

Video Guide : [Link](#)

Youtube Video Guide : [Link](#)

Blog Website

Assignment_ID: assignment_category_0006

Our company focuses on a Blog website. Currently, we are looking for a front-end web developer to build a blog website using React, firebase, react router, node, express, mongodb etc.

We have a list of tasks for you.

Main Requirements

Home Page:

1. Home page will have a header (simple navbar), banner (hero section), recent blog posts section, newsletter section, and footer
2. Navbar will have the following links:
 - a. Home
 - b. Add Blog
 - c. All blogs
 - d. Featured Blogs
 - e. Wishlist
3. If the user is not logged in, the Login and Register button should show, and if the user is logged in show his profile picture and logout button.
4. You will have to show six blogs in the recent blogs section. Each blog should have a title, image, short description, category, details button and wishlist button. Clicking the wishlist button will add the blog on wishlist page
 - a. ☐ Clue: While inserting new blog into the database include the current time, so you can sort blogs by date or time

5. In the newsletter section if a user input an email and submit, then the user will receive a toast message. Message could be "Thank you for subscribing to our newsletter" or relevant message. You don't need to send emails.
6. Add two extra sections on the home page with the five sections mentioned above.
 - a. ☐ Tips: This is the place to shine. Add something different and unique to make your website look different than others. And Make them relevant to your website.

All Blogs Page:

1. All blogs pages will have all the blogs added by the users.
2. Blogs can be filtered by category and add a search field to search blogs by blog title
3. Each blog should have a title, image, short description, category, details button and wishlist button.
4. Clicking the wishlist button will add the blog to the logged in user's wishlist and when the user navigates to the wishlist page he/she can see the blogs they wishlisted. ☐
Tips: Maintain a separate database collection for wishlist.
5. Clicking the details button user will navigate to blog details page

Blog Details Page:

1. Blog Details page will have all the information about the blog like title, image, short description, long description
2. Under the blog details there will be a textarea.
3. Here users can comment and all the comments with comment owner name and profile picture will be shown in the comment section. (Clue : Create a comment collection.)
4. While a user comment in a blog keep blogs _id, user name, user profile picture with each comments data. So you can filter the comments by blogs _id from the backend.
5. Users cannot comment on his/her own blog. show "Can not comment on own blog" or relevant text in comment section instead of a textarea if a blog writer is current user (Clue : Compare blog email with firebase user email and conditionally render the textarea)
6. There will be a conditional update button . If the current user is a blog owner then show the update button. Clicking the update button will navigate a dynamic route where blog details can be updated. (Clue : Same as previous clue)

Add Blog Page:

1. Create a form which will take blog information like title, image url, category, short description, long description and a submit button.
2. category must be a dropdown element (use `<select></select>` tag)

Update Blog Page:

1. Logged in users can update/edit the blogs they added.
2. This page will be a private/protected route

Featured Blogs Page:

1. This page will have a table including top 10 posts .
2. Top posts will be calculated based on the word count of the long_description of a specific blog.
Clue: Check the length.
3. Each table of row will include Serial Number, Blog Title, Blog Owner, and Blog owner Profile Picture
4. This table won't be a normal table. You have to use **any one** of the following table libraries to visualise.
 - a. [Ka-table](#)
 - b. [React-data-table](#)
 - c. [Mui-datatables](#)
 - d. [react-table-library](#)

Wishlist Page:

1. In this page show all the blogs wishlisted by a specific user.
2. Each blog should have a title, image, short description, category, details button and remove wishlist button
3. Clicking on remove wishlist will remove the blog from his/her wishlist

Authentication:

1. You Must implement Email and password based Authentication. This means, you will have to implement the Registration and the login page.
2. Also, implement at least one extra login which could be (facebook, github, google, etc).
3. Add a 404 page (not found page)
4. On the Login page, display errors when email password authentication fails
5. On the Registration page, display errors when the password :
 - a. is less than 6 characters
 - b. don't have a capital letter
 - c. don't have a special character
 - d. don't have a numeric character

Bonus Requirements:

Git Commits and Readme:

1. Create a meaningful readme for your project, including at least 5 bullet points explaining your project. Also, ensure you have:

- a. A minimum of 20 meaningful git commits on the client-side repository.
- b. A minimum of 10 meaningful commits on the server-side repository.
- c. Live site url and name on readme

Reload Functionality:

1. Implement a feature where reloading protected/private routes (after login) does not redirect the user to the login page. Instead, keep the logged-in user on the protected route.

Responsive Design:

1. Make the homepage of your website mobile and desktop responsive.
2. Tablet responsiveness is optional.

Tanstack query:

1. Instead of loader and useeffect use tenstack query for data fetching on all over the pages on your projects

Framer Motion Package:

1. Use the [Framer Motion](#) on the home page.
2. Implement it at least in one place on the home page. Using Framer Motion on other pages is optional.

JWT Authentication:

1. Implement JWT authentication on all the private routes.

Toasts:

1. Use toasts or sweetalert on all the crud operation success message and login / signup errors.

Website title:

1. Use your website name on website title with a relevent favicon

Optional (But Highly Recommended)

You must have to Implement any two tasks from the following optional list:

Loading Skeleton:

1. Instead of using loading spinners use loading skeletons while data fetching
 - a. Clue :You can explore [react-loading-skeleton](#)

Photo View:

1. On clicking the image of the blog the image will be in full screen preview mode.
 - a. Clue: You can explore [react-photo-view](#)

React Intersection Observer:

1. Try to implement an animation effect on a specific section if the specific section is in the viewpoint of the user. You can combine framer motion with this to implement animation (Clue : You can explore [react-intersection-observer](#))

Component library:

1. Use any component library without daisy ui . Here are some suggestions
 - a. [Chakra-ui](#)
 - b. [ant design](#)
 - c. [flowbite react](#)
 - d. [Rsuitejs](#)
 - e. [material ui](#)

Additional Information

1. You can host images anywhere.
2. Consider hosting your site on Vercel. Netlify hosting may require additional configurations.
 - a. info : Firebase recently deleting websites Host your server-side application on Vercel. If necessary, you can host it elsewhere
3. Deploy both the server-side and client-side applications on the first day to address any hosting or GitHub push issues promptly.

Follow these guidelines to complete the project successfully:

1. Avoid spending too much time on the website's concept; spend 15-20 minutes deciding on an idea and start working on it.
2. Don't waste excessive time finding the perfect images. Start with a simple concept and add different images later.
3. Focus on one task at a time and complete it before moving on to the next one. If you encounter difficulties, proceed to the next task.
4. Think before coding and work sequentially.
5. Prepare for potential electricity interruptions.
6. Utilise chat gpt to get help on project-related tasks.
7. Remember, creating a meaningful and user-friendly blogging platform is the primary goal, so prioritise these core features while also exploring optional enhancements to make your project stand out.