

Cover Letter

I am a Front-End React Developer, my particular strengths lie in building out UI and UX Front-End Applications with React, I am ambitious and forward thinking, resourceful and flexible.

Explain why you are a good fit for this position

- I really value and embody integrity, quality and reliability.
- I work well in a team, I learn really fast.
- Hungry to learn and willing to adapt.
- Highly ambitious and love working with people.
- Sharp and forward thinking, and able to take initiative where required.
- Creative, focused and constructive.

Name three things in your resume we should not miss and why.

My minified Sirion Cloud App I built with Serverless

- This app has been built with, React, CSS, AWS Amplify, React Router, Bootstrap and a Deezer API.
- This app has been built with stateless React components, React hooks in an attempt to move away from class components and AWS Amplify.
- The React hooks - "useContext" and "createContext" are used to set a global state and manages the applications Authentication in this way.
- React Router is used to keeps the UI and the URL in sync.
- The Application's main components are wrapped in functions that grant or deny access to its's child components based on the global authentication state variable's value, if the function returns false the user is redirected to a login page, this has been set up using AWS Congnito service.
- There is a "useEffect" hook in use that allows users to stay logged in when a page refresh occurs. I created a custom hook that manages user-form-field inputs in my application.
- There is a music-genre-bar on the "SearchTracks" component that communicates with a Deezer API and returns data-objects which is then integrated into the DOM. I created a component wrapper that handles all the routes which has made the routing of this application very easy.
- Apart from our Auth integration using AWS Cognito we also use another service called DynamoDB which is a database the user will use to store they're favourite tracks.
- Upon login the user will find a "home page" which will query their unique database bucket if it exists and display everything in there, here and in the component that handle's creating new database entries (SearchTracks.js) you can also find delete functionality that allows user's to remove tracks from their buckets.

My Sirion-Cloud app I built out with React.JS.

- This app has been built with React, Sass and a Deezer API
- Sirion Cloud uses a Deezer API and returns data-objects which is then integrated into the DOM, the UI and UX of this site is fresh and eye catching.

- This application has been built with stateful components which uses props to pass state to other components.
- This application contains in short a home-component, album-component, tracklist-component and a favourites-component.
- I have used JavaScript's fetch method to make requests to the Deezer database.
- You will find a music player that keeps track of the current music-item selected across the board, while it can also skip back and forth, I built this out from scratch.
There is also multiple functions that check and make sure that the music player is in sync with the applications state, for instance: it relies on a 'onplay' and 'onpause' event listener to ensure that the UI of this player is in sync with whats been selected by the user.
- I have made use of React Refs in the app to also sync users input with the app's main state, for instance a html input field within a child component passes information to a top level component and then sets the new state and returns a function to call on our Deezer API.
- I have created a module that contains all our fetch functions, this module is exported and used by the home-component of the application and returns requested music data.
- Sirion Cloud makes use of inline class-name queries to ensure that the UI of the app is reflecting what's occurring with the main state of our application.
- I have created a favourites component that acts as a storage facility for the user's tracks that he/she will like.
- Sirion Cloud uses state to also keep track of which songs where selected as favourites and reflects that in the UI throughout the users session.

My Movie Database Application

- This app has been built with React, Sass, React Router and a film database API
- The Movie Database App is an example of your typical google play store application.
This application has been built with stateful components which uses props to pass state to other components.
- React Router is used to keeps the UI and the URL in sync.
- Here I make use of "The Movie Database" API.
- I have stored all my API fetch functions in a module.
- Here we have made use of React's Ref attribute to access a top level component from a child component's "input field".
- I have used React's JSX syntax to inject component-state-data-objects into html components.
- Here is a basic run though of the applications build.
 - 1.** The App component renders a SearchBar and FilmDisplay component, and here I pass functions and state as props the child components.
 - 2.** FilmDisplay is assigned a route of "/" making it the default-route/
homepage, we then get redirected by means of a React-Router Redirect feature and the url changes to /film/:id, which also becomes the path to FilmDisplay's child that actually contains all the UI components that reflect the API data, this way assigning the URL route attribute "id" to the data-object's id selected by the user we can keep track of our routing.
 - 3.** FilmInfo is rendered by FilmDisplay and when this component is Mounted a API call fires and fetches the default-data of our home page along with the corresponding id attribute of this object that the API function returned, which is then assigned to our route's "id" attribute.
 - 4.** When a user uses the SearchBar component to search for film titles we access a module containing all our API fetch methods, we do this by using a "onChange" event listener on a input-html-component in SearchBar.

5. Once our search function has done its job and returned an array of data-objects the SearchBar component will display the array, once a user selects an item from the list the state and UI of FilmInfo changes by means of an onClick event listener.

Personal Information

First Name / Surname

Jean Charl Greeff

Email

greeffjean@gmail.com

Nationality

South African

Date of Birth

1994/05/09

Gender

Male

Work Experience

Dates

Jan 2019 - Current Date

Position

Freelance Developer

Experience

I provided the following clients with websites:

1. AquaHound
2. Rousant International
3. Casa Ponta Nduvene (Private Entity)

Dates

2018 - 2019

Position

Entrepreneur

Experience

1. Ran a Health Food stall at Earth Fair and Gardens Market in Cape Town
2. Co-owned and managed a small scale health food company. This entailed Purchasing of products, packaging and distribution.

Dates

Jan 2018 - Aug 2018

Position

Surf Coach

Experience

Teaching, Client Relations

Employer / Address

Surf Emporium 66 Beach Rd, Muizenberg, Cape Town, 7945

Dates

January 2013 - February 2018

Position

Freelance Visual Artist

Experience

1. I have been commissioned by several clients both locally and abroad.
2. I have been awarded an Arts Residency Scholarship by AMEXCID.
3. I have exhibited:
 - 3.1 Twice abroad - Puerto Vallarta, Jalisco Mexico.
 - 3.2 Solo Exhibition, Mexican Embassy Pretoria, Pretoria, South Africa
 - 3.3 Solo Exhibition Alliance Francais, Cape Town, South Africa

Education and Training

Dates

2018 - Current Date

Principal Studies

HTML5, CSS3, Javascript, React, JQuery, Git, Command Line, API Interactions, Responsive Design, SASS, Chai, Mocha, AWS, Seed, Netlify.

Qualification

Certificate

Institution

Codecademy, New York

Dates

2017 - 2018

Principal Studies

Intensive study of - Colour, Perspective, Line, Composition, Drawing, Painting, Visual Art Elements.

Qualification

None

Institution

Vivian van de Merwe (Private Tuition - Stellenbosch University Lecturer)

Dates

2008 - 2012

Principal Studies

Afrikaans, English, Maths, Life Orientation, CAT, Hospitality, Visual Art
Qualification

National Senior Certificate

Institution

Jan van Riebeeck High School

Languages Spoken

English / Afrikaans