What is BTS-Dashboard?

BTS-Dashboard is a website designed for the music enthusiasts. It primarily focuses on the BTS and BTS-related content, but its modular-based approach allows to use the website's assets for different music groups.

How does it work?

BTS-Dashboard uses Angular and Firebase.
BTS-Dashboard essentially has three screens: Public, Sign In, and user screens.

Screen by Screen Specs

Public View

- About Section
    - Wikipedia bio
- Play Spotify discography
- Twitter Feed
    - From @BTS_twt and @bts_bighit
- YouTube Uploads
    - From BANGTANTV channel
- Sign in button
- Navigation tabs
    - Home
    - My Dashboard
- Social Media buttons
    - Youtube, Spotify, Twitter (x2)
- Unified color scheme
- Dashboard layout
- BTS header

Sign In Page

- Choose between sign in method
    - Google, Twitter, Email/password

User View

- Users can sign in
    - Google, Twitter, Email/password
- Select preferred features for their dashboard
- Store these preferences in database
- See top 10 trending BTS tweets
- See how many users are engaged with BTS Twitter content within a timeframe
- Spotify track plays data visualization

Code behind the website

The following is the explication of code by the folders containing the code files.

functions folder contains all of the Firebase functions and classes that the product is using:

- DatabaseAgent and its extended classes SpotifyDatabseAgent and TwitterDatabseAgent are the classes responsible for storing and retrieving data from Firestore. It takes in a collection argument that is a name of the Firestore collection.
- getSpotifyClient and getTwitterClient are the functions that return the classes containing the necessary API addresses.
- TweetUpdater is a class that updates the given tweet collection with a passed in set of new tweets. The old tweets in the collection are being deleted.
- SpotifyTopTracks is a class responsible for getting top tracks from a given music band on Spotify in a particular country. It is also responsible for getting audio features of the track given this track's ID.
- index.js contains all of the aforementioned functions and classes being used and also being exported for the rest of the Angular project to use.

The src folder contains Angular components, scripts, and assets used by the project:

- app folder:
    - about section is a drop-down button that has a summary of who BTS are.
    - dash section is the template for the public page of the website that has containers to place other components such as twitter-card.

    - twitter-card is a component that displays the tweets from the database.
    - spotify-card is a component that displays BTS tracks which can be played.
    - youtube-carousel is a component that embeds the recent YouTube videos of BANGTANTV.
    - nav-bar is the bar used for navigation and conveying the user their status(like being logged in).
    - site-footer shows information about the authors of the website.
    - user and user-components have user-specific content, such as graphs for BTS Twitter and Spotify.
    - settings is a component designed to let the user have the preferences for the shown graphs. It lets the user enable/disable the specific graphs.

The scripts folder contains the JavaScript files needed to run the Firebase functions:

- ChartCreator class is used to create Google Chart graphs with the data and html element that are passed in.
- displayTweets is used to format , translate, and display the tweets. Data, html element, and a collection must be provided.
- FirebaseConfigs serves for Firebase logging in.
- PullFromSpotify activates the Spotify API functions that retrieves the BTS tracks.
- SpotifyDataPipeline and TwitterDataPipeline are designed to prepare the data for the charts to plot with specified features. Each file has a getData method from index.js of functions folder to get the raw data.

Resources

1. Embedding a Twitter feed by hashtag: https://embedsocial.com/knowledge-base/embed-twitter-hashtag-feed/
2. Embedding YouTube videos: https://developers.google.com/youtube/v3/quickstart/nodejs
3. Spotify-web-api library from GitHub by user thelinmichael: https://github.com/thelinmichael/spotify-web-api-node
4. Firebase authentication: https://firebase.google.com/docs/auth

5. Spotify API: https://developer.spotify.com/documentation/web-api/

- Specific twitter handles that publish BTS news

Twitter API Info
API key: qDX2Cdq7tUCDYE72KhP6xMP8O
API secret key: HX3NMzwkvcEdUiyBoJWH7uIibd2TgbPZTM1hrKtG5JsX3SeB8G
Bearer Token:
AAAAAAAAAAAAAAAAAAAAABgqOwEAAAAAdSkYa7Y%2BXeAOuB7C%2FJGFYDRzLRE%3DfKutNvzq1sLWUwJ7XIqbR9UrVM9D3IOK7P22v1WO1OO3kGx7vk

Spotify API info
ClientID: '26045288d9bc429c9517290636e7b4af'
ClientSecret: 'a8b4300bc8164b88b745c7cbe2e19b75'

Front-End Design templates & pieces
https://mdbootstrap.com/docs/angular/
Firebase Docs
https://firebase.google.com/docs
Angular Docs
https://angular.io/docs
Gradient underline
https://sharkcoder.com/visual/underline
Angular Material checkbox
https://v7.material.angular.io/components/checkbox/overview
Other Resources
https://stackoverflow.com/questions/47129368/angularfire-doesnt-load-app-root
https://stackoverflow.com/questions/62648670/deploying-angular-app-to-firebase-hosting-github-actions