**Difference between before and after**

Graphical user interface

Description automatically generatedThis was the initial wireframe for how the app would look

Graphical user interface, website

Description automatically generatedThis is how it turned out

Graphical user interface, text

Description automatically generatedI am glad at how my app turned out as I think the finished product looks neater than what I initially had planned. The first wireframe is trying to force everything onto the one page whereas the finished application had you first search for the artists on the home page.

Then after you an artist has been searched and clicked on; the app displays a list of the artists albums.

Graphical user interface, website

Description automatically generated

Then when an album has been chosen the application displays a list of songs from that album and allows the user to play a 30 second preview of the song and adjust the volume of the preview. The user can also press the view in Spotify button to redirect to the Spotify website showing the selected album.

A picture containing text, screenshot, indoor, computer

Description automatically generated

**Component Tree**

Components Folder

about

album

artist

home

navbar

search

shared/services

1. The about components only use is to demonstrate that routing is implemented.
2. The album component was used to display the albums of the artist that had been searched. Also shows the songs within the album and allowed for the audio to be played.
3. The artist component displays a list of artists when something is inputted into the search bar.
4. Home component is where the routing goes back to when home or the title is clicked in the navbar. Calls on the search component in the ts file
5. Navbar is used to display the home and about routing links on the navbar. Also displays the title.
6. Search component uses the Spotify web API to search for an individual artist which returns a list of artists.
7. The service ts file within shared/services is probably the most important file in the project which allows authentication by getting the token for the API and has most of the methods which are called upon in the html files which retrieve the JSON from the Spotify Web API.

**The main CSS classes used**

A picture containing text, screenshot, monitor, screen

Description automatically generatedThe main CSS class that I used for my application was Bootstrap. It is all within the html files of the project. I mainly used containers and cards to display the data retrieved from the API. I used cards to display the artists when searched

I also used cards to display the albums when an artist is selected

Graphical user interface, website

Description automatically generated

Once again made use of cards in bootstrap by displaying the songs in an album

A computer screen shot

Description automatically generated with low confidence

Reflection

Overall, I think my project went well. At the start I struggled a lot with the authentication of the Spotify web API. I could not get the API working which really hindered progressing with the application as I was not able to display the artists or albums without the API functioning properly. I learned a lot about typescript while doing my project such as writing methods in ts files and calling upon them in html files. I also learned about authentication in detail as I had to perform extensive research by watching videos and reading articles to learn how to gain access to the API. A few things I wish I could have done are getting the application to refresh the token, I could not get that part working so I took out the code to let the application work. Another factor as to why I did not get that part working was my poor time management when working on my app. I ran out of time to configure the refresh of the token. I also should have found a way to incorporate a database into my project which is why I have marked down my project in the marking scheme below.

**Marking Scheme**

Graphical user interface, application, table, Excel

Description automatically generated

I am happy with how my project turned out, but I feel the lack of a NOSQL database pulls the marks down a bit. Also, the fact you need to get a new auth token from (<https://developer.spotify.com/console/get-search-item/?q=tania%20bowra&type=artist&market=&limit=&offset=&include_external=>) every hour to allow the API to work, drags the mark down as well.

**Firebase Hosting**

<https://spotify-project-16f89.web.app>

The token will be expired when you see it.