



TECH SCREENING FRONT-END

CASUS

TECH SCREENING FRONT-END

1	INTRODUCTIE	3
2	FRONTEND	4
3	QUALITY ASSURANCE	4
4	BACKEND	5
5	FOCUS	5
6	EXTRA REQUIREMENTS	6
7	RULES	7

INTRODUCTIE

Het leven van een IT Rockstar gaat natuurlijk niet over rozen. Zowel zakelijk als privé hebben wij dagelijks te maken met de meest uiteenlopende irritaties. Als Rockstar in hart en nieren doe je er natuurlijk alles aan om die ongemakken weg te automatiseren en als de wereld daar ook beter van wordt is dat uiteraard mooi meegenomen.

Daarom bieden we jou graag een technische challenge aan! De uitdaging is als volgt:

Zoals onze naam doet vermoeden hebben we een bijzondere link met muziek, en dan vooral keiharde Rockmuziek. Dat genre is enorm uiteenlopend en heeft een prachtige historie.

Om orde op zaken te kunnen stellen willen we graag onze playlists kunnen samenstellen voor elk moment van de dag. Van onze IT Rockstars wordt verwacht dat ze zo'n app snel en kundig in elkaar zetten, dan maak je altijd een top indruk bij je collega's en vrienden! Aan jou dus om de volgende specs te vertalen naar de nieuwe app™

We verwachten dat de app te openen is met Visual Studio Code en dat de frontend is geschreven in Javascript of Typescript. Je mag eventueel je eigen favoriete framework kiezen. Lees alle specificaties goed door voordat je begint met de ontwikkeling van de applicatie.

Good luck and may the force be with you.

FRONTEND

De frontend moet een webapplicatie worden waarmee een IT Rockstar zijn of haar eigen favoriete muziek kan beheren.

Specificaties:

- › De web applicatie moet in de huisstijl van Team Rockstars IT zijn en uiteraard goed op een mobiel apparaat werken.
- › Een IT Rockstar kan een overzicht van alle artiesten inzien. Met een zoekbalk kan de IT Rockstar snel zijn favoriete artiest vinden.
- › Een IT Rockstar kan een artiest detail pagina met alle nummers van de artiest opvragen.
- › Een IT Rockstar kan playlists aanmaken en verwijderen.
- › Een IT Rockstar kan nummers aan een playlist toevoegen.
- › Een IT Rockstar kan een playlist inzien met alle toegevoegde nummers. Hier heeft de Rockstar ook de optie om een nummer uit de playlist te verwijderen.

De benodigde bestanden vind je hier:

<https://www.teamrockstars.nl/sites/default/files/db.json>

<https://www.teamrockstars.nl/sites/default/files/RST-073-logo-Team-Rockstars-IT.png>

https://www.teamrockstars.nl/sites/default/files/styleguide_1%20interactive%20elements.png

https://www.teamrockstars.nl/sites/default/files/styleguide_1%20static%20elements.png

QUALITY ASSURANCE

We zien nu al aankomen dat dit een super populaire app wordt. En bij een veelgebruikte app horen goede unit tests en goed leesbare code. Immers gaat deze applicatie nog lang onderhouden moeten worden door mede-developers. We verwachten dan ook unit tests voor de frontend & duidelijke naamgeving en documentatie waar nodig.

Bovendien kunnen we het waarderen indien er e2e tests geschreven zijn om de kwaliteit nog beter te kunnen waarborgen.

BACKEND

De backend bestaat uit een JSON bestand dat eenvoudig uitgelezen kan worden. In dit JSON bestand zijn artiesten en nummers terug te vinden.

- › Met *JSON Server* is het zeer eenvoudig om binnen 30 seconden een lokale API op te zetten. De data uit zowel *artists.json* als *songs.json* kan vervolgens lokaal opgehaald worden met een GET request. Zie: <https://github.com/typicode/json-server>
- › Het opslaan van de playlist mag met *JSON Server*, maar het mag ook in de browser door middel van *Local Storage*.

Voorbeelden van get requests zodra JSON Server is opgestart:

<http://localhost:3000/songs>

<http://localhost:3000/artists>

<http://localhost:3000/songs?genre=Rock>

<http://localhost:3000/songs?artist=Spinal%20Tap>

FOCUS

De focus van deze hele exercitie ligt op de functionaliteit. Daarnaast ontvangen we graag ook vooraf toelichting over bepaalde shortcuts die genomen zijn en een aanpak/ architectuur keuze/ strategie die je gebruikt hebt.

Verder wordt alle code beoordeeld op de volgende aspecten:

- › Duidelijke documentatie in de vorm van een *readme* van hoe de app werkt en gestart kan worden.
- › Wordt er op een juiste wijze gebruik gemaakt van een framework of library zoals Angular, AngularJS, React, Vue, jQuery of een ander framework naar keuze.
- › Code style
- › Performance
- › Huisstijl en opbouw (S)CSS
- › We verwachten geen code coverage van 100%. Acht tot tien unit tests en enkele e-2-e tests zijn voldoende
- › Genereren van file bundles door middel van een build tool
- › Gebruik een UI-Kit om jouw spaarzame tijd goed te besteden.

EXTRA REQUIREMENTS

Om jou als potential IT Rockstar extra te challengen, willen we jou vragen om uit onderstaande lijstje nog een aantal extra requirements te selecteren die je toevoegt in jouw applicatie. De keuze voor de requirements is vrij, het aantal niet.

Zonder te veel in hokjes te denken...hieronder een indicatie voor de aantallen. Meer mag altijd!

Junior: minimaal 1 extra requirement
Medior: minimaal 3 extra requirements
Senior: minimaal 4 extra requirements

Waar moet je dan uit kiezen? Zie hieronder:

› Responsive applicatie

Zorg dat de applicatie er op mobiel, tablet en desktop netjes uit ziet.
Je mag fluid design gebruiken of breakpoints.

› Vang onbekende urls af in routing

Zorg dat het navigeren naar een onbekende pagina in je applicatie netjes wordt afgevangen. Je mag zelf invullen welke informatie erop getoond wordt.

› Gebruik een grid systeem

Zoals flexbox, CSS grid, 8-point-grid of een eigen opgezet grid-systeem.

› Infinite scrolling / paged scrolling

Voorzie de artiestenpagina met een soort van paginering.

› Nette error afhandeling backend calls

Zorg dat een error vanuit een backend call netjes wordt afgevangen en teruggekoppelt naar de gebruiker. Hoe dit eruit ziet in de app, mag je zelf bepalen (alert, pop-up, etc).

› Caching (minder backend calls, snelle performance laadtijd etc)

Zorg dat data gecached wordt en dit bij een volgende bezoek op dezelfde pagina wordt uitgelezen (mits aanwezig). Bij geen bestaande cache, wordt er een backend call gedaan.

› Voeg een button animatie toe

Op click, hover of focus: laat de button iets doen. Geanimeerd met CSS.

› Gebruik een CSS preprocessor

Implementeer een preprocessor, zoals SASS/LESS en schrijf je styling in deze taal.

› Implementeer design patterns

Zorg dat je mappenstructuur en bestanden design patterns reflecteren.
De indeling en architectuur van je bestanden, styling, componenten etc.

› Bouw een loginpagina

Maak een inlogpagina op basis van Facebook, Google of email/wachtwoord.

› Koppel de nummers met Spotify

Zorg dat er een soort van koppeling is tussen de nummers en Spotify.

In de aangeleverde database data staat een Spotify trackID die je kunt gebruiken.

Link naar de Spotify WEB API: <https://developer.spotify.com/documentation/web-api>

Maak een developer account aan met een Spotify free account.

RULES

Ondanks dat wij niet van regels en verplichtingen houden, zijn er toch een aantal zaken waar je rekening mee moet houden:

- › Je mag maximaal 4 uur van jouw kostbare tijd aan de casus spenderen. Op deze wijze krijgen wij het meest realistische beeld van jouw skills, vaardigheden en snelheid.
- › Het is overigens geen schande als je het binnen deze tijd niet 'af' krijgt!
- › Kwaliteit is in deze beter dan kwantiteit, dus zorg dat hetgene dat af is ook goed is.
- › De applicatie moet met simpele commando's te runnen zijn en voorzien zijn van duidelijke instructies hoe dit te doen.
- › Je levert de solution minimaal 48 uur van te voren aan bij iemand van de techboard. De gegevens van de techboarder worden verstrekt door de Agent die jouw kwalificatietraject begeleidt.
- › Na het gesprek volgt er een definitieve GO / NO GO vanuit de Rockstar! Bij een GO ontvang je binnen 48 uur een contractvoorstel vanuit jouw Agent!



WWW.TEAMROCKSTARS.NL