Victor Huang

victor.huang1@uwaterloo.ca | github.com/vichua2006 | in/victor-qibin-huang | victorhuang.vercel.app

Education

University of Waterloo Expected Graduation 2029
Bachelor of Computer Science, Honours Co-op, ***3.8/4.0*** *GPA* *Waterloo, ON*

Awards

Winner of the 2021, 2022, and 2024 NB High School Programming Competition, winning over **$1500** in prizes.
Certificate of Distinction in the 2024 Euclid Math Competition, placing top **5%** out of **23,000** participants.
Certificate of Distinction in the 2024 Canadian Computing Competition, ranking top **20%** among **4000** students.

Skills

Languages: Python, C++, C, TypeScript, Java, Shell, SQL, Scheme, HTML, CSS
Frameworks/Libraries: GraphQL, Prisma, React, Flask, Selenium, Matplotlib, AutoGen
Tools: Git, Vim, Linux, SSH, K8s, Postgres, Docker, Raspberry Pi, Arduino, GCP

Experience

Backend Software Engineer | *Typescript, Prisma, GraphQL, SQL, K8s* Mar 2024 - Present
Hack the North *Waterloo, ON*

- **Owned** design and implementation of database schema and GraphQL endpoints for an event gamification system, supporting tools and features for **1200+** users with real-time quest tracking, point transactions, and user analytics.
- Automated internal ideation workflow with Slack and Notion API, reducing manual documentation time by **30%**.
- Created **data reports** on event metrics with **Metabase + PostgreSQL**, aiding logistics in making data-driven decisions.

Software Engineer | *Python, OpenCV, SSH, Raspberry Pi* Jan 2024 - Present
Waterloo Aerial Robotics Group, Autonomy *Waterloo, ON*

- Captured **2k+ brightspot images** to fine-tune drone's computer vision system, **reducing false detections by 99%**.
- Investigated new Raspberry Pi and OpenCV camera modules, authoring comprehensive setup and troubleshooting documentation, significantly reducing debugging time for future hardware and software integration.

Research Programmer | *Python, OpenAI, AutoGen, Git* Dec 2023 - Jun 2024
UCSB Human-AI Integration Lab *Santa Barbara, CA*

- Built a framework for LLM agents to simulate human interactions through persona-based interaction.
- Developed a multi-agent conversation system using **MS AutoGen**, allowing **8+ agents** per discussion session.
- Utilized **prompt engineering** techniques (such as N-shot learning, Chain of Thought, Prompt Chaining, etc.) and **RAG** memory pipeline to enhance agent performance by 25% and virtually **eliminated all hallucinations**.

Research Intern | *Python, Matplotlib, Numpy, Pandas* Jul 2024 - Aug 2024
University of New Brunswick, Physics Department *Fredericton, NB*

- Analyzed **10,000+ laser intensity** data points across varied exposure levels, enabling precise profile characterization.
- Applied **Gaussian fitting** and built 2D/3D visualizations in **Matplotlib**, gaining deeper insight into beam distribution.
- Collaborated on the design, testing, and programming solutions of new lab exercises for an upper-year laser course.

Projects

Apocalift ↗ | *JavaScript, Flask, Python, Arduino, C++, HTML/CSS*

- Built web-based RC vehicle rental platform using **Flask**, featuring **real-time video** streaming capabilities.
- Developed RESTful API endpoints to transmit user controls to the RC vehicle via **ESP32** wireless communication.

RxSync ↗ | *TypeScript, React, Bootstrap CSS, Bland AI, OpenAI*

- Built AI-driven pharmacy coordination tool with **React**, streamlining drug requests through automated voice calling.
- Implemented **custom hooks** to sequentially call 3 APIs, retrieve call transcripts, and summarize using **OpenAI API**.

Theater Movers ↗ | *C++, Arduino, Git, KiCad*

- Prototyped a 3D-printed intelligent lighting fixture with **dual-axis rotation** for community stage productions.
- Developed custom algorithm to support **synchronized stepper motor acceleration** on both axes.

UNB Weather Station | *Raspberry Pi, Python, Shell, Linux, SSH, I2C*

- Created **3** custom weather stations using **Raspberry Pi** and environmental sensor (BME280), synchronized using a high-precision RTC module (Real Time Clock) to measure the local adiabatic lapse rate.
- Remotely connected to RPi via SSH and configured **Linux** operating system to run custom **Python** and **Shell** scripts.