Introduction

Background pattern

Description automatically generated

Assignment 2 – Front End

<<Segmentation Fault – Dan Cristian Deac>>

Graphical user interface, text, application

Description automatically generated

The documentation covers the front-end part of the Software Design project named “Segmentation Fault”, which is a Stack Overflow inspired website for asking and answering questions. This is my first time using React to develop a web front, and first-time doing front end for the matter. I want to make something that not only looks good but is intuitive to use and easy to understand. I have not finished yet implementing everything because I need to tie everything to the backend to test some features and create them.

## Tech Stack

This design was realized using **React** with typescript. In addition, I used a framework called **ChakraUI** for more components and help with design. It is a mobile first tool so It ensures the app can be perfectly fit for mobile screens. For routing I use next/router and Recoil module and for quick testing I used the **Firebase database**. IDE: VSCODE

## Architecture

Graphical user interface, text, application

Description automatically generated with medium confidence

**src/atoms**: This folder contains atomic or reusable components that are not specific to any particular feature or page of the application. The AuthModalAtom.ts file contains the implementation for an authentication modal that can be used throughout the application, and the questionAtom.ts file contains the implementation for a question component that can be reused in different parts of the application.

**src/chakra**: This folder contains files related to the Chakra UI library, which is a popular UI library for React. The buttons.ts file contains custom button styles and configurations for the application, while the theme.ts file contains custom theme variables for the Chakra UI library.

**src/components**: This folder contains all the reusable components that are specific to a particular feature or page of the application. The components are further organized into subfolders, such as layout, modal, and Questions. **moda**l folder contains components that display modal dialogs, such as an authentication or confirmation modal.

**src/firebase**: This folder contains files related to the Firebase backend service. The clientApp.ts file contains the configuration for the Firebase client app, while the error.ts file contains custom error handling logic for Firebase errors.

**src/hooks**: This folder contains custom hooks that can be used throughout the application. The useQuestions.ts file contains a custom hook for fetching and managing questions from the Firebase backend service.

**src/pages**: This folder contains all the top-level pages of the application. The answers folder contains the implementation for a page that displays answers to questions, while the index.tsx and app.tsx files contain the implementation for the application's main entry point and top-level component, respectively.

Overall, this folder structure follows a clear separation of concerns and a **modular approach** to organizing the code. The components are organized by feature or page, making them easier to locate and reuse, and the reusable components are further organized into subfolders for clarity. The custom hooks and Firebase-related files are also organized into their own folders, making them easier to manage.

## Logic diagram

1.General diagram

Diagram

Description automatically generated

2.AuthModal

Diagram

Description automatically generated

3.Home QuestionsDiagram

Description automatically generated

## Routing

Routing is an important aspect of any front-end application that determines how different pages or components of the application are accessed and navigated by the user. In this project, we are using the **next/router** package in combination with **React** to handle routing.

In this code snippet, we are using the push() method of the router object to navigate to the /submit page. This method takes the desired URL as its argument and navigates to that page.

In addition to navigating to a particular page, we can also use routing to handle user authentication and access control. For example, in the following code snippet, we are checking whether the user is authenticated before allowing them to access the /submit page:

const setAuthModalState = useSetRecoilState(authModalState);

  const router = useRouter();

  const createQuestion = () => {

    if (!user) {

      setAuthModalState({ open: true, view: "login" });

      return;

    }

    router.push(`/submit`);

  };

In this code snippet, we are using the **useSetRecoilState** hook to set the state of the **authModalState** when the user is not authenticated. This opens a modal prompting the user to login or register before they can access the /submit page.

# Interface:

## 1.Modals

Graphical user interface, text, application, chat or text message

Description automatically generated

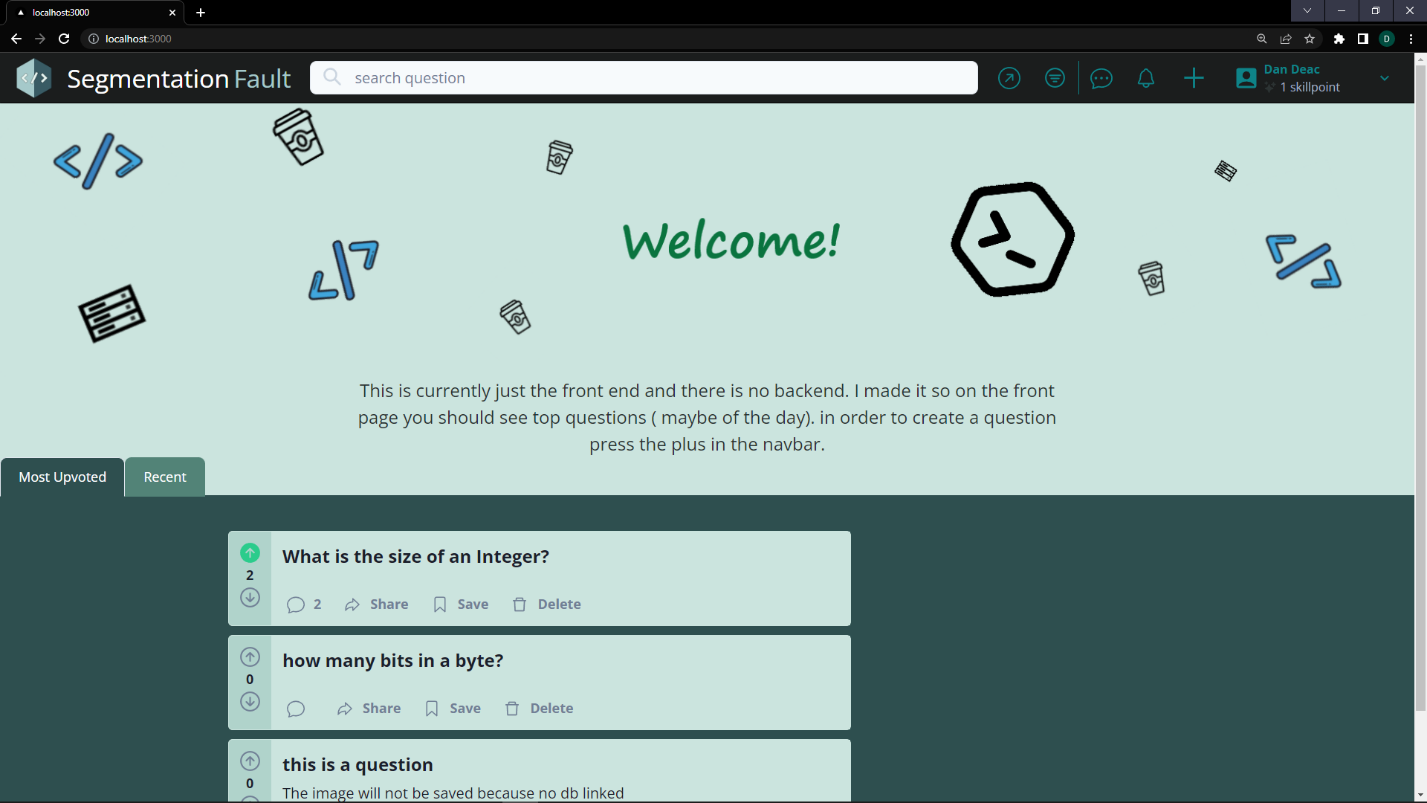
Graphical user interface, text, application, chat or text message

Description automatically generated

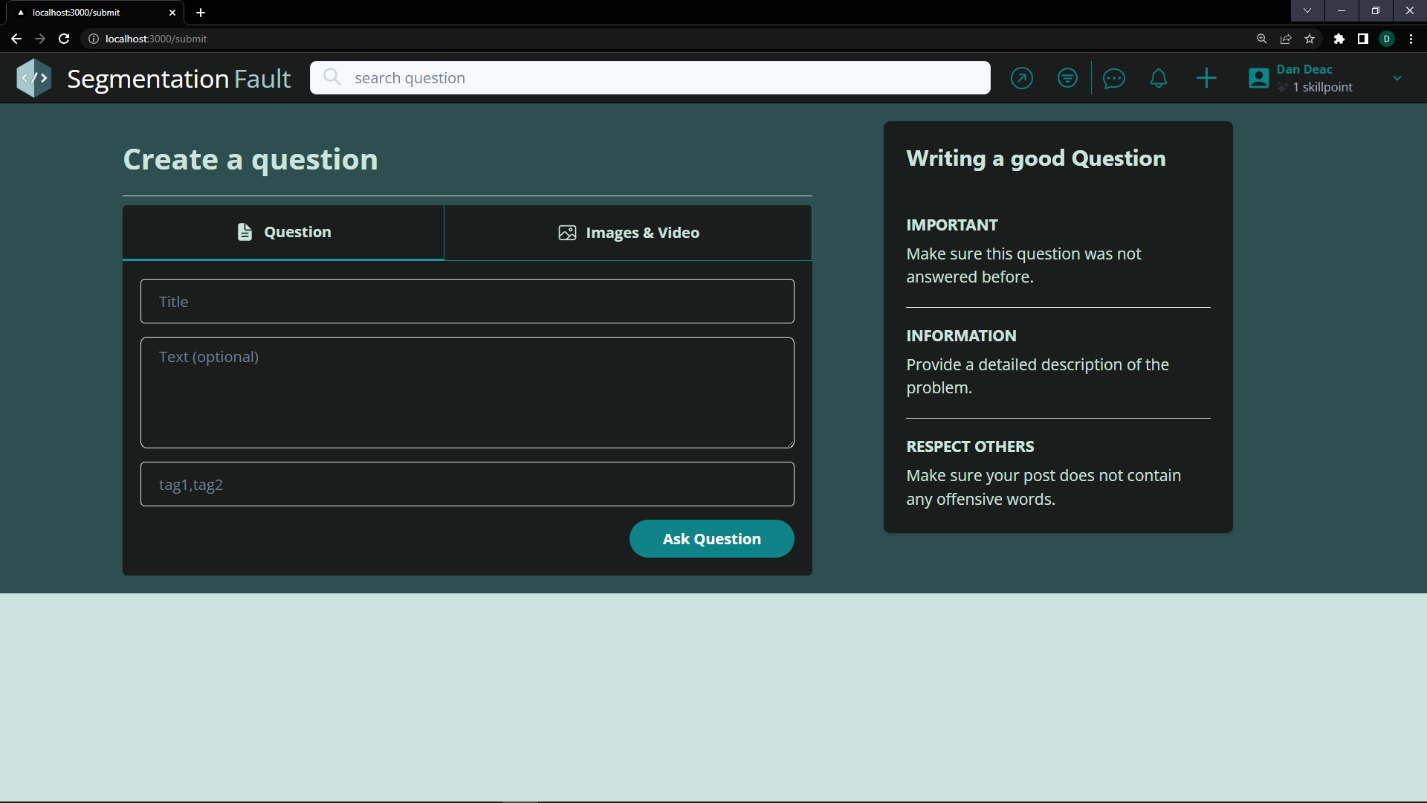
Graphical user interface, text, application, chat or text message

Description automatically generated

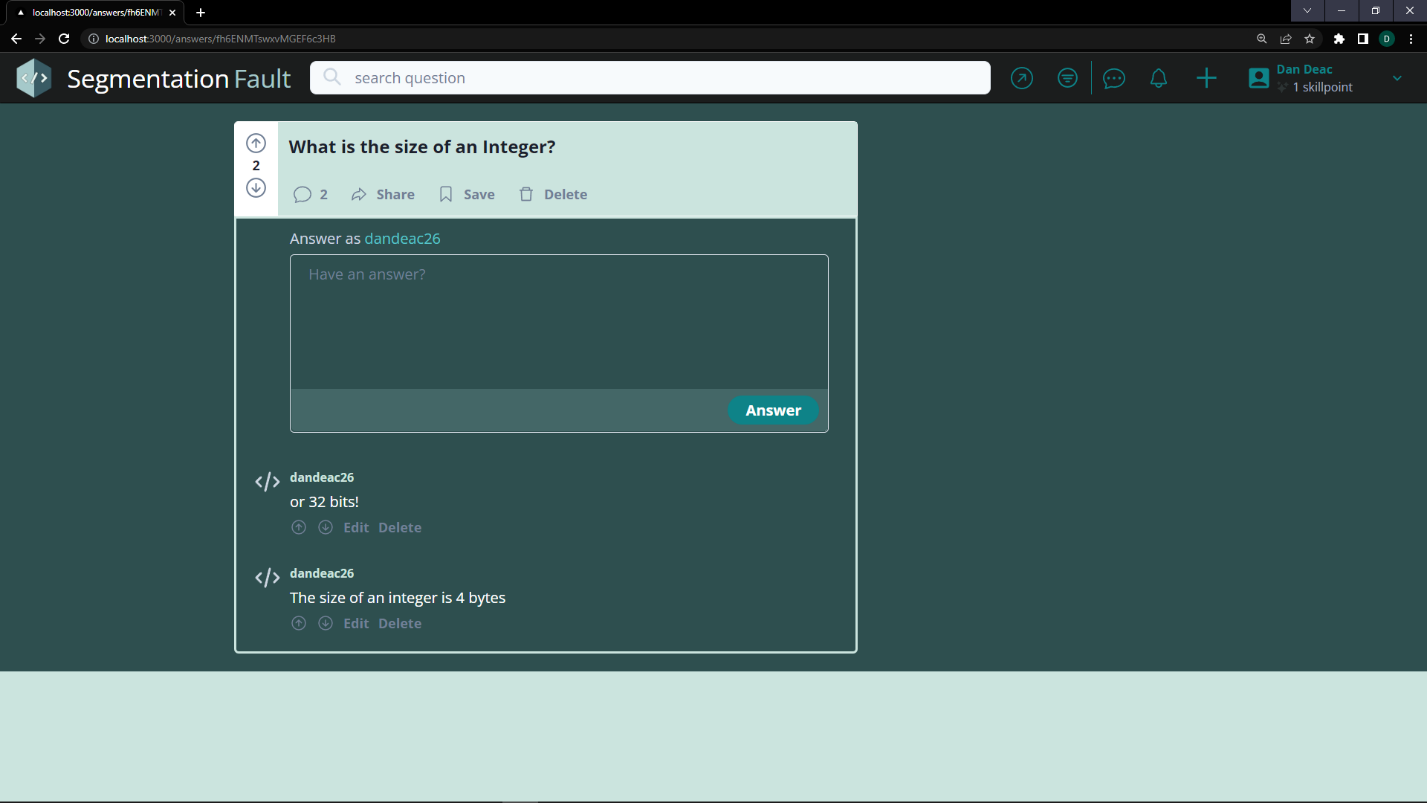
## 2.Home



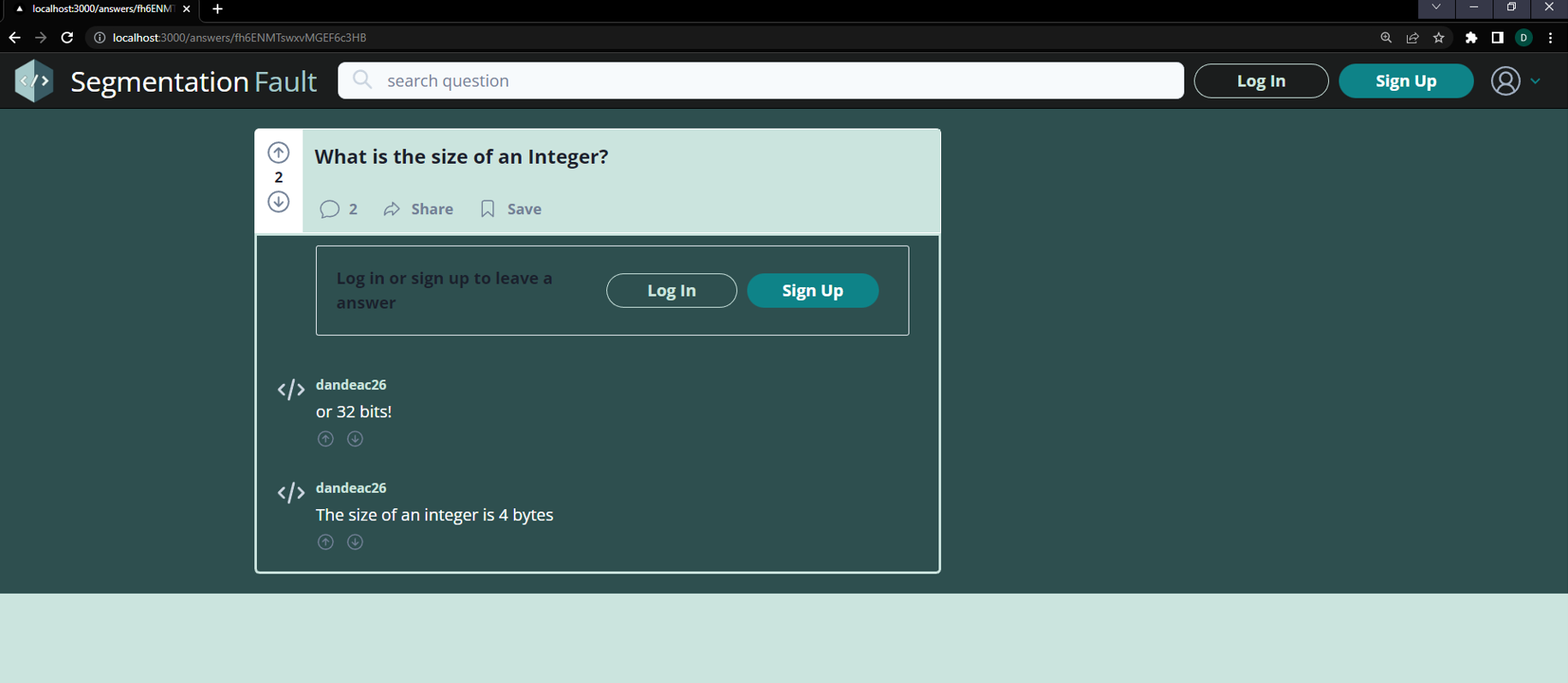
## 3.Ask Question



## 4.Answer



## 5.Before Login



# Conclusion

This is not really a conclusion because this is going to get tied to a database but I had to end on a note. The current state of the interface is quite mediocre. I need to tie things to the database and to the backend in order to perform significant operations and make sure things work fine. I will switch from firebase the mariadb I have on my workstation and hope things work out fine.

## REPO: <https://github.com/dandeac26/SegmentationFaultWeb.git>

!!! Use Branch: **BranchFromWindows**. I have not yet merged / pushed to the main branch for now

# Resources

Here are the resources I used for learning and developing this application:

- StackOverflow.com – for inspiration ofc.

- <https://www.youtube.com/@freecodecamp>

- https://www.youtube.com/watch?v=rCm5RVYKWVg&t=11744s