Curtis Daniel **Larsen**

+64 027 580 7600 | curtis.larssen@gmail.com | curtisdaniel.com | github.com/CDLar | LinkedIn

EXPERIENCE

Rusticated LLC **Frontend Software Engineer, Various** Sep 2020 - Present, Pennsylvania, PA

- Responsible for the frontend design and creation of a support ticket interface for users and admins
- Working remotely with another dev to connect the ticket and message systems to the backend API

TECHNICAL PROJECTS

Twitter Guessr https://twitter-guessr.netlify.app/

Multiple choice quiz application based on Twitter API. Has access to Twitter Dev privileges to request and refresh data from the official Twitter API. Utilizes several libraries that interact with Twitter widgets.

- **React** – Utilized several custom hooks, as well as built-in hooks to track user input, high scores, current game state, and user preferences as well as add them to local storage for future visits
- **Javascript** – Used JS scripts along with Twitter Dev access tokens to access official Twitter API and make data requests to refresh the historic quiz database as well as grab new Tweets for the daily quiz
- **Styled Components** – Global styles was used for the baseline app styles and Theme Provider was used to handle dynamic switching between dark and light themes for a better user experience

Github Battler https://githubbattler.netlify.app/

Dual-purpose app built following Ui.dev. Can load in the most popular repositories from the Github API based on programming language or choose two Github users to battle based on their Github profile statistics

- **React Router** – Utilized switches and routes to keep the two different apps separate from each other, players selected for battle app would be stored to the URL path for easy saving/sharing of results
- **Error Catching / PropTypes** – Several error checks during repo loading as well as in the functions themselves to ensure any errors are caught and prevent the user from inputting invalid queries
- **Responsive Design** – Simple but responsive design utilizing flexbox to ensure that the design remains consistent across multiple viewports and screen sizes

Bank Design Template https://demo-bank.netlify.app/

Design template for a desktop banking application, it hosts a variety of features such as language select, user dashboard, a login system and a real-time currency exchange window.

- **React** – Multiple context hooks were used to cleanly manage various information and states within the app. The page would then be dynamically rendered based on login state, selected language, user, etc.
- **Axios** – Used along with useEffect hooks to make requests to an exchange rate API based on active currency (USD default), multiple requests would be used to calculate the difference between dates
- **Material-UI Framework** – Used the MUI framework integrated with Styled Components for most of the design, including use of the API to create custom themes as well as style overrides for components

Fantasy Sports App https://gm-stats.netlify.app/

Fantasy sports application that tracks fantasy league statistics/records as well as player ownership spanning over multiple years, the data is collected and exported in CSV format from Fantrax.com.

- **React Context API** – Extensive variety of user-selectable themes based on professional sports teams that dynamic recolor the entire app and save the user's preference in local storage for future use
- **Javascript / CVS** – Due to the data being exported in CSV format, JS scripts are used to first convert it into JSON format so it can then be mapped and filtered and finally rendered to the UI
- **React Custom Hooks** – Use of custom React hooks to implement tooltips that display more statistics to further improve the user experience as well as manage other functionalities such as state toggling

EDUCATION

Carleton University **Integrated Science (B.Sc.)** Sep 2012 - Apr 2015, Ottawa, CA