A close up of a logo

Description automatically generated

**School of science and technology**

**Course code and name:** COSC2638 Cloud Computing

**Assignment:** 2

**Assessment name:** Build a scalable app on Cloud

**Name and ID:** Ho Tuan Huy - s3756868

**Objective**

I would like to create a forum or Facebook like app where people can share their experiences about trips, meal or movies that they watched. Facebook is the most popular social network at the moment but the problems with Facebook is everyone can post what they like to share. There is no limitation about what they can posts on their personal wall. Although there are some groups for the specific purpose, but it will be hard for user to find the best groups or community that will suit their needs. Therefore, I just want to create a social app for a specific purpose when people can only allow to post their review about the related topic of that forum. At the moment, I also create a forum where people can search for movies and write their review about the movies that they watched.

**Motivation and Purpose**

As I mentioned before, I created this web app to support my movies database website. Unlike IMDB or Rotten Tomatoes, my forum will be more like a group on Facebook where user can interact with others. On IMDB, users can only post their reviews, they cannot receive any feedback from others. The interact with users will give a fair and multidimensional point of view about one discussion topic. Some people might like that movie, some might not, however if a movie have a large fan base of a actor who participate in that movie, the positive judgement will over shadow the fair judgements. Therefore, not only movie database website, all the reviews website about tourism or foods and drinks should implement the forum functions.

The forum is very simple, users will be required to create an account to comments and share their idea on the forums. Nevertheless, users are not required to login to view the posts, they are free to read all the posts on the forum. I can use this forum and implement it with all others review website.

**Requirements:**

* Registrations for the forum
* Login and Logout functions
* Create a new Post in the forum
* View all the posts of the forum
* Auto generate some basic information for user when they had successfully created an account
* Allowing users to edit their profile
* Allow user to delete their posts
* Like and unlike the posts
* Allow users to comment on the posts
* Allow user to view all the profile of other users (which included the posts of that user)

**Expected Result**

**Use Case Diagram**

A close up of a map

Description automatically generated

Users:

* Signup: user can sign up for the account. Users will have to provide their email, password and username to create an account.
* Login/Logout: once the account has been created, the email and password will be stored in the Firestore database. Once user login with the correct email and password, the token will be generated in the Authorization Header to allow user to login. For users to logout, I just need to remove the token from the Header.
* View Posts: user does not need to log in to view all the posts. All the posts will be fetched from the Screams Collection in Firestore database.
* Like/Unlike and Comment on Posts: each post will have the different ID and Collections of its. Inside the body of each post will contain some information of like and comments and user details.
* User Profile: when users have successfully created the account, the system will automatically generate some basic information for them. The will all have the same default image and the joined date. Users can change their profile image and some bio information.

**Activity Diagram**

A close up of a map

Description automatically generated

As mentioned before, users are not required to login to view all the posts, all the posts are opened for everybody to view. However, users are required to login to comments and like the posts. First of all, user have to provide to email, password and username to create the forum account. Then, will use that email and password to login. The email and username when register for the forum have to be unique, especially the username have to be unique so we can distinguish between users. After successfully logged in the system, user can create a new post, comment and like on other users’ posts. Finally, user can log out of the system. However, if the users forget to log out the system, we will automatically log out user from the system for security reason.

**Architecture and Implement**

**Backend**

**Why using Firebase?**

For the backend API, I prefer to use Google Firebase for my database and backend. Firebase is a platform for developing mobile applications and websites, including simple and powerful APIs without a backend or server. Firebase also helps programmers shorten the time to deploy and scale the applications they are developing.

Firebase is a database service that operates on cloud - cloud platforms. Attached to that is Google's extremely powerful server system. The main function is to help users program the application by simplifying operations with the database. In particular, simple API application programming interfaces. The goal is to increase the number of users and get more profits.

In particular, it is a versatile service and extremely good security. Firebase supports both Android and IOS platforms. It is not difficult to understand that many developers choose Firebase as the first platform to build applications for millions of users around the world.

Firebase provides NoSQL realtime cloud database. Stores data in JSON format and allows synchronization with the client When connecting Realtime Database to Android, iOS, Javascript SDK, a realtime database will be created and shared for all users. All clients will receive updates when data changes

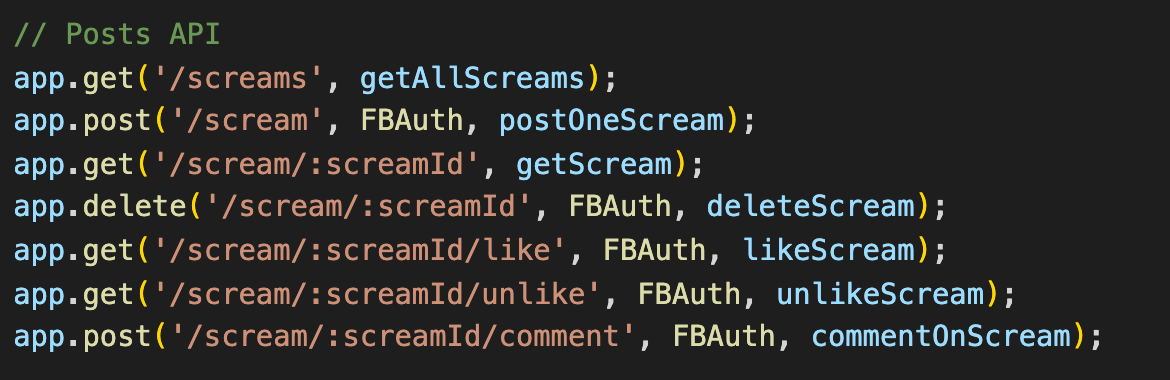
Using Firebase Authentication user will authenticate your account in many ways, email, Facebook, Twitter, Google or Github. Firebase Authentication allows creating a new user to save the user's authentication to the Firebase Database, no longer having a hard time configuring on the server side. We can even confirm emails after registration and forget the password.

A screenshot of a cell phone

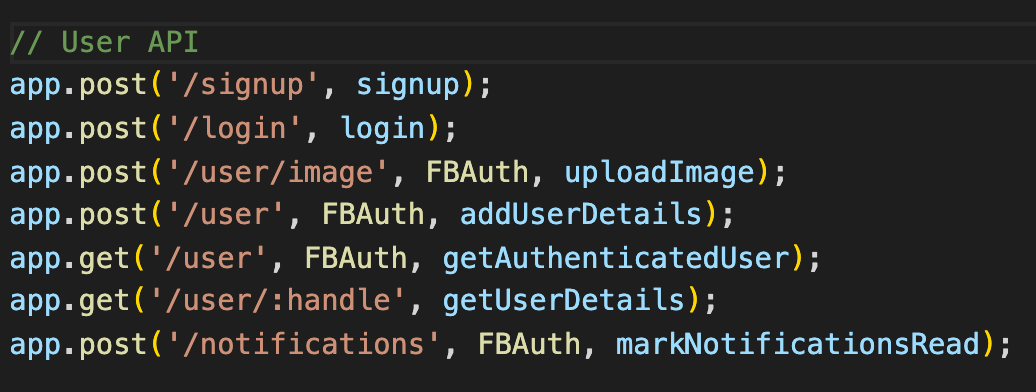
Description automatically generated

First of all, we have to create a project on Firebase. Then, we have to open Terminal and cd to the backend folder and use “npm install -g firebase-tools” to install Firebase Tools on our PC. After that, we have to run “firebase login” to login our Google Firebase account. Then we have to run “firebase init” and we also have to select the third option which is “Functions: Configure and deploy Cloud Functions” to create a functions folder which will be our backend folder. We will also be asked to select the project on Firebase that we want to work with. All of these steps are very simple and straight forward with guide in the Terminal.

For the backend design, I created screams.js to handle the functions that related to posts and the users.js to handle the functions that related to users. Post and status are the reserve words, so I replaced them with Screams. The admin.js is for getting the firebase database, the FBAuth.js is to authorization and the validators is for validate the login, logout and authorization to edit the profile. The Endpoint URL will be coded in the index.js file. The index.js is also handle some functions such as: create notification on comments and like, change user Image and delete posts.



Posts API endpoint URL



User API endpoint URL

We also have to create the Indexes for some collections

A screenshot of a cell phone

Description automatically generated

To test the Firebase URL on localhost, we just simply run “firebase serve” and the default port will be port 80. If we would like to deploy the API to Firebase, just use “firebase deploy”.

Function API: https://asia-east2-socialapp-32fb9.cloudfunctions.net/api

**Front End**

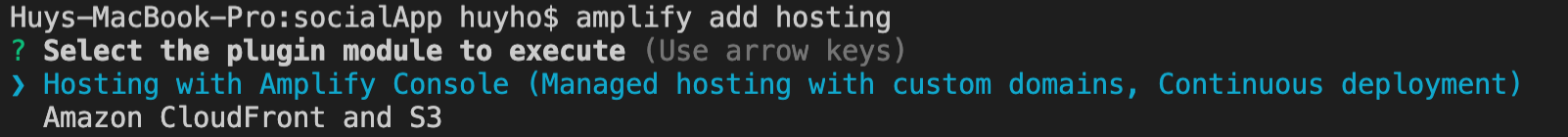
For the front end, I prefer to use React-Redux. Creating a react app is very simple. We just need to use “npx create-react-app app-name” to create a React App. Moreover, I also use some library such as react-router-dom to link the pages without reloading the whole website. I also use “axios” to fetch the API and “@material-ui-core” framework for CSS design. I also use “dayjs” to calculate the joined date and the time that the posts have been created. Developer can run project on localhost:3000 by using “npm start”.

Each components of the project such as like buttons, signup form, login form will be a separate JavaScript file and will be called out in its page file. There are three main pages: Home, Login and Signup. Each page will be navigated by using “react-router-dom”

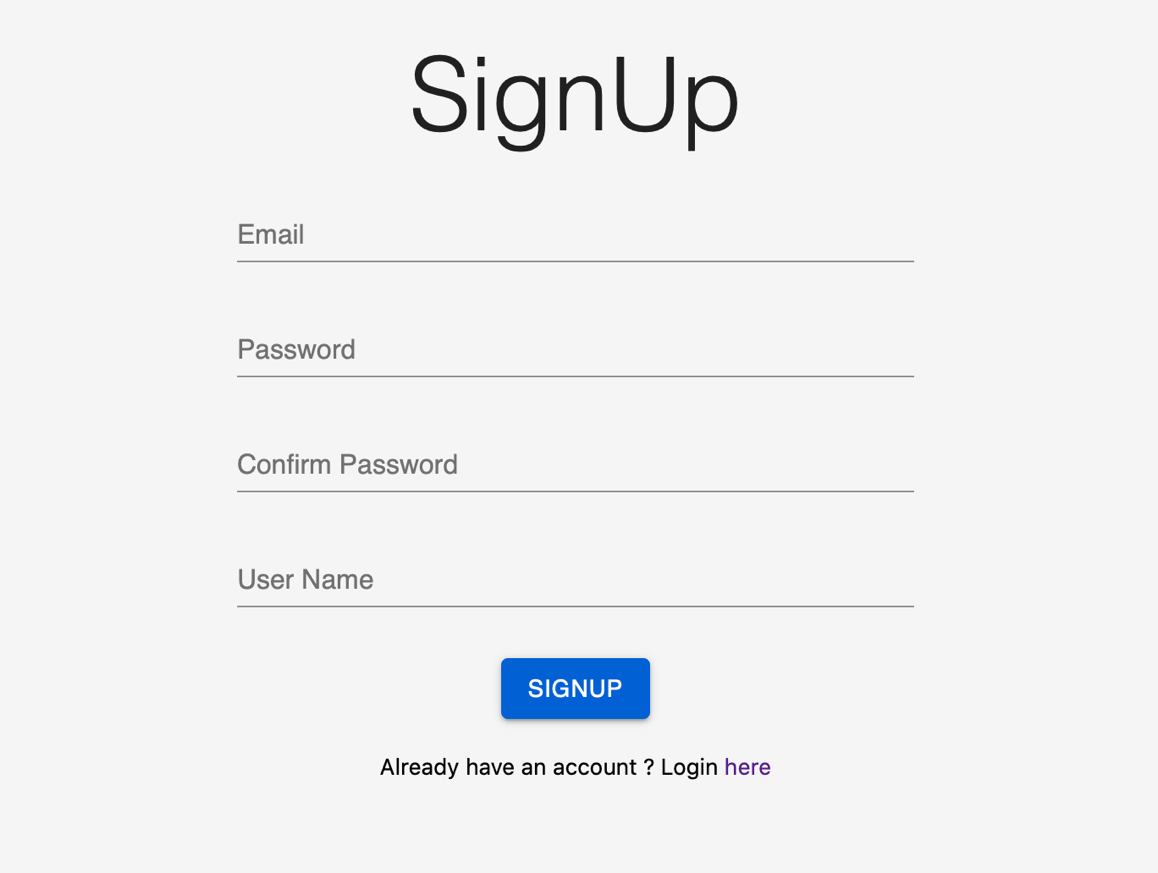
For the front end, I prefer to use Amazon Amplify to publish my static website to the S3 Bucket and CloudFront.

First of all, we have to install Amplify by using “npm install -g @aws-amplify/cli” to have Amplify work on our PC. After that, we will cd our Terminal to the front-end folder and run “amplify init” to install all the necessary files and configurations for our Amplify. This step is very simple, we just need to read carefully the instruction in the Terminal to get all the name correctly. We need to be more careful on the scripts to run and start the project in React. The scripts might be different for different people and frameworks. We need to look at the “package.json” file to check our start and build scripts.

After that, we have to add hosting by using “amplify add hosting”. Select “Amazon CloudFront and S3” option. Then, I use DEV options to deploy my front end to S3 Bucket. The DEV option only provide the HTTP connection. After that, we just need to run “amplify publish” to deploy our front end to S3 Bucket.



**User Manual**



Users have to provide their Email, Password and Username to create an account. After successful create an account, user will be automatically login our website. The email address has to be unique and have not been used before. Likely, the Username also have to be unique. The password and confirm password have to be the same and have more than 6 characters. In addition, all the fields are required to be filled.

A screenshot of a cell phone

Description automatically generated

User have to provide the matched Email and Password that they signed up to log in.

A screenshot of a cell phone

Description automatically generated

User can click on the @Name to go to their Personal Wall to view their profile and Posts. They can click on the pencil next to the profile image to change the image. They can also change their bio by clicking on the second pencil at the right corner. Finally, users can log out by clicking on the logout icon on the left corner.

A picture containing drawing

Description automatically generated

On the Navbar, user can click on the Add Icon to create a new post. The house Icon is to go back to the homepage and the bell Icon is for viewing the Notifications. When there are the new coming notifications, the number of the unread notifications will be showed.

A screenshot of a cell phone

Description automatically generated

The blue notifications are stand for “Read Notifications”. The unread notifications will be displayed in read.

A screenshot of a cell phone

Description automatically generated

For the posts in the homepage, if there is a read trash appear in the top right corner, that means these are your posts and you can delete them. Users cannot delete others’ posts. Users can click on the expand Icon in the bottom right corner to expand the post to view all the comments of the posts or to write the comments to the posts. The number of likes and comments are also be displayed on the post cards. Users can also click on the name of the person who write the post to view their profile and posts.

Website URL: <http://socialapp-demoapp.s3-website-ap-southeast-1.amazonaws.com>

Functions API: <https://asia-east2-socialapp-32fb9.cloudfunctions.net/api>

GitHub: <https://github.com/huynino96/socialApp>

GitHub project that use the app: <https://github.com/huynino96/SEPM-moveeHub-Group4>

**References:**

Redux Dev Tools:

<https://github.com/zalmoxisus/redux-devtools-extension>

Firebase Firestore:

<https://firebase.google.com/docs/firestore?hl=vi>

Firebase Authentication:

<https://firebase.google.com/docs/auth?hl=vi>

Firebase Storage for Image:

<https://firebase.google.com/docs/storage/web/start?hl=vi>

Firebase Cloud Function (for JavaScript):

<https://firebase.google.com/docs/functions?hl=vi>

AWS Amplify:

<https://docs.amplify.aws/cli>