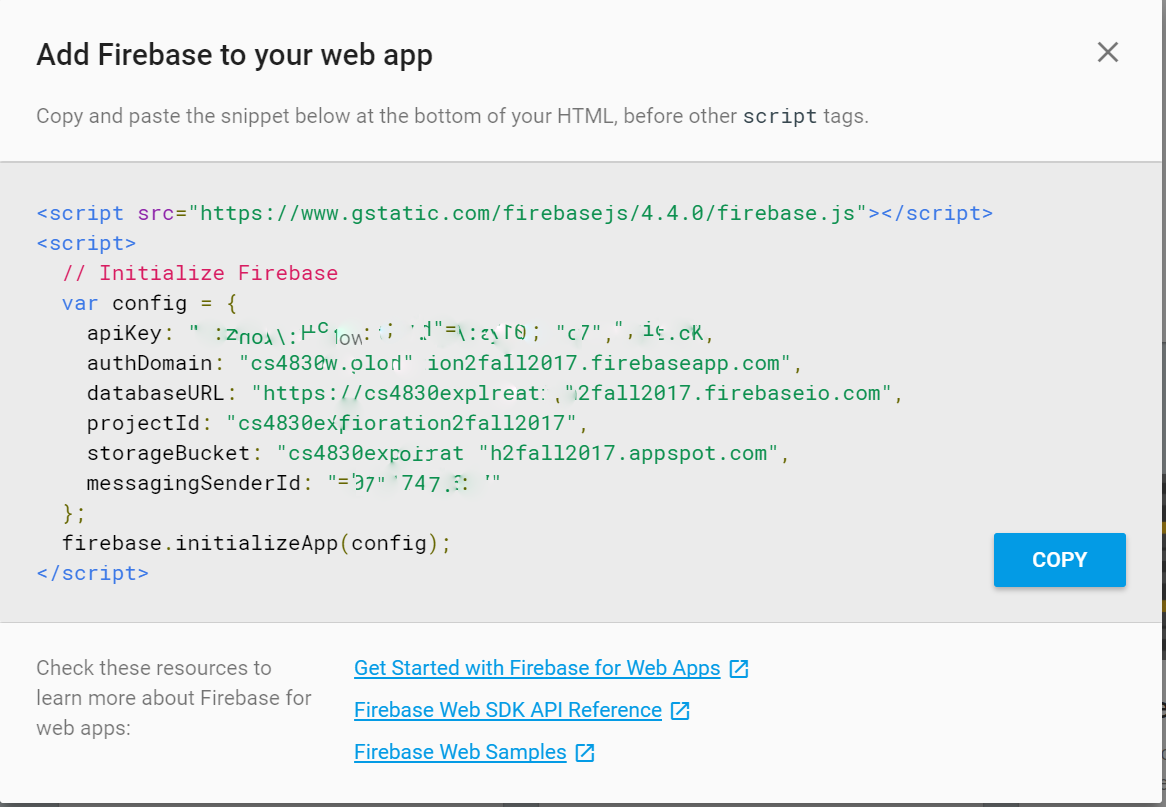
Journal:

For this exploration, I have dived into Google’s Firebase ecosystem, an all-in-one ecosystem that provides a NoSQL database with storage, authentication framework, and simple hosting system. Google’s Firebase is useful for developers to use since it supports web apps, both Android and iOS, and a lot more language for back-end scripting.

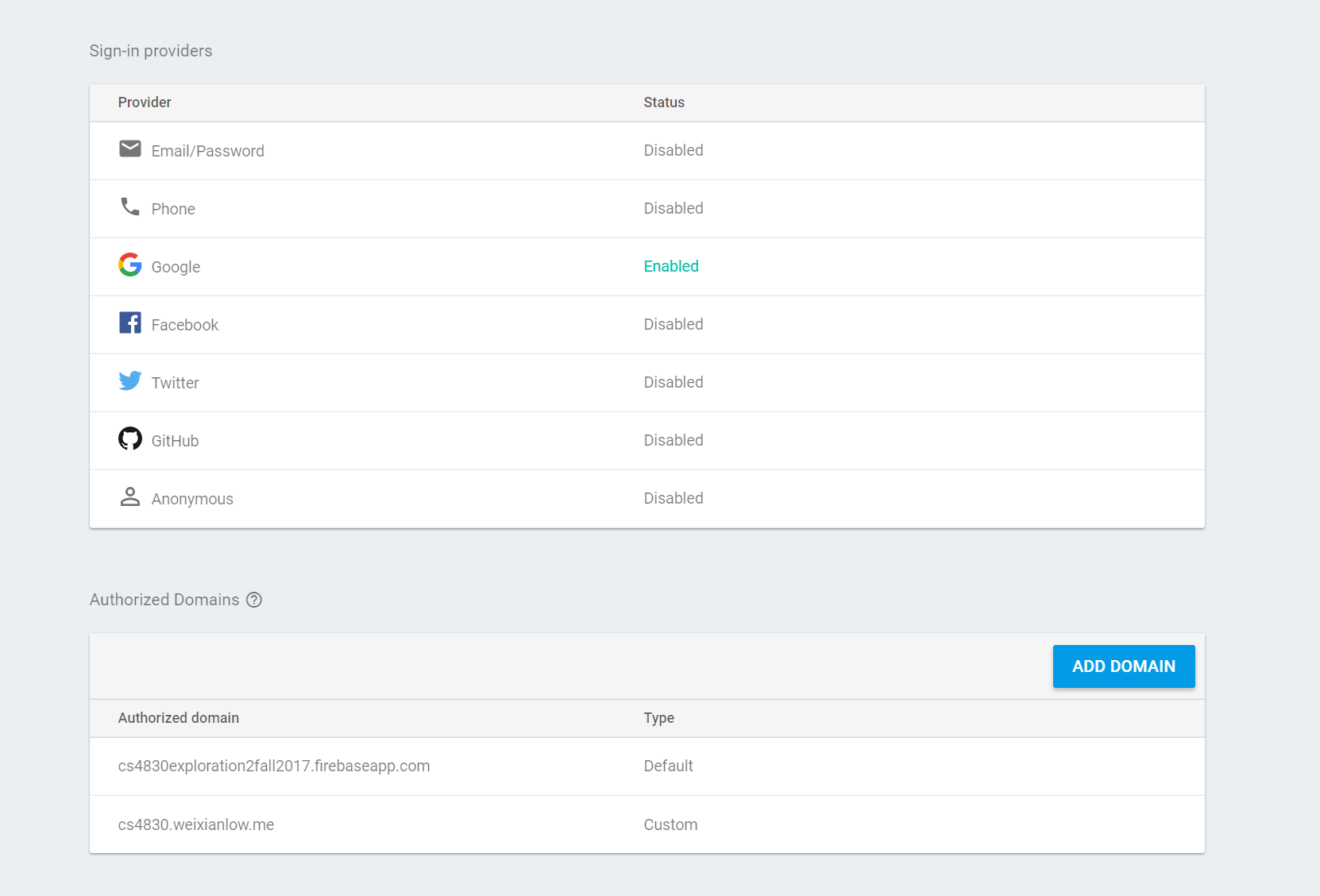
In this exploration, I’m focusing on the authentication framework and Realtime Database provided by Google. Firebase Authentication allows developers to have the ease to authenticate user into their application by providing not only Google login, but third-party services such as Facebook, Twitter, and GitHub. Firebase Authentication also provides an open-sourced UI that allows developers to create a standardize login UI in a Public Repository at GitHub (<https://github.com/firebase/firebaseui-web>).

The Firebase Realtime Database allows developers to use the same database cross-platform on different ecosystem, while also creating a “live” database, where data is provided in Realtime to the user based on how the developer has developed the app.

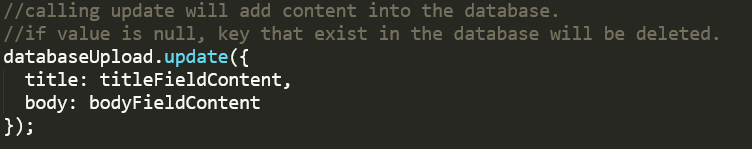
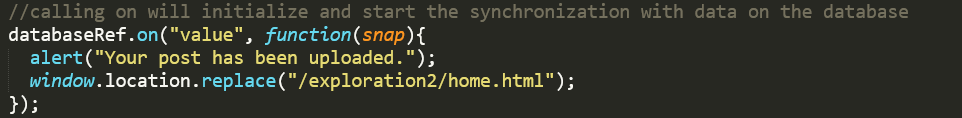
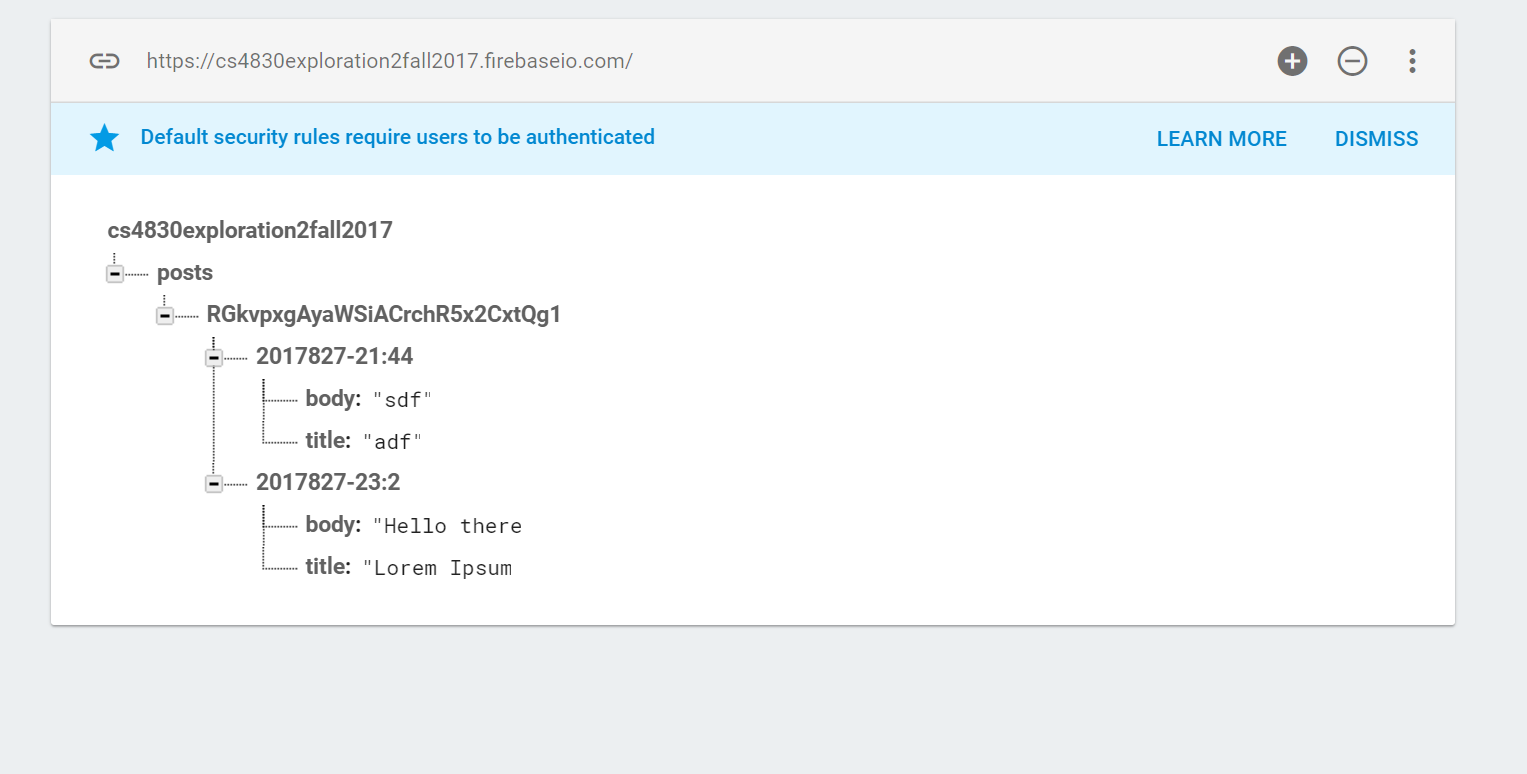
To get started on how to start using Firebase for web, you would need to create a JavaScript file to contain all your configuration and tokens. By heading to the Firebase Console, you could obtain all your configuration and save it into a file, where it could be referenced by all the webpages that uses Firebase. An example configuration file would look like this:



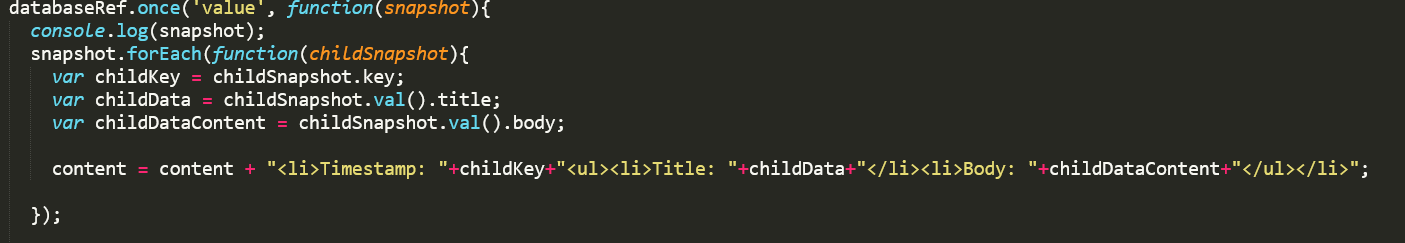
To use Firebase Authentication:

1. Head over to Firebase Console, and navigate to the Authentication section
2. Turn on which sign-in method you would prefer, for my case, it would be Google Sign-in,
3. Make sure that you have added your custom URL in the allowed URL to be called from an authorized domain. 
4. Head to your webpage source code and input this segment of code: 
5. Be sure to change the signInSuccessUrl and add additional signInOptions if you have more than one sign-in method.
6. Add a div with the id = firebaseui-auth-container to display the login UI on wherever you want on the webpage.

To write to Firebase Realtime Database:

1. Ensured that you have properly configured your firebase to include the necessary config information.
2. To write data to the server, the best way to do it is to form it into a json object.
3. First thing to write a data is to properly structure your data into an easy form, avoid nested data, and make it as flat as possible.
4. Once figuring out how to structure your data, you would then create a reference to navigate to the structure you want to store:C:\Users\NICKL\AppData\Local\Microsoft\Windows\INetCache\Content.Word\databasereference.png
5. Once you have created a reference on where you wanted your data to be stored, you can then call on update to store data like so:
6. Once you’re done with the update, the last step is to push the data to the database by calling the on function in firebase:
7. If successful, your database will be as so:

To read from Firebase Realtime Database:

1. Reading from Firebase is easy, you would need to first create a database reference to pull the needed data as shown as how we did on writing a data.
2. Next you’ll have this segment of code to pull a snapshot from the database:
3. Be sure to call .val() to obtain the value of the key/value pair.

Sources:

<https://firebase.google.com/docs/>

* General Configuration:
  + <https://firebase.google.com/docs/web/setup>
* Firebase Authentication:
  + <https://firebase.google.com/docs/auth/web/google-signin>
  + <https://github.com/firebase/firebaseui-web>
* Firebase Realtime Database:
  + <https://firebase.google.com/docs/database/web/read-and-write>
  + <https://firebase.google.com/docs/database/web/lists-of-data>
  + <https://firebase.google.com/docs/database/web/structure-data>

Links:

Github Repository: <https://github.com/weixianlow/CS4830_Exploration2_Fall2017>

Instances: <http://cs4830.weixianlow.me/exploration2>