

Hamza Amar

Software Engineer

Calgary, Canada (Open to Remote) | (403) 991-5056 | rhamzax@gmail.com
[linkedin.com/in/hamza-amar](https://www.linkedin.com/in/hamza-amar) | github.com/rhamzax

TECHNICAL SKILLS

Front End and Back End | Python, C, C++, Java, HTML5, CSS, JavaScript, SQL, React, Typescript, , NextJS Terraform, AWS
Relevant Courses | Prob Stats & Machine Learning, Digital Circuits, Programming Fundamentals for C and C++, Practices for Database Management and Design, Pract'l Data Struct Algorithms, Full Stack Web Development, Object-Oriented Princp'l Devpt
Developer Tools | Proficient in Git for version control and collaboration, Github

SOFTWARE ENGINEERING APPLICATIONS/EXPERIENCE

Schulich Racing | TypeScript, React, Node.js, Firebase, Flask, Python3, PostgreSQL September 2022 - Present
Full Stack Software Engineer

- Utilized React and TypeScript to implement a modular and component-based architecture, advanced features such as server-side rendering and code splitting to improve website performance and SEO, and built reusable UI components.
- Integrated Firebase to handle email inquiries for the website, utilizing Cloud Functions and Firestore for real-time inquiry management and automated responses, improving the overall efficiency of the inquiry management process.
- Implemented a Flask-based back-end for the website, allowing for efficient data retrieval and manipulation. Developed endpoints to receive data from IoT devices, passing it through a Redis cache before storing it in a PostgreSQL database for real-time processing and improved performance.
- Github:** <https://github.com/SchulichRacingElectrical/sr-velocity>

Pokedex | NextJS, React, TailwindCSS 2022

- Designed and built a highly interactive Pokedex application using Next.js framework and its advanced features such as server-side rendering, dynamic routing and code-splitting to enhance the overall user experience and performance.
- Implemented a robust API integration using axios and implemented the PokeAPI to retrieve and display detailed information such as Pokemon's abilities, types, stats and moves for all Pokemon species in the Pokedex.
- Utilized Next.js's built-in functionality such as the getInitialProps and getServerSideProps to efficiently manage the data and real-time updates to the Pokedex, allowing users to have the most up-to-date information.
- Github:** <https://github.com/rhamzax/PokedexReact>

Museum Gallery | Python, MySQL 2022

- Developed an intuitive art gallery application using Python and MySQL, featuring a user-friendly interface for browsing, searching, and managing over 50 art pieces, with added functionality for artists and administrators to easily add, update, and delete art pieces according to various criteria such as artist, medium, and style.
- Designed and built a MySQL database for a museum gallery, including creating tables for artworks, artists, exhibitions, and acquisitions, also implementing constraints to ensure data integrity and accuracy within the database.
- Github:** <https://github.com/Maan-Khedr-ENSF-300/museum-project-no-name-team>

MSA Membership Tracker | Python, Gspread, Imaplib, Crontab 2022

- Developed an object-oriented Python application that utilizes the Gmail API and Gspread library to scan a Gmail account for specific emails related to e-transfers and match them with reference to e-transactions. The application automates the process of reconciling e-transfers with corresponding emails, thus reducing manual errors and saving time.
- Implemented automated email parsing and data extraction techniques to streamline the process of matching e-transfers with corresponding emails. This included using regular expressions to extract relevant information such as e-transfer amounts, dates, and reference numbers from emails.
- Utilized Google OAuth 2.0 for secure user authentication and authorization, ensuring the safety and privacy of user data.
- Github:** <https://github.com/rhamzax/MSATracker>

EDUCATION

University of Calgary - Bachelor of Science, Software Engineering

September 2021 - Present