Hiking Trails API

General Requirements:

1. Must support firebase Google Auth.
2. App must store user data and app data securely on Firebase.
3. Use Angular structure to implement routes, components, services, HTTP calls, and observables.
4. App should have zero critical bugs and up to three small bugs (a critical bug is defined as one that blocks a user from using or completing a major feature in your app and for which there is no workaround).
5. Commit and push your work to GitHub daily.
6. Publish web app on public website.

App Specific Requirements:

The goal of the Hiking Trails API is to display different hiking trails related to a given zip code. Your app must have a minimum of four different pages, but you can add more pages/details than are listed here if you would like. It is up to you how you route between pages 2-4.

1. The first page (only page that order matters) that the user sees should be a login page to login with Firebase Google Auth.
2. The second page should use the 'getTrails' endpoint. Allow the user to enter a zip code, which should then be converted into latitude and longitude that can be plugged into the `getTrails` endpoint. After receiving a response from the endpoint, display the details about each returned trail. Optional - Display the overall rating of the trail from all user ratings, and display user comments on the trail as well.
3. The third page should use the 'getTrailsById' endpoint. If a user searches trails by zip code, and then clicks one of the trails, it should take the user to a page that only displays the details for that single trail. Use the weather API to also display the weather for the next 5 days along with the details of the trail (use the lat. & long. converted from the search page). User should have the ability to rate, comment, and click whether they have completed the hike before, and click if they're interested in the hike for the future.
4. The fourth page should be divided up into two sections. One section will be all of the trails that the user has already completed, and the second section will be all of the trails that they are interested in. Use the 'getTrailsById' to help request the associated data. Clicking any of these trails should be able to route to the trails details page. The user should have the ability to change a trail that they're interested in, to now being completed.

I would recommend (optional) using the Angular Material 'mat-card' to help display/separate each trail. ("https://material.angular.io/components/card/overview").

Setup:

Trails API

1. Go to "https://www.hikingproject.com/"
2. Click "Sign In" on the top right
3. After logging in, go to "https://www.hikingproject.com/data"
4. Store your API key (should be on the right side of the page) in a place that is inaccessible to the public

MapQuest API

1. Go to "https://developer.mapquest.com/plan_purchase/steps/business_edition/business_edition_free/register"
2. Enter required info
3. Go to "https://developer.mapquest.com/user/me/apps" to create/access an API key
4. Store your API key (click "my application" drop down) in a place that is inaccessible to the public

Weather API

1. Go to "https://openweathermap.org/api"
2. Click "Sign Up" at the top
3. Once signed in, click the "API keys" button next to the "Setup" button
4. Store your API key in a place that is inaccessible to the public
5. Go to "https://openweathermap.org/api" to see all API documentation

Endpoints:

The endpoints listed below are examples only and can be altered to the way you need them.

Trails API

1. Get trails by lat and long - "https://www.hikingproject.com/data/get-trails?lat=40.0274&lon=-105.2519&maxDistance=25&key=YourAPIKeyHere"

MapQuest API

1. Get latitude and longitude by zip code - "http://open.mapquestapi.com/geocoding/v1/address?key=YourAPIKeyHere&location=50010"

Weather API

1. Get Weather by latitude and longitude - "http://api.openweathermap.org/data/2.5/forecast/daily?lat=UserLatHere&lon=UserLongHere&cnt=5&APPID=YourAPIKeyHere"