Youssef Khafagy

Burlington, Ontario | 416-854-4222 | khafagyy@mcmaster.ca | LinkedIn | GitHub | Portfolio

Profile

McMaster Software Engineering co-op student available for Summer 2026, with shipped experience spanning custom-trained deep learning models, LLM-integrated tools, full-stack web applications, and systems-level software. Has hands-on experience training and deploying production ML models with PyTorch, building generative AI pipelines using OpenAI, Groq, and Gemini APIs, and shipping end-to-end applications from model training to cloud deployment. Delivers consistently in collaborative, multi-contributor Agile teams.

Education

McMaster University — Hamilton, Ontario

Software Engineering, Bachelor of Engineering — 2024 – 2028

- McMaster Award of Excellence ($3,000), awarded for an outstanding admission average of 97%

Technical Skills

Software Languages: Python, Java, C/C++, Bash, MATLAB, Verilog, JavaScript, TypeScript, HTML, CSS
Frameworks & Tools: Git/GitHub, Next.js, React, Tailwind CSS, PostgreSQL, Docker, VS Code, IntelliJ, Maven
Engineering & Design: AutoCAD, Quartus Prime, MATLAB, 3D Printing
AI/ML & Generative AI: PyTorch, MobileNetV2, FastAPI, HuggingFace Hub, OpenAI API, Groq API, Google Gemini API, Anthropic API, Ollama, Prompt Engineering, AI Automation, Generative AI Tools
Concepts: Object-Oriented Programming, SOLID Principles, UML, SDLC, Digital Logic Design, Agile/Scrum

Extracurricular Activities

McMaster Advanced Space Systems CAN-SBX — Hamilton, Ontario

Web Dev Team Member — *2025 – Present*

- Enhanced and maintained reusable React and Next.js frontend components for the live MASS CAN-SBX website, improving responsiveness and visual consistency across a shared multi-contributor production codebase
- Co-authored the software requirements specification for the website, translating team discussions into formal scope definitions, browser compatibility standards, and maintainability requirements that guided development planning
- Participated in Agile Scrum sprints through task breakdown, weekly stand-ups, and sprint reviews, applying cross-team communication and project coordination in an active multi-disciplinary engineering environment

Projects

ScreenSense: AI Visual Attention Predictor | *Python, PyTorch, FastAPI, TypeScript, HuggingFace, Docker, Vercel*

- Trained a custom 6.6M-parameter MobileNetV2 + U-Net saliency CNN from scratch on the SALICON dataset (10,000 images with crowd-sourced human fixation maps), using a composite KL-Divergence + CC + BCE loss with mixed-precision training (AMP) and a frozen-encoder warmup strategy to fit within 4 GB VRAM, achieving AUC-Judd 0.9613 and CC 0.8756 - surpassing the published SALICON baseline on all 5 standard saliency metrics
- Deployed a FastAPI inference backend containerised with Docker on HuggingFace Spaces, with model weights hosted on HuggingFace Hub and auto-downloaded at startup, decoupling model versioning from code deployment while keeping inference under 2 seconds on CPU - making the tool actionable for advertisers, UX designers, and product teams validating layouts before launch
- Built a post-processing pipeline in Python that converts raw saliency maps into ranked hotspot coordinates via peak suppression, entropy-based spread scores, and a 3×3 region grid, then surfaced results through a Next.js three-mode heatmap viewer and rule-based design recommendations requiring zero ML knowledge from the end user

TriageFlow: AI-Assisted Hospital ER Intake | *Next.js, TypeScript, PostgreSQL, Docker, Groq/Gemini API*

- Engineered a full-stack AI triage platform targeting ER check-in delays, using prompt-engineered Groq and Gemini LLM calls to auto-generate structured JSON symptom summaries, with strict server-side schema validation ensuring every AI output was clinically reviewable before reaching staff
- Architected Next.js API routes backed by a Dockerized PostgreSQL database, applying Server-Sent Events (SSE) to push real-time queue updates to staff dashboards across concurrent patient sessions without polling
- Enforced a mandatory human-in-the-loop design throughout - AI output is scoped strictly to structured intake summarisation with no diagnostic authority, while staff retain full override control over every urgency level, ensuring clinical accountability is never delegated to the model

WireFish: Network Diagnostics Toolkit | *C, Linux, Raw Sockets, Bash, GitHub Actions, gcov*

- Developed a unified CLI diagnostics tool in C, implementing TCP/ICMP host scanning with per-host RTT statistics, ICMP traceroute with manually assembled packet headers and 16-bit checksums, and live bandwidth monitoring via `/proc/net/dev`
- Constructed the output formatting module to support table, CSV, and JSON export modes, and designed a `DEBUG/INFO/WARN/ERROR` logging subsystem routing exclusively to stderr, enforcing separation of concerns across a 4-person modular codebase
- Integrated a GitHub Actions CI pipeline and validated correctness using Bash unit tests and `gcov` coverage analysis, applying test-driven development and collaborative Git branching to sustain code quality across all contributors

Catan Simulator: Model-Driven Game Engine | *Java, JUnit, UML, Papyrus, SonarCloud, GitHub Actions*

- Designed a 15+ class UML domain model for Settlers of Catan directly from a natural language rulebook (SFWRENG 2AA4), using Papyrus to auto-generate the Java class skeleton with attributes, operations, and inheritance mapped to fields, methods, and `extends` relationships
- Implemented the game simulator applying all five SOLID principles: `RuleValidator` decoupled from game flow, `Settlement`/`City` substitutable via `Building` (LSP), new actions extensible without modification (OCP), iterating between UML model and code to resolve rule invariants across a 4-person team
- Extended the codebase with human player support, JSON board visualization, a regex-based command parser, and a finite automaton turn model (Roll → Build → Go), validated with 15+ JUnit tests, SonarCloud static analysis, and CI across a 100+ commit repository

Future Sight: Navigation Cane for Visually Impaired Users | *Arduino, C/C++, AutoCAD*

- Fabricated an assistive navigation cane using ultrasonic sensors and tactile pin actuators to deliver real-time obstacle-proximity feedback for visually impaired users, applying human-centred design thinking to a real accessibility problem
- Programmed embedded control logic in C/C++ on an Arduino to translate live sensor distance readings into proportional actuator responses, debugging hardware-software integration under tight memory and processing constraints
- Produced AutoCAD layout schematics to coordinate physical sensor, actuator, and wiring placement across the team, practising cross-disciplinary communication and technical documentation throughout the build

Experience

Industria Pizzeria — Burlington, Ontario
Server — *Jun 2024 – Sep 2024*

- Delivered attentive table service across multiple sections simultaneously, maintaining order accuracy under time pressure and demonstrating composure and reliability in a high-volume environment
- Coordinated with kitchen staff to align order timing and resolve service issues during peak hours, applying clear verbal communication and adaptability to keep front-of-house operations running smoothly

Tim Hortons — Burlington, Ontario
Crew Member — *Sep 2021 – Nov 2023*

- Managed 100+ daily customer transactions with consistent accuracy during high-volume rush periods, demonstrating reliability, composure under pressure, and strong prioritization
- Collaborated with teammates across drive-thru and front-counter stations to sustain service pace, applying teamwork and time management during peak-hour rushes

Certifications

WHMIS Certified, McMaster Faculty of Engineering
B1 DELF (Diplôme d'Études en Langue Française), French language proficiency certification issued by the French Ministry of Education