ARYAN MALIK

arryan@umich.edu | (608)982-1695 | aryanmalik.com | www.linkedin.com/in/arryan | https://github.com/arryan135

EDUCATION

University of Michigan **Ann Arbor, MI**
B.S. in Computer Science *Aug 2020–Dec 2023*

- *GPA*: 3.91/4.00; University Honors, Dean's List
- *Completed*: Programming and Intro Data Structures • Discrete Math • Multivariable Calculus • Data Modeling I & II
- *Currently Enrolled*: Data Structures and Algorithms • Foundations of CS • Linear Algebra • Database Mgt. System

PROGRAMMING SKILLS

Languages: C++, Java, Python, Javascript(includes ES6), HTML, CSS, SASS, TypeScript, SQL, R, MATLAB
Frameworks: ReactJS, NextJS, NodeJS, Jest, Mocha, Webpack, Esbuild, JSS, MaterialUI, Bulma, Bootstrap
Technologies: Git, Mongoose, MongoDB, Redis, Redux, Express, JSON Web Token, Socket.io, RESTful API

PERSONAL PROJECTS (complete list at https://github.com/arryan135)

CheatBook. *A CLI for an interactive coding environment that simplifies the creation of programming-based cheatsheets. It receives 30+ downloads/week at peak traffic.*

- Built a React+Typescript frontend that uses Esbuild, a JavaScript bundler, to automatically transpile user's code and safely execute it within HTML IFrame in the browser. Optimizes performance through web assembly and caching to ensure code execution time is as fast as that of any traditional IDE thereby enhancing user's productivity.
- Refactored CLI to use Redux and React Hooks to handle complex state changes. Utilizes a Redux middleware and Express server on a NodeJS backend to persist cell contents on a local JSON file.
- Uses Lerna to optimize the workflow of managing the CLI's multi-package architecture that is based on a RESTful NodeJS API which uses HTTP GET and POST methods to read and save the cell data. Resolved undefined problems by applying Node's resolution algorithm to connect internal packages with absolute links instead of symbolic ones.

PerfLoad. *A real-time performance monitor that shows a local machine's CPU load, and memory usage.*

- Collects performance data through a NodeJS Server that is internally connected to a Socket.io Server via TCP/IP.
- Ensures performance optimization and scalability using NodeJS Cluster module so that multiple Node Clients can run seamlessly on multiple threads. It uses Redis(in-memory database) to prevent unnecessary new socket connections when a Node Client reconnects.
- Designed this multi-tiered, distributed backend system to also delegate a MongoDB server to eradicate the possibility of duplicate machines on the database before a final Socket Server connection is established with the ReactJS Client that displays the data using HTML5 canvas animations.

Palette Splash. *A web app that helps in creating personalized color palettes to generate color themes for other websites.*

- To make palette creation user-friendly, the website implements React Sortable Higher-Order Components(HOC) to create an invisible grid that allows the two-dimensional movement/reordering of the color boxes.
- Build upon React Pure Components to ensure performance optimization by preventing unnecessary renders.
- Designed using components from MaterialUI and custom JSS styles to avoid specificity conflicts.

IG DB Clone. *A clone of Instagram's database made primarily using mySQL.*

- Uses foreign keys to establish a many-to-many relationship among SQL Tables that represent Instagram users, photos posted by the users, users' comments, likes for users' photos, users' followers, and tags for the users' photos.
- Integrated mySQL database with Javascript, to populate the tables with artificial data entries.
- Implements SQL Data Triggers to prevent data entries to tables when a user attempts to follow itself.

EXPERIENCE

EECS 280 Intro Data Structures, University of Michigan **Ann Arbor, MI**
Supplemental Instruction (SI) Leader *Aug 2021–Present*

- Aid students in building conceptual clarity of fundamental data structures concepts through weekly sessions.
- Spearhead the development of weekly project-support sessions to help students understand the project specifications and coach them on optimal implementation strategies in a collaborative learning environment, thereby expanding access to Computer Science education at the university.
- Organized weekly office hours to resolve IDE and logic errors while debugging coding problems in student projects.

UW Madison Material Science Engineering **Madison, WI**
Research Scholar *Jan 2020–Sep 2020*

- Worked within a team of 4 to design and build an automated, open-source toolkit with analysis routines that are central to understanding the performance of machine learning models as they pertain to materials research.
- Used linear programming to show that the K-Medoids algorithm optimizes memory consumption while generating data clusters to run predictions and subsequently implemented this algorithm on the framework using Python.
- Implemented ElemNet (deep neural network algorithm) and IRNet (deep regression network) to overcome the technical challenge of leveraging AI that captures the chemistry of elements to predict their material properties.
- Analyzed miscellaneous performance data using NumPy and visualization libraries like Matplotlib, Seaborn, and Plotly to make model performance visuals more informative.