Massimo Caruso

(514) 944-5977 | massimo02caruso@gmail.com | linkedin.com/in/massimocaruso | github.com/Extinctable

Education

Montreal, Canada **Concordia University** **Jan 2023 – Present**

- **Major:** Software Engineering, BEng
- **Relevant Courses:** Data Structures and Algorithms, Operating Systems, Databases, Embedded Systems, Machine Learning, and Deep Learning

Experience

PayFacto - Payment Technology Solutions **May 2025 – August 2025**
Software Project Manager, Intern

- Spearheaded data acquisition, cleaning, and preprocessing of raw merchant data sets for the deployment of MEVWeb, a province-wide cloud-based SaaS platform mandated by Revenu Québec.
- Participated in end-to-end deployment of MEVWeb software both remotely and on-site, including planning, testing, and installation across restaurant partners.
- Led and supported physical hardware transitions by uninstalling legacy MEV devices and installing MEVWeb-compatible components such as printers and routers.
- Collaborated with project delivery, field services, and sales teams to ensure seamless coordination of deployments and merchant onboarding.
- Tested MEVWeb software builds and deployment packages, reporting bugs and verifying stability before production rollout.

AtHackCTF **Nov 2024 – March 2025**
Challenge Designer and Developer, Permanent Part-time

- Designed a complex RFID-based Capture the Flag (CTF) challenge using a real ATM machine and MIFARE Classic cards, which communicated with the machine's reader to simulate a security environment.
- Developed three flags requiring participants to:
  - Extract the card's PIN from memory by reverse-engineering the RFID data.
  - Manipulate the card's balance data, allowing the participant to alter the funds stored on the card.
  - Modify the card's UID to impersonate an admin and escalate privileges within the system.
- Implemented an interactive ATM interface, including buttons for navigation and a printer to issue flags upon successful completion of challenges.
- Facilitated the learning of hardware security, from memory manipulation to privilege escalation, within a real-world context.
- Prepared over 600 MIFARE Classic cards by writing custom data to each card and ensuring proper labeling and formatting for participant use.

Projects

Food Database System
- Developed an online nutritional database integrating APIs (FatSecret, TheMealDB) to provide detailed nutritional data, recipes, and dietary metadata.
- Cleaned and validated JSON data, storing in hybrid databases (PostgreSQL + MongoDB) with optimized queries, indexing, and aggregates.
- Automated data migration between SQL and NoSQL using Python, improving retrieval efficiency with custom recipe-name generator for FatSecret API.

Teacher-Student Feedback Web Application
- Designed responsive frontend components (sidebar, header, landing page) with smooth navigation across devices.
- Connected frontend forms to backend APIs for feedback submission and confirmation; optimized Azure-hosted SQL database queries.
- Performed acceptance testing, bug resolution, and version control (Git), maintaining coding standards and repository organization.

Linear Regression Model
- Built a multiple regression model to analyze life expectancy factors (WHO dataset, 193 countries, 2000–2015).
- Applied backward elimination to reduce predictors from 20 to 6, achieving adjusted $R^2 = 0.77$ and predicting Canada's 2013 life expectancy within 0.5 years of reported value.
- Used Python (Pandas, NumPy, Scikit-learn) for preprocessing, modeling, and visualization.

GetNextLine Implementation (C)
- Implemented a custom function to read lines from standard input using low-level system calls (‘read‘) and manual memory management.
- Designed buffer-zeroing and dynamic allocation strategies to handle varying input sizes safely while preventing overflow.
- Handled edge cases (EOF, null bytes, `\n`/`\r` line breaks, empty input) with correct memory cleanup using `malloc`/`free`.

Limited printf Implementation (C)
- Built a custom ‘printf‘ supporting %s, %c, and %d format specifiers without the C standard library.
- Used only system calls (`read`, `write`) and implemented supporting functions (`my_strlen`, `my_puts`, `my_itoa`) for string handling and integer conversion.
- Managed variadic arguments via stack frame manipulation (`x86args.h`), demonstrating low-level assembly interface and systems programming.

Skills

Languages: (Proficient): C, Java, LaTeX, HTML/CSS, Javascript, Python, SQL; (Familiar): C++, Clojure, Erlang
Frameworks: React, Node.js, Express.js, Flask
Libraries: Pandas, NumPy, Matplotlib, Scikit-learn, PyTorch
Developer Tools: Git, Docker, Makefile, MongoDB, PostgreSQL, Neo4j, VS Code, Eclipse, Jupyter Notebook
Methodologies: Agile development, Scrum, Waterfall