

Frontend - Final Project Assignment

Problem Statement

Gigih adalah seorang anak muda yang menyukai musik. Hampir setiap hari dia mendengarkan musik di Spotify dari berbagai genre. Dia ingin membuat *playlist* (daftar putar) yang berisikan lagu lagu yang sedang ia gandrungi. Bantulah Gigih dengan membuat sebuah aplikasi berbasis web sehingga Gigih dapat membuat *playlist* dengan mudah.

Di aplikasi tersebut, Gigih ingin agar aplikasi tersebut hanya dapat diakses oleh orang yang sudah *log-in* terlebih dahulu. Jika Gigih belum *log-in*, maka Gigih tidak dapat mengakses halaman untuk membuat *playlist* tersebut, dan akan di-*redirect* ke halaman login.

Ketika selesai login, maka Gigih ingin melihat halaman untuk membuat *playlist*. Pertama tama, ia ingin agar dia bisa mencari lagu kesukaannya. Ia ingin agar informasi tentang foto album, judul lagu, nama penyanyi, dan durasi lagu muncul di hasil pencariannya. Ia juga ingin memilih lagu yang nantinya ingin dia jadikan *playlist*. Ketika ia mencari dengan kata kunci yang lain dan menemukan lagu yang sudah pernah ia pilih, ia tidak ingin lagu tersebut bisa dia pilih lebih dari satu kali. Ia juga ingin dapat membuang pilihan lagu yang sebelumnya dia pilih.

Jika sudah selesai memilih lagu, Gigih ingin agar ia bisa membuat *playlist* dari pilihan lagu yang sudah dia pilih sebelumnya. Gigih juga ingin selalu menuliskan judul dan deskripsi dari *playlist* tersebut. Untuk judul, ia ingin agar judul dari semua *playlist* yang dia buat menggunakan aplikasi ini, memiliki panjang minimal 10 karakter. Ia juga ingin agar *playlist* yang dibuat selalu merupakan *private playlist* dan bukan merupakan *collaborative playlist*.

Ketika *playlist* sudah berhasil dibuat, Gigih ingin mengetahui bahwa *playlist* tersebut sudah berhasil dibuat.

Technical Requirements

1. **Hooks:** terdapat minimum satu React component yang menggunakan hooks
2. **Flex & Grid:** terdapat masing - masing minimum satu penggunaan flex dan grid pada aplikasi kamu.
3. **Typescript:** terdapat minimum satu Typescript Type dan satu file Typescript yang aktif digunakan.
4. **Redux:** terdapat minimum satu *store*. *Store* ini harus aktif digunakan pada aplikasi.
5. **Eslint:** tidak terdapat satu pun warning eslint pada keseluruhan kode kamu.
6. **Testing:** terdapat minimal satu *test case* yang meliputi `toBeInTheDocument()` dan `userEvent` (mensimulasikan user interaction). Juga aplikasi kamu harus lolos test yang sudah kamu buat.

7. **Documentation:** terdapat readme yang minimum mengandung beberapa poin dibawah ini. Sebagai referensi readme, kamu dapat melihat website berikut ini <https://www.makeareadme.com/>
 - a. Nama aplikasi dan deskripsi
 - b. Fitur yang kamu implementasi. Tuliskan juga fitur fitur bonus diluar requirements yang kamu implementasikan.
 - c. Cara install dan cara menjalankan aplikasi kamu.

Aturan Main

1. Aplikasi yang akan kamu buat adalah aplikasi berbasis React.
2. Kamu dapat menggunakan UI Library apapun. Buatlah aplikasi kamu seindah mungkin, tentunya kamu ingin Gigih menyukai apa yang kamu buat.
3. Pastikan bahwa aplikasi kamu sudah di-deploy di Internet dan bisa diakses oleh siapapun. Whitelist pula akun Gigih supaya ia bisa mencoba aplikasi kamu.

Panduan Pengumpulan

1. Batas akhir pengumpulan adalah **22 April 2022 23:59 WIB (UTC+7)**.
2. Dilarang untuk melakukan perubahan terhadap kode setelah batas akhir pengumpulan.
3. Tambahkan akun Gigih di Spotify App Developer Console kamu pada menu *Users and Access*, sehingga Gigih nantinya dapat mencoba aplikasi kamu. Email: gigih2frontend@gmail.com
4. Kumpulkan hal hal berikut ini pada link submisi <https://bit.ly/gg-fe-finalproject>
 - a. URL *production* dari aplikasi kamu.
 - b. Link repo Github. Pastikan repo kamu adalah *public repository*.
 - c. Compressed (.zip format) dari kode aplikasi kamu

Bonus:

Kamu akan mendapatkan bonus poin jika kamu melakukan hal dibawah ini

1. Menambahkan satu fitur di luar problem statement. Salah satu contohnya adalah Gigih dapat melihat profile picture mereka. Buatlah sekreatif mungkin, dan tuliskan pada readme apa saja yang kamu tambahkan, jika kamu tidak menambahkan pada readme maka kamu tidak akan mendapatkan nilai bonus.
2. **Testing:** menggunakan MSW untuk mock API async call

Kriteria Penilaian

1. Aplikasi kamu harus bisa memenuhi *problem statement* dan *technical requirements* di atas.
2. Clean code dan struktur kode kamu juga akan dinilai. Struktur folder dan penamaan juga harus konsisten. Akan dinilai pula kemampuan kamu untuk mengabstraksi problem, menjabarkannya menjadi komponen-komponen yang sensible (tidak terlalu nested), testable, extensible, dan maintainable.