Blog Website

Assignment_ID: assignment_category_0006

Are you a front-end developer who thrives on building beautiful and functional web experiences? Do you enjoy clean code and keeping up with the latest tech trends? We're looking for you! We're building the next generation of our blog website and need a talented developer to join our team. You'll collaborate with designers and back-end developers to bring mockups to life using React, Firebase, and other cutting-edge tools. You'll be responsible for crafting a user-friendly and engaging experience for our audience. This includes building dynamic web pages, ensuring smooth navigation, and optimising website performance for all devices.

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.
- 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.

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 buttons 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. And each blog should have
    a. Title
    b. Image
    c. short description
    d. Category
    e. Details button
    f. Wishlist button
5. Clicking the wishlist button will add the blog to the currently logged-in user's wishlist. **[This is a challenge requirement and hence, not included in main marks. Check the challenge requirement for more info]**
    a. 💡 Clue: While inserting a new blog into the database include the current time, so you can sort blogs by date or time
6. In the newsletter section if a user inputs an email and submits it, then the user will receive a toast message. Message could be "Thank you for subscribing to our newsletter" or a relevant message. You don't need to send emails.
7. 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 from others. And Make them relevant to your website.

All Blogs Page:

1. All blogs pages will have all the blogs added by any user.
2. Blogs can be filtered by category and add a search field to search blogs by blog title. 💡Tips: Use MongoDB text search, follow the doc: https://www.mongodb.com/docs/manual/reference/operator/query/text/
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 collection in the same database for wishlist.
5. Clicking the details button user will navigate to the 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 the comment owner's name and profile picture will be shown in the comment section. (Clue: Create a comment collection in the same database.
4. While a user comments on a blog, keep the blog's _id, user name, and user profile picture with each comment's data. So you can filter the comments by blogs _id from the backend while showing the comments of that specific blog on this details page.

5. Users cannot comment on his/her own blog. show "Can not comment on own blog" or relevant text in the comment section instead of a textarea if a blog writer is a

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 the previous clue). Description of the update route is described below.

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 or any package)

Update Blog Page:

1. Logged-in users can update/edit the blogs they added.
2. When the user opens this page information of the editing blog will be automatically filled on the form, so that the user can only change what he actually needs to change rather than writing the whole form again
3. This page will be a private/protected route

Featured Blogs Page:

1. This page will have a table including the 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

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

Challenge Requirements:

Wishlist:

1. Blogs on the homepage will have a wishlist button and clicking on the wishlist button will add the blog to the currently logged-in user's wishlist.
2. Maintain a separate collection on the database for storing wishlist items. Don't forget to include the user's email or id, in order to filter the wishlist on the Wishlist page
3. User server-side filter query to filter wishlist for currently logged-in user

Datatables:

1. The table on the featured blogs page will not be a normal table, implement any of the following library to create a data table, where each column of the table is sortable by clicking on the column header
   a. Ka-table
   b. Tanstack-Table (recommended)
   c. React-data-table
   d. Mui-datatables
   e. react-table-library

Tanstack query:

1. Instead of loader and use effect 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.

Optional (But Highly Recommended):

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 from 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.

What to submit:

- Your assignment ID/variant (**assignment_category_0006**)
- Your client-side code GitHub repository
- Your server-side code GitHub repository
- Your live website link