Test Frontend

ENGLISH (2 WEEKS)

At Hosco, we like to listen to music at all hours, even while coding. For this reason we need you to develop a web application to be able to assess the technical part while still enjoying our favourite artists.

You have to create an SPA that consists in a simple navigation between two pages with React, Angular or Vue.

On the first page we would see a search bar where you can enter the terms (whether artists, songs, albums, genres ...). The search results should be listed on the same page, showing the song title and artist, and more in detail, the album title, release date, cover thumbnail, song length, genre and price. It should offer the ability to sort the list over these last three fields.

Each result should navigate to a second page in which, with a similar design to current music players, allows us to see the cover detail, basic information about the song and the basic controls to listen the song, play and pause, and skip to the previous and next song in the list of search results.

Furthermore, this detail page should have a button that allow us to share what we are listening in our favourite social networks.

If we go back, we should see the listing page with the query and result of the previous search.

Send the project in a zip file or upload it to Github/Gitlab (if the repo is public, please do not mention Hosco so we can reuse the test). We will not value the level of completion as primary, but mainly the architecture and code styling, as well as the adopted solutions, the use of design patterns or best practices will be what determines a good result.

To implement this test you will need to use the iTunes Search API:

https://affiliate.itunes.apple.com/resources/documentation/itunes-store-web-service-search-api/

Here you will find a lot of documentation in addition to the necessary, so you can go directly to the sub-section called "Search Examples", and verify that you only need to use a call like “https://itunes.apple.com/search?term=nirvana” for a search of the legendary Nirvana songs.

You will not need to use any additional parameters in the call. You can see at the bottom of this doc a response example (we have removed unnecessary fields).

One more thing: code should run in docker, this means that a Dockerfile should be included with the code/repo !!!

IMPORTANT:

- There is no time limit to deliver the result, and time will be taken into account
- Tests are more than welcome
- Please include a README.md file with clear instructions so we can build/run the app

```
{
  "resultCount": 1,
  "results": [
    {
      " wrapperType ": "track",
      " kind ": "song",
      " artistId ": 32940,
      " collectionId ": 850697616,
      " trackId ": 850697815,
      " artistName ": "Michael Jackson & Justin Timberlake",
      " collectionName ": "XSCAPE (Deluxe)",
      " trackName ": "Love Never Felt So Good",
      " collectionCensoredName ": "XSCAPE (Deluxe)",
      " trackCensoredName ": "Love Never Felt So Good",
      " collectionArtistName ": "Michael Jackson",
      " previewUrl ":
"http://a599.phobos.apple.com/us/r1000/049/Music/v4/50/f0/1c/50f01ccf-4d
d4-eea0-1c94-7d9ce591fdfb/mzaf_6052792677521696178.plus.aac.p.m4a",
      " artworkUrl30 ":
"http://a1.mzstatic.com/us/r30/Features2/v4/f4/25/67/f425676c-bebd-611b-
226c-ed4b6e8675f1/dj.sxghjyyn.30x30-50.jpg",
      " artworkUrl60 ":
"http://a5.mzstatic.com/us/r30/Features2/v4/f4/25/67/f425676c-bebd-611b-
226c-ed4b6e8675f1/dj.sxghjyyn.60x60-50.jpg",
      " artworkUrl100 ":
"http://a3.mzstatic.com/us/r30/Features2/v4/f4/25/67/f425676c-bebd-611b-
226c-ed4b6e8675f1/dj.sxghjyyn.100x100-75.jpg"
    }
  ]
}
```