Ehan Idrak Hassan

Canadian Citizen, U.S. Permanent Resident

📞 416-268-2420 ✉ ehan20204@gmail.com in linkedin.com/in/ehanhassan ⚙ github.com/ehan5000

Education

Carleton University   **September 2022 – April 2026**

Bachelor of Computer Science (Honours), 4th-Year Standing   *Ottawa, ON*

- **Selected Coursework:** Data Structures & Algorithms, Discrete Mathematics, Systems Programming, Operating Systems, Object-Oriented Software Engineering, Database Systems, Web Applications, Programming Paradigms

Technical Skills

- **Languages:** C/C++, C#, Go, Java, Python, JavaScript, HTML/CSS, SQL, Lua, PHP
- **Frameworks:** React, Node.js, Express.js, Flask, Tailwind, OpenGL, Qt, JavaFX, TensorFlow, ASP.NET, Spring Boot
- **DevOps/Cloud:** Git, Docker, AWS S3, AWS Lambda, Swagger/OpenAPI, Linux
- **Databases:** PostgreSQL, MongoDB, SQLite
- **Tools:** VS Code, IntelliJ IDEA, Eclipse, PyCharm, NetBeans, Unity, Tableau, Figma

Experience

Government of Ontario, Ministry of Finance   **May 2025 – Present**

Full Stack Developer Intern   *Toronto, ON*

- Developed and deployed public-facing government web pages with **HTML/CSS**, **JavaScript**, **PHP**, and **ASP.NET**, and improving accessibility for over **500,000+ users** and ensuring full **AODA/WCAG 2.1** compliance.

- Led development of an in-browser screen reader simulator for accessibility audits; implemented real-time DOM parsing and narration with **TypeScript**, **TailwindCSS**, and **Docker**, enabling QA team to test 50+ UI components without external tools.

- Engineered accessibility workflows and backend logic in **ASP.NET**, supporting keyboard navigation, focus management, and speech synthesis; improved usability scores by **15%** based on internal accessibility benchmarks.

Projects

⚙ **AidVault: Secure Aid Distribution API**   **Go | Docker | AWS S3 | PostgreSQL | Swagger/OpenAPI**

- Developed a scalable backend system in **Go** to help nonprofits securely register, track, and fulfill humanitarian aid requests across regions, improving delivery transparency and reducing manual coordination time by over **40%**.

- Integrated secure file uploads to **AWS S3** using pre-signed URLs, enabling scalable, fault-tolerant storage of sensitive verification documents; ensured persistent access control, safe multi-user handling, and uninterrupted access across nonprofit partners.

- Containerized backend services with **Docker** and automated deployments via CI/CD pipelines, accelerating onboarding with full **Swagger/OpenAPI** documentation and significantly reducing release errors by **90%**.

⚙ **LocutusAI: Real-Time AI Speech Analyzer**   **Node.js | Express.js | React | Deepgram API | Vite | JavaScript**

- Engineered a real-time speech analysis pipeline using **Node.js**, **Express.js**, and the **Deepgram API** to detect filler words, stuttering, tone, and pacing, helping users strengthen fluency and confidence in both live and recorded presentations.

- Supported live and uploaded audio input with **Multer**, **Axios**, and **CORS**, enabling users with speech impediments and other speaking challenges to receive personalized feedback through a seamless **React/Vite** interface.

⚙ **AI Pathfinding Framework**   **C++ | OpenGL | CMake | Linux**

- Implemented and benchmarked **A*** and **Dijkstra** algorithms in **C++** across procedurally generated mazes, improving obstacle handling logic and traversal efficiency by over **50%** in varied and complex simulation environments under constraints.

- Rendered real-time visualizations of pathfinding behavior using **OpenGL**, animating step-by-step traversal with dynamic highlights and user interactions, which improved comprehension of search strategies by over **80%** in usability tests.

⚙ **Live Private and Group Messaging Chat Server**   **Node.js | Socket.IO | JavaScript | HTML/CSS**

- Designed and deployed a real-time chat system with **Node.js** and **Socket.IO**, enabling private, public, and group messaging between multiple browser clients; explored core networking concepts like sockets, event handling, and client-server data flow.

- Implemented input validation, custom message routing, user-specific formatting, and real-time message rendering to simulate production-level chat features; enhanced debugging and usability through multi-client testing and dynamic UI logic.