Mike Barberry

mikebarberry@protonmail.com

github.com/mikebarberry

(971) 977-2927

TECHNICAL SKILLS

Languages: JavaScript, Python, Go, Bash, SQL, HTML/CSS, MongoDB QL
Libraries/Frameworks: React.js, Node.js, Next.js, Pandas, NumPy, Plotly, Scikit-learn, Pygame
Tools & Platforms: Git, Linux, Docker, Nginx, AWS (Lambda, S3, ECS, CloudFormation, IAM, etc.)
Management + Design: Azure DevOps, Jira, Figma

PROFESSIONAL EXPERIENCE

Digital Infuzion Remote
Software Engineer Dec 2022 – Present

- **Flu Hub -** Engineered a full stack NIH website to connect influenza researchers with relevant information. Built the frontend using React.js and Next.js. Created and maintained the headless CMS (Strapi) backend. Wrote Dockerfiles and AWS CloudFormation templates for both ends. Deployed to AWS ECS after generating IAM permissions and CloudFormation stacks.
- **iDPCC Data Search -** Developed a complex data search page for a NIH website. Discovered a novel approach to React state management utilizing renderProps, the children prop, context, custom hooks, and "lifting state up" to create a single root component that reflects the DOM structure of the rendered page, isolates HTML and CSS in wrapper components, allows state to be passed easily to anywhere in the component tree without a global store, and makes the JavaScript logic easy to read and understand.
- **Impact Analysis Parallel Coordinates -** Cleaned and processed fragmented data from multiple files into a format consumable by a frontend. Researched and identified a little known yet useful library from the Facebook research team called HiPlot. Implemented a parallel coordinates chart using HiPlot with custom coloration, table columns, and extended it to include an additional feature that calculates the mean and median values of selected data. Built with Parcel and deployed on AWS S3 as a static website.
- **ANDI Scraper -** Engineered a Node.js automation tool that uses Puppeteer and ANDI (a Social Security Administration library for Section 508 accessibility compliance) to collect and report on ANDI alerts: It accepts either a text file of URLs or a single URL to a sitemap that it parses for individual URLs. It then uses Puppeteer to automate clicking buttons, and uses CSS selectors and JavaScript to identify relevant parts of the page and collect content. During the process it uses CSS classes to highlight the related DOM element causing an alert and takes a screenshot. At the end of the program, it generates a pretty PDF report and developer logs that associate the screenshots with individual alerts.
- **Impact Analysis NLP and Scatter Plots -** Created a Python program that reads from an Excel file, collects data from certain columns, and performs NLP on one of them: It uses two different custom implementations of frequency algorithms (bag-of-words and TF-IDF), and two different scikit-learn reduction algorithms (UMAP and PCA) to generate various scatter plot versions. Developed a front end using HTML, JavaScript and DOM APIs to display each option as a list item and open the plot in a new page on the click event. Deployed the frontend as a static website to S3.

Competitive Solutions Remote
Software Engineer April 2019 – Dec 2022

- **NanoID -** Implemented feature to display reduced IDs to users: Researched libraries and decided on NanoID. Integrated their functions into the database model and set up automatic generation for new users. Created a Node.js script to populate existing user data with NanoIDs. Modified backend API endpoints to accept and send the new, shortened IDs along with the rest of the payloads. Updated the frontend React code to display the shorter IDs.

- **Document Upload -** Engineered a solution to allow admin to upload documents and attach them user data: Created React frontend with multiple steps where admin could select a file from their local computers, choose a priority level, add a message, and select other pre-filtered users from a dropdown to send an email notification to. Developed the backend APIs to perform additional business logic such as sending the actual emails, perform the filtering of which users to offer as email recipients, uploading the files to AWS S3 and updating the MongoDB database collections. Created a new MongoDB database collection to store document metadata and implemented a solution similar to a left-join in SQL where user MongoDB collections referenced the ObjectID of the appropriate documents. Developed a MongoDB middleware hook that interacted with both collections to perform updates as necessary, for example when deleting a document.
- **Data Coordination -** Created Node.js AWS Lambda functions, AWS IAM permissions and S3 buckets to facilitate secure data exchange with a partner organization.

PERSONAL PROJECTS

MikeBarberry.com Next.js website with 3D animation | **catfacts** Multithreaded, low latency Python cat program | **Quote Generator** Class based JS for TV quotes | **GoTasker** Golang to do list

EDUCATION

Flatiron School **Bootcamp:** Software Engineering 2019
University College London **MA:** Philosophy 2016
Indiana University Bloomington **BA:** Neuroscience 2015