Email: aumkarmali539@gmail.com GitHub: https://github.com/AumkarMali

Aumkar Mali

Education

University of Waterloo, Undergraduate in Mechatronics Engineering **Sept 2024 - June 2029**

Experience

Freelance Software Developer, Fiverr **June 2021 - Oct 2022**

- Marketed development skills through **Fiverr** and utilized **YouTube** to advertise expertise in programming and robotics.
- Implemented clients' program ideas within **tight timeframes of 3-7 days**. Project demands included **Python GUIs (Pygame, Kivy, Tkinter)**, **NLP chatbots** (trained in specific contexts such as guiding customers), and **website development** using **JS** for business purposes (e.g., developing the back end for a window blinds company page).

Owner of Tutoring Business, Part-time tutor **April 2022 - Present**

- Founded **STEM Scholar Tutors** as a personalized and affordable tutoring service for struggling students post-COVID.
- Built company website using **JavaScript, HTML, and CSS** to effectively connect students with tutors.
- Trained 6 high school students over the course of 1 - 3 years, significantly enhancing their academic capabilities.

Volunteer Summer Camp & Math Tutor, Mississauga's Ram Mandir **Sept 2021 - Sept 2022**

- Dedicated over **110 hours** to organizing religious festivals, and lead other volunteers in maintaining the temple.
- Tutored over **30 middle school students** during a **2-month summer** camp in mathematics, and programming basics.

Projects

Automated Chess Robot (*Project Showcase*)

- Built a robotic arm using **MATLAB & Simulink** and constructed it with an aluminum frame and servo motors.
- Made schematic of hardware on Circuito.io and later connected electronic components using a breadboard.
- Developed a chess AI in **Python** using the **minimax algorithm** with **alpha-beta pruning**,
- Optimized algorithm with non-linear board evaluation function which altered the program's playing behavior, allowing it to make **3,000 and 100,000 calculations**, consistently predicting the optimal scenario for victory.
- Enhanced accessibility using the **Tkinter library** to create a **GUI** that displayed the board setup through the game.
- Integrated the GUI with backend code by employing **subprocesses** that ran both programs simulaneously.

Android Homework Help App (*Source Code*)

- Developed an Android app using **Python** and the **Buildozer library** that enables students to input mathematical problems and receive step-by-step solutions, complete with explanations and accompanying diagrams or graphs.
- Applied the **WolframAlpha API** to generate the detailed and accurate algebraic solutions based on user prompts.
- Designed a GUI using the **Kivy library**, which utilized a hosted Python program in the cloud to receive output data.
- Utilized: Python, Buildozer library for Android-Python development, pythonanywhere hosting platform, Kivy.

Bluetooth RC Car with Metrics (*Project Showcase*)

- Designed a **microcontroller-based** RC car using **bluetooth transmitter**, **IR sensor**, and an **Arduino** circuit board.
- implemented a **circuit schematic** to establish connections between the microcontroller and electronic components, such as a relay for motor control, infrared (IR) sensor for measuring speed and distance, and a servo motor for steering.
- Integrated hardware using **Arduino IDE and C++ code**, containing basic arithmetic operations required to convert onboard data into measurable output that could be displayed on the programmed mobile app (**MIT App Inventor**).
- Utilized: C++ code, Arduino Uno, HC-05 Bluetooth Transmitter, Circuit Schematics, Android Development.

Interactive Discord Bot with Natural Language Processing (*Source Code*)

- Created a Discord bot using the **Discord Developer Portal** and **Python** to manage servers using commands (e.g., sending weather data from **The Weather Channel's API**) and use **machine learning** to communicate with members.
- Integrated a prebuilt **SQLite file database** with preset queries and updated its existing data with new speech from interactions with server members, allowing the bot to better communicate with more human traits such as humor.
- Deployed the bot through the cloud using a **Flask server**, enabling other servers to use the bot in real-time indefinitely.
- Utilized: Python, Flask, Discord API, SQLite, ChatterBot corpus, Repl.it static hosting.