What is needed to have a working BTS-Dashboard project on your computer?

BTS-Dashboard uses Angular and Firebase to bring its features to the user. In order to host the project on a local host and to deploy it, you must first install Visual Studio Code. Here is the link that goes through its installation: https://www.youtube.com/watch?v=MlIzFUI1QGA

After installing VS Code, proceed to the installation of Node.js: https://www.youtube.com/watch?v=31dRWcPcvhM

The next step will be to install Angular: https://www.youtube.com/watch?v=mMZOJ5z55hE

Finally, install Firebase: https://www.youtube.com/watch?v=6AX-fhx59Hg

After all of these steps, go to the project directory in the VS code. Do *firebase deploy --only functions* in the terminal.

Finally, do *ng build* and then *ng serve* in the terminal, and go to your local host.

If you want to deploy the project on the https://btsdashboard-d7ad5.web.app/, enter *deploy firebase* in the terminal in the project directory.

URLs for the used APIs:

https://developer.twitter.com/en/docs

https://developer.spotify.com/documentation/web-api/

https://developers.google.com/youtube/v3/quickstart/nodejs

https://developers.google.com/chart

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.
- 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.