirect you to the details page.

   For showing 6 data you can use the limit operator of MongoDB

7. **Extra Section:** Add 2 relevant and meaningful extra sections that are mentioned above on the Home page.

Authentication

8. **Login Page:** When you click the login button on the navbar it redirects to the login page. You have to use a password and email-based authentication to log in. The login page will have-
   a. Email
   b. Password
   c. Google login, GitHub - implement any of one
   d. A link that will redirect to the Register page

🎯**Here the email and password should match with the registered email and password. If it doesn't match, show an error. You can show an error by using toast/sweet alert if you want.**

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
   ★ 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
   ★ If any of this isn't fulfilled it will show an error/toast
   ★ After successful login or Register you need to show toast/sweet alert

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

---

10. **Add New Campaign Page (/addCampaign)**
    Create an Add New Campaign page where there will be a form for the user to add a Campaign. The form will have:
    ○ image/thumbnail (use image URL)
    ○ Campaign title(string)
    ○ Campaign type (personal issue, startup, business, creative ideas) you can take it as a dropdown list.
    ○ description
    ○ Minimum donation amount (number)

- ○ Deadline (Date)
- ○ User Email (Read Only/ you can not edit this field)
- ○ User Name (Read Only/ you can not edit this field)
- ○ “Add” button

This will be a private/protected route.
When you fill in the data and submit the “Add” button, these data will be stored in your database and you will show a success message through toast/sweet alert.

11. **Details page (/campaign/:id)**
Details route will be a private/protected route. Please make sure that if the user is not logged in, the private route redirects to the login page. On this page, you will show all the information you have stored in the database. There will be a “Donate” button on this page, after clicking on the “Donate” Button the campaign data (with logged-in email and username) will be stored on the database in the **donated collection** with the user email and username.

12. **All Campaign (/campaigns)**
Create an All Campaign page where you will see the campaigns all the users have added to the database. You will get the data from the database and show it here in the table. Which property you want to show for each campaign is up to you. There will be a See More button on each table row.

Update: Clicking the See More button will redirect you to the Details Page.

13. **My Campaign (/myCampaign)**
It will be a private/protected route. On this page, a user can see all the Campaigns he/she has added to the database. here a user can only see his/her added data, but he/she can not access other’s data. Here you need to show the data in table Format. Each row of the table will contain these information-
- ○ 3-4 information about the campaign
- ○ “Update” Button

  - ○ “Delete” Button

14. **Update Campaign Page (/updateCampaign/:id)**

    It will be a private/protected route. When a user clicks on the “Update Button” it will take the user to the Update page, where there will be a form for the user to update a Campaign. The form will have the following:

    - ○ All the fields you have taken in the Add New Campaign Page (But user name & user email will be read-only)
    - ○ “Update” button

This will be a private/protected route. When you fill in the data and submit the “Update” button, these data will be updated to the previous data in your database and you will show a success message through toast/sweet alert.

🎯🎯**(Optional):** If you don’t want to create an Update page, you can also use a **modal** to update your data. For this when you click on the “Update” button it will open a modal but make sure you are logged in before updating the data.

15. **Delete:**

    If the user clicks the delete button, the Campaigns will be removed from the data. Before the delete, ask for a delete confirmation.

16. **My Donations (/myDonations)**

    It will be a private/protected route. On this page, a user can see all the Donated Campaigns he/she has added to the database. You need to show all the data in Card format. here a user can only see his/her added data, but he/she can not access other’s data. Which data you want to show is your choice.

17. Other Requirements

- **404 page:** Add a 404 page/Not Found Page
- **Loading Spinner:** Show a loading spinner when the data is in a loading state.

Challenges

- Implement a dark/light theme toggle for the home page.
- Explore this package and implement at least 2-
  - Lottie React
  - React-simple-typewriter
  - React Awesome reveal
  - React-tooltip
- At the top of the **All Campaign** page, there will be a button where you will implement sort functionality based on "Minimum donation amount". You can sort by ascending or descending or, it's up to you.
- In the Details page if the deadline is over you can't add the campaign to the donated collection rather you will show a message/toast/modal.

What to submit:

- Your client-side code GitHub repository
- Your server-side code GitHub repository
- Your live website link