Clouds Assignment 5

Objective: In this assignment, you will create and deploy a web application on the Cloud using Google Firebase.

1. Configure Firebase Project

Firebase is a Backend-as-a-service platform developed by Google for creating mobile and web applications on the Cloud.

Firebase provides many services such as Authentication, Firestore database, Real-time database, storage, hosting, functions and machine learning. In this assignment, you will use part of these services.

Your application will use Firebase authentication to allow users to sign in, Firebase real-time database and Firestore to store data, and Firebase hosting to deploy it on Cloud.

Firebase Authentication allows you to use different sign in methods such as Sign in with Google, phone number, email and password, Facebook, Github and many others. In this assignment, you will use Sign in with Google to allow users to sign in to the application using their google account. You can read more about Firebase Authentication by clicking here.

Firebase Real-time database allows you to store noSQL data on the cloud in JSON format. Querying the database will return subtree of data in JSON format.

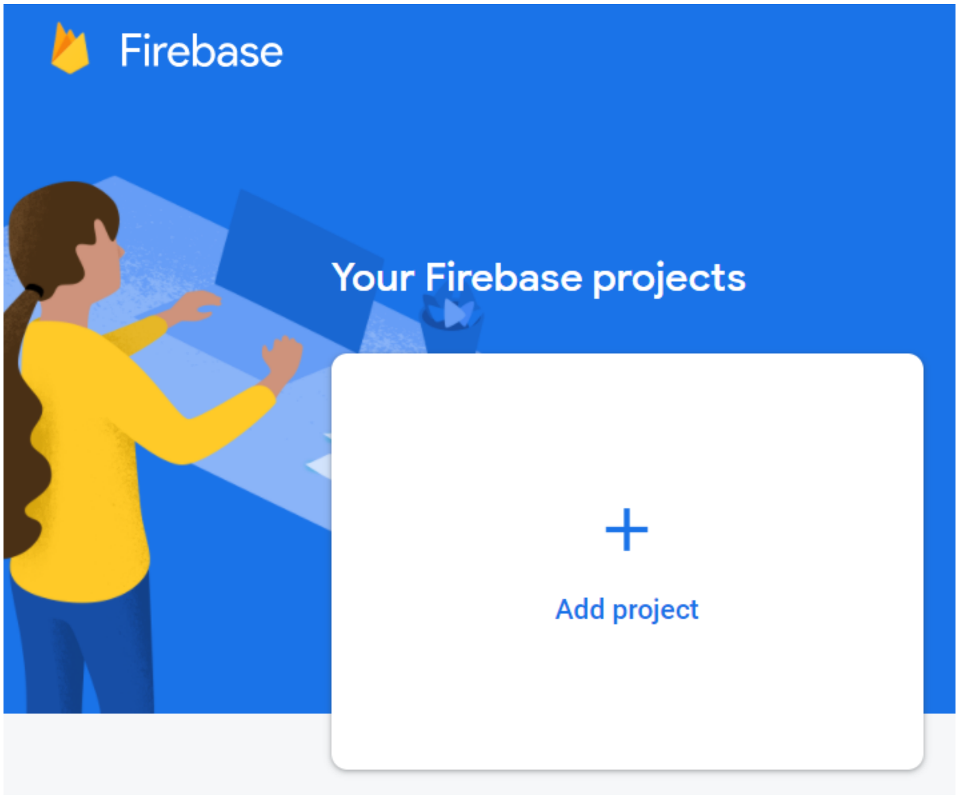
Firestore database allows you to store noSQL data on the cloud in a Collection of documents format. Querying the database will return subset of a collection of documents.

Click here to read more about Real-time database and firestore and to understand the difference between them.

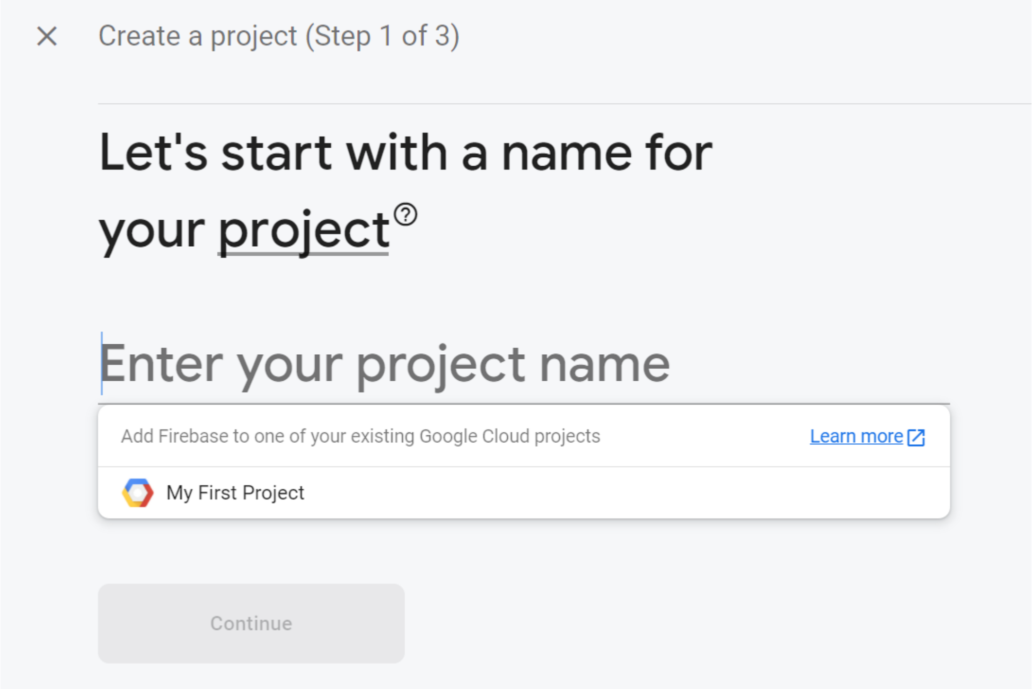
Firebase hosting allows you to deploy your web application on the Cloud. Click here to read more about it.

To start your assignment, you have first to use a Google account to access firebase console: https://console.firebase.google.com/

From there, click on “Add project”.



Next, assign a name to your project.



Create a project (Step 1 of 3)

Let's start with a name for your project

Enter your project name

Add Firebase to one of your existing Google Cloud projects

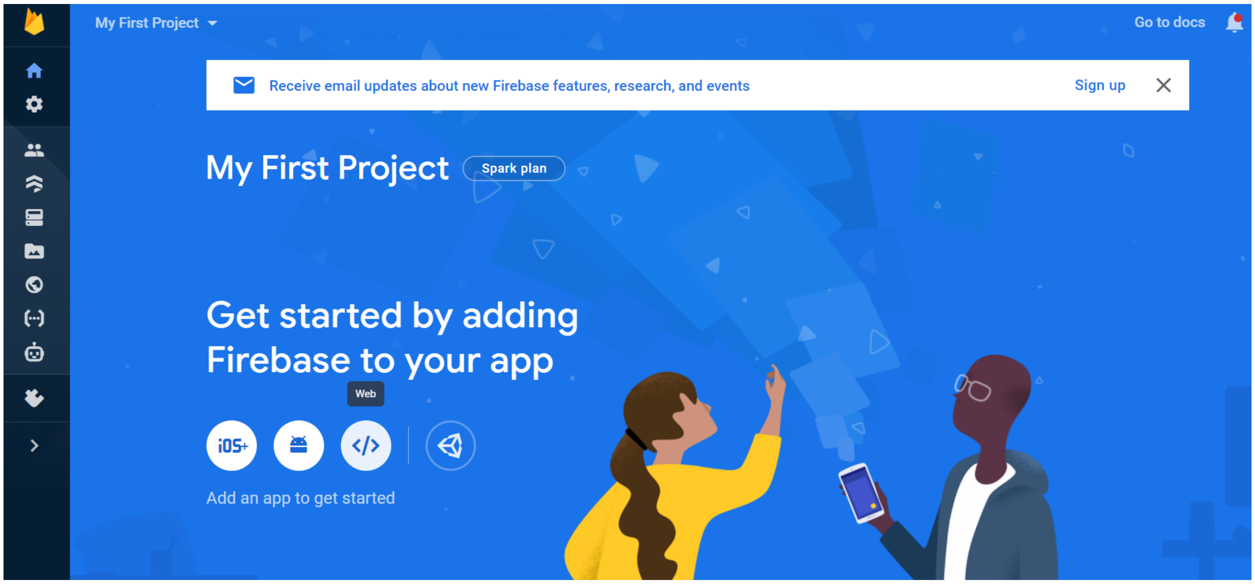
Learn more

My First Project

Continue

Then continue with the wizard to create your project.

Once you create your project, you will be redirected to the dashboard of your project.



My First Project

Go to docs

Receive email updates about new Firebase features, research, and events

Sign up

My First Project

Spark plan

Get started by adding Firebase to your app

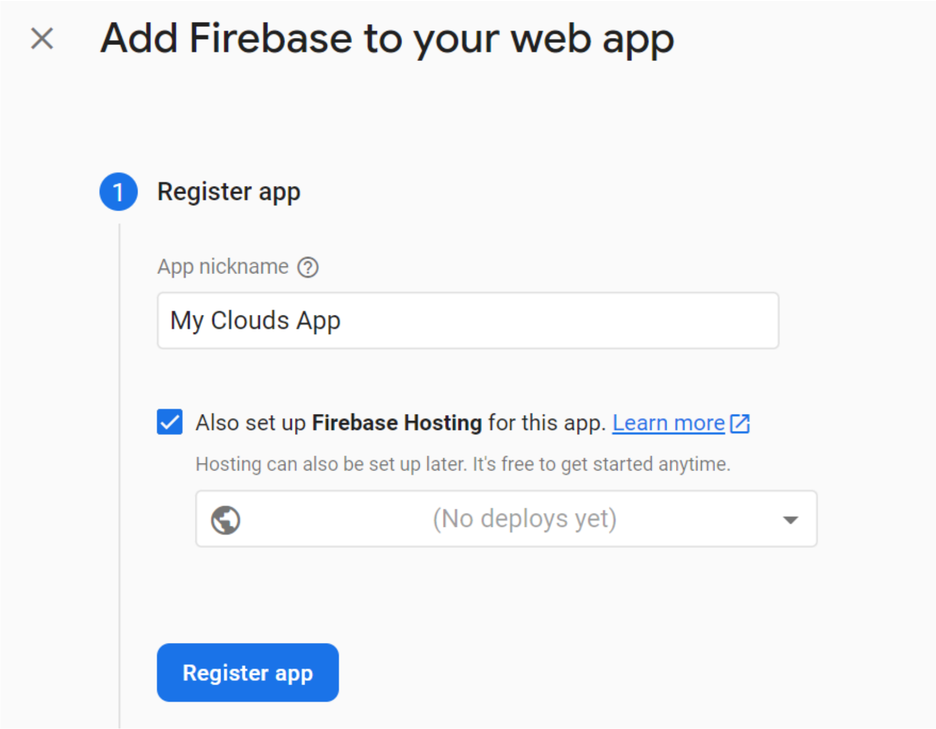
Web

iOS+

Add an app to get started

On the left side menu, you can access the settings of your project, in addition to the different services provided by firebase. You will use many of these services in your assignment.

But first, you need to add a web app to your project. To do this, click on the “</>” icon, under the “Get started by adding Firebase to your app”.



Add Firebase to your web app

1 Register app

App nickname

My Clouds App

☑ Also set up **Firebase Hosting** for this app. Learn more

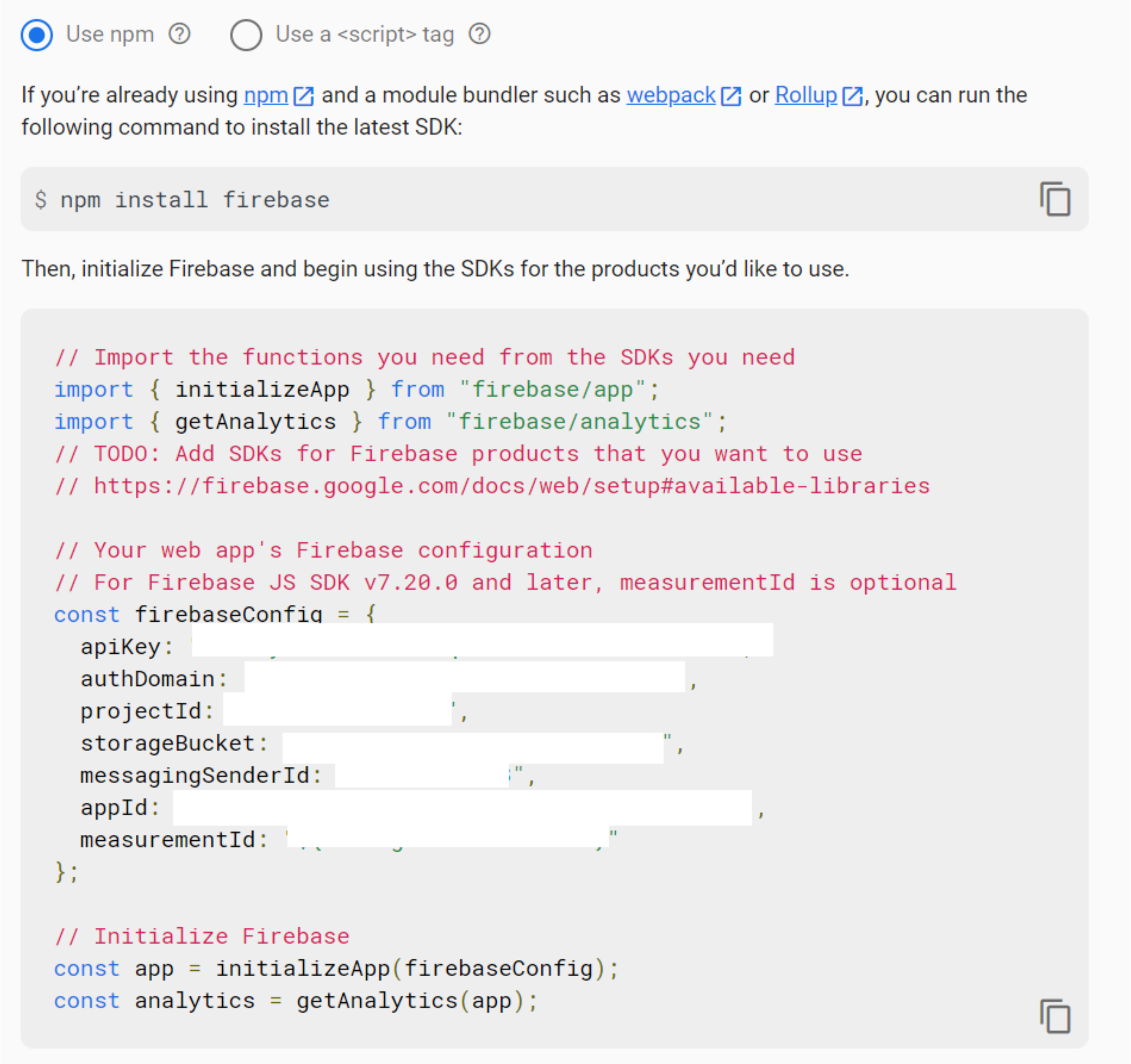
Hosting can also be set up later. It's free to get started anytime.

(No deploys yet)

Assign a name to your app, and check the firebase hosting option, the click “Register app”.

Next, you will have to install firebase. You can do this by running the npm command on terminal or cmd.

Then you will need to copy the firebase config to a web project that you created using a front-end JavaScript based project or a framework such Angular, React, Vue.



● Use npm ○ Use a <script> tag

If you're already using npm and a module bundler such as webpack or Rollup, you can run the following command to install the latest SDK:

```
$ npm install firebase
```

Then, initialize Firebase and begin using the SDKs for the products you'd like to use.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { getAnalytics } from "firebase/analytics";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: [redacted]
  authDomain: [redacted],
  projectId: [redacted],
  storageBucket: [redacted],
  messagingSenderId: [redacted],
  appId: [redacted],
  measurementId: [redacted]
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const analytics = getAnalytics(app);
```

After adding firebase to your web app, from the left side menu create firebase real-time database and a firestore database. Make sure to allow read and write rules. I recommend that you set the rules to test mode during development.

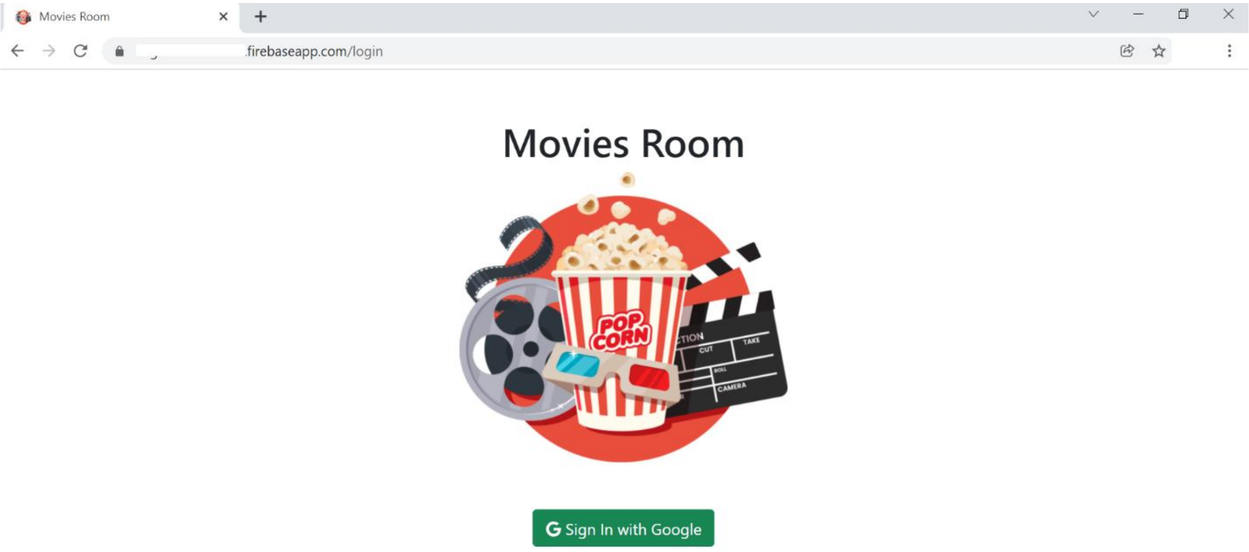
2. Create Web Application using Firebase

The application that you need to develop will allow a user to browser a movies database and pick his wishlist from it.

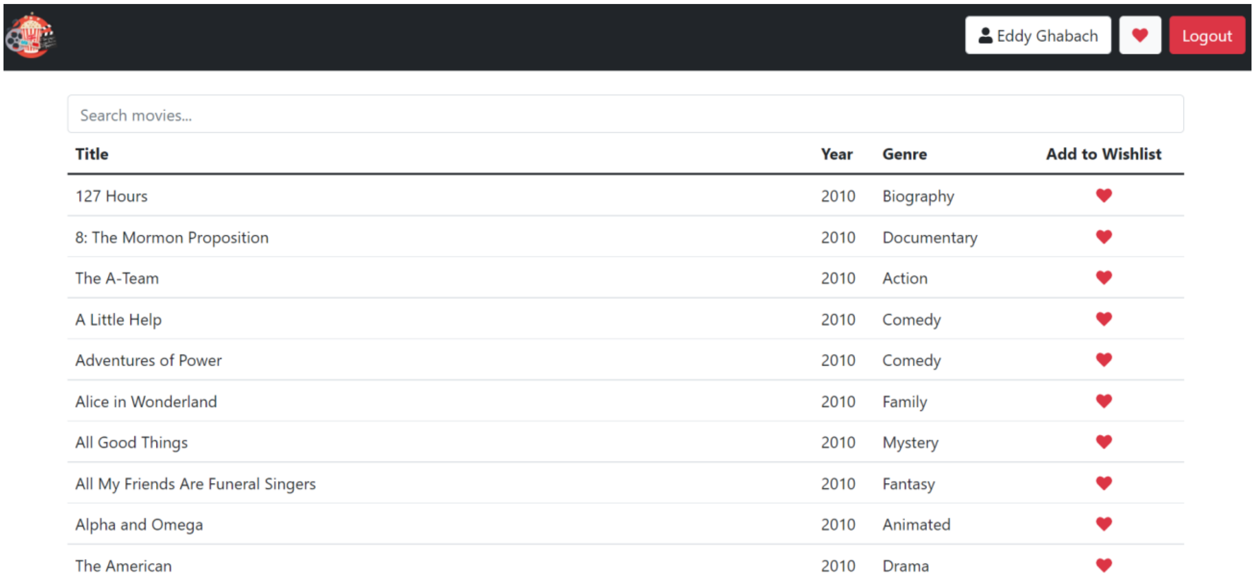
First, when a user accesses the application, he will need to Sign in with his Google account.

You are requested to use a “favicon” (the icon that appears on the tab) and use “Movies Room” as title for your website (also displayed on the tab). You can pick any image and use it on the login page.

When a user clicks on “Sign in with Google” button, the Google sign in pop up screen will open so he can sign in with his google email. To implement this functionality, you have to enable the “Sign in with Google” method from the Firebase menu. In addition, a code corresponding to this functionality has to be added to the front-end application.



Once the user is signed in, the dashboard appears.

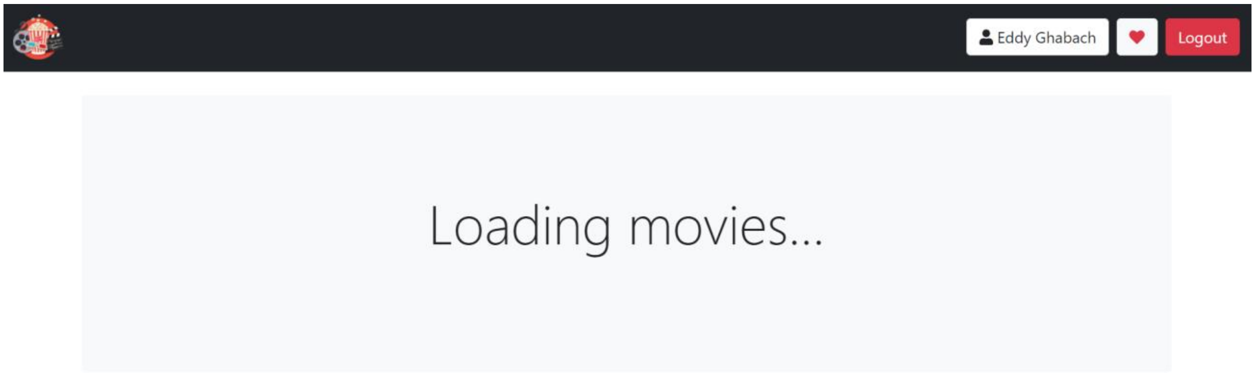


Title	Year	Genre	Add to Wishlist
127 Hours	2010	Biography	♥
8: The Mormon Proposition	2010	Documentary	♥
The A-Team	2010	Action	♥
A Little Help	2010	Comedy	♥
Adventures of Power	2010	Comedy	♥
Alice in Wonderland	2010	Family	♥
All Good Things	2010	Mystery	♥
All My Friends Are Funeral Singers	2010	Fantasy	♥
Alpha and Omega	2010	Animated	♥
The American	2010	Drama	♥

First, we have the navbar on top. On the left side you will need to display the logo of your application (you can use any logo). On the right side, You will display the name of the user (this information you get it from the Google Sign in response), a button to access the wishlist (the heart icon) and a button to Logout. The left-side logo, has to return the user to this dashboard whenever clicked (i.e. when the user is inside wishlist and clicks on logo). The wishlist button when clicked will take the user to the wishlist page (to be introduced later). The logout button, once clicked must take the user back to the login page.

Below the navbar, the list of movies has to be displayed. You will be provided with a JSON file containing the data (movies.json). You will need to import this data into the firebase real-time database that you created. Your application has to collect this data from the firebase real-time database and display it.

The short delay after signing in, before the movies are loaded and displayed, the interface has to show that movies are loading. An interface similar to the one below.



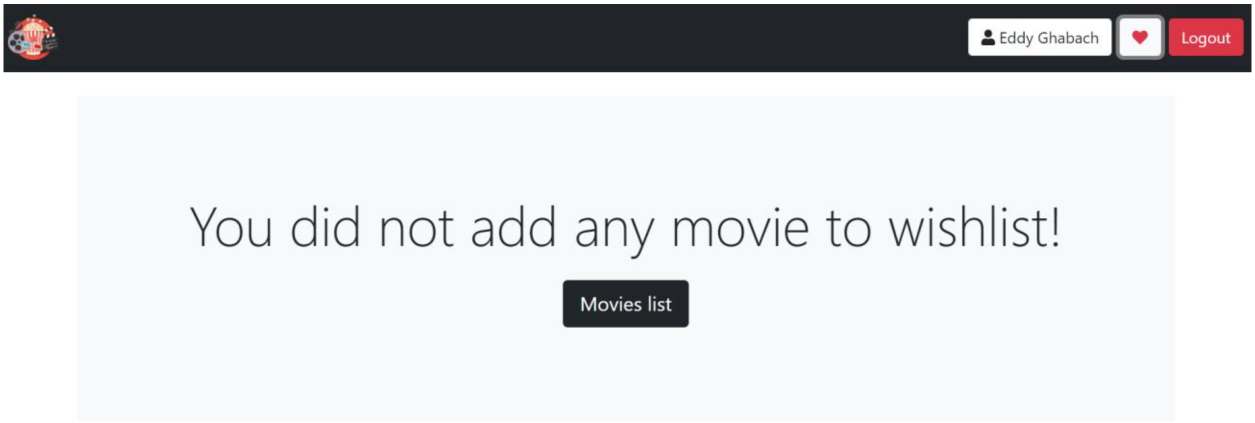
Loading movies...

When the user clicks on the heart beside a specific movie, the movie has to be added to the wishlist of this user. Respectively, the movie has to disappear from the movies list (you don’t have to delete it from database, but hide it on the interface). To add a movie to the wishlist of a user, you will need to save this information in firestore. Hence, for each user you will save his own wishlist of movies in firestore. Note that, when a user signs in, he must see on dashboard only the movies that are not on his wishlist.

When the user clicks on the wishlist button (heart icon) on the navbar, we will enter his wishlist page.

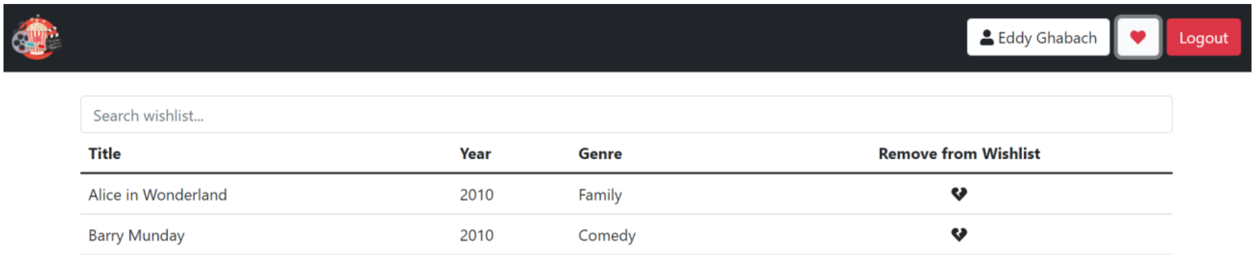
If the user has no movies on his wishlist, the page must appear as shown below.

The user can click on the “Movies List” button to get back to the dashboard.



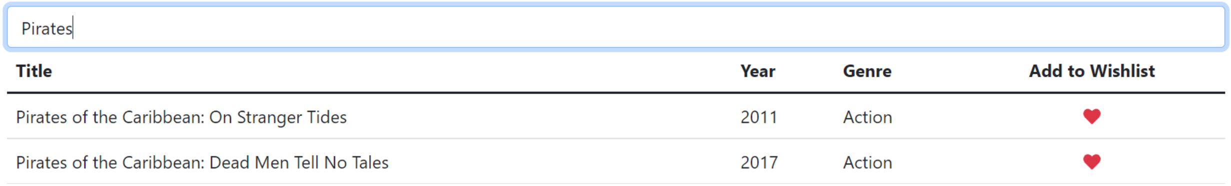
You did not add any movie to wishlist!

If the user has movies on his wishlist, the movies appear as shown below. The user can click on the remove from Wishlist icon to remove a movie from his wishlist. Note that anyone can sign in to the app, and each user has his own wishlist. When a movie is removed from the wishlist, it appears again on the dashboard.



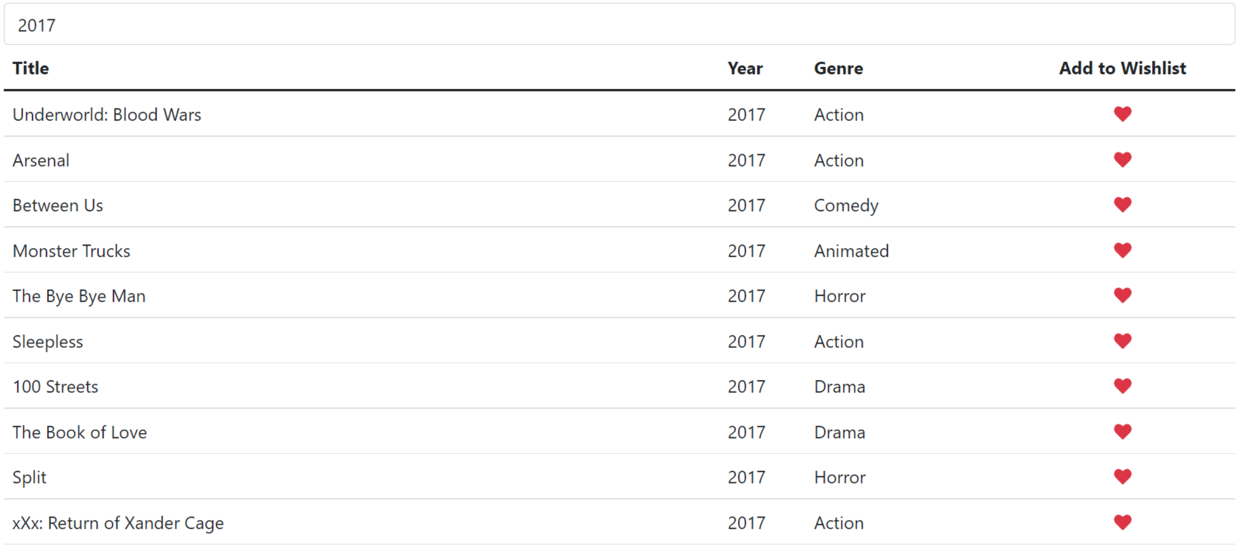
Title	Year	Genre	Remove from Wishlist
Alice in Wonderland	2010	Family	💔
Barry Munday	2010	Comedy	💔

The search field, in dashboard or wishlist, allows the user to search movies by title, year and genre.



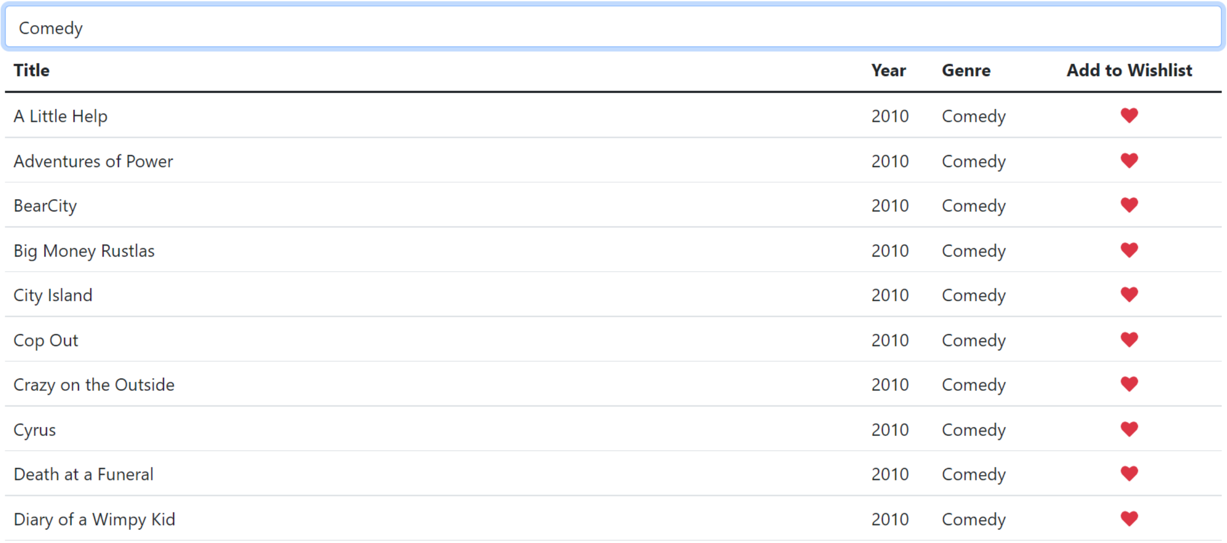
Pirates

Title	Year	Genre	Add to Wishlist
Pirates of the Caribbean: On Stranger Tides	2011	Action	♥
Pirates of the Caribbean: Dead Men Tell No Tales	2017	Action	♥



2017

Title	Year	Genre	Add to Wishlist
Underworld: Blood Wars	2017	Action	♥
Arsenal	2017	Action	♥
Between Us	2017	Comedy	♥
Monster Trucks	2017	Animated	♥
The Bye Bye Man	2017	Horror	♥
Sleepless	2017	Action	♥
100 Streets	2017	Drama	♥
The Book of Love	2017	Drama	♥
Split	2017	Horror	♥
xXx: Return of Xander Cage	2017	Action	♥



Comedy

Title	Year	Genre	Add to Wishlist
A Little Help	2010	Comedy	♥
Adventures of Power	2010	Comedy	♥
BearCity	2010	Comedy	♥
Big Money Rustlas	2010	Comedy	♥
City Island	2010	Comedy	♥
Cop Out	2010	Comedy	♥
Crazy on the Outside	2010	Comedy	♥
Cyrus	2010	Comedy	♥
Death at a Funeral	2010	Comedy	♥
Diary of a Wimpy Kid	2010	Comedy	♥

Finally, you have to disallow a user to access the dashboard and wishlist pages if he is not signed in. Respectively, you have to disallow a signed in user to access login page.

When done, build your project, and deploy it on Firebase hosting.

Push your website to a repository that you create on gitlab.eurecom.fr (in case you created a private repository make sure to add eddy.ghabach@eurecom.fr as developer or maintainer).

Deliverable: at the end of this assignment, you have to fill and submit on moodle a file called “Clouds-Assignment5-Deliverable.docx”, which is provided to you in the assignment 5 section on moodle.