Aymen Shoteri

(289) 689-1515 | aymenshoteri@gmail.com | linkedin.com/in/aymen-shoteri | github.com/AymenS02

EDUCATION

McMaster University - Mathematics and Computer Science **Hamilton, ON**
Honours Bachelor of Science *Expected Graduation May 2025*

- **Academics:** Cumulative GPA of 3.5/4.0
- **Relevant Coursework:** Advanced Calculus I & II, Theory of Algebra, Introduction to Software Development, Logical Reasoning for Computer Science, Data Structures and Algorithms, Computer Graphics (OpenGL), Introduction to Programming (Python). "Data Structures and Algorithms (Java),

TECHNICAL SKILLS

Programming Languages: Proficient in Python, Java, JavaScript, SQL, C, HTML, CSS, Haskell, OpenGL, and Python Developer Tools: Tableau, SQLite3, Pytorch, Eclipse, and GitHub

PERSONAL PROJECTS

Crypto Tracker Application Using React (with API Integration)
Project in JavaScript *August 2023*

- Developed the Crypto Tracker web application using **React**, demonstrating strong proficiency in modern web development technologies.
- Implemented real-time data retrieval and display, providing up-to-date cryptocurrency pricing and historical information to users.
- Designed and structured the user interface using **JSX**, **HTML**, and **SCSS** to ensure an intuitive and visually appealing experience.
- Integrated Font Awesome for enhanced iconography, **Axios** for **API** requests, and **Zustand** for efficient state management.
- Implemented **React Router** for smooth navigation and **classNames** for dynamic **CSS** class application.
- Facilitated cryptocurrency searches, allowing users to quickly find and view the current prices of their preferred digital assets.
- Utilized **Sass** to enhance the styling capabilities and maintain code maintainability.

Connect 4 Game Made Using Java (with AI)
Project to Java and AI interface *April 2023*

- Developed a Connect 4 game using Java programming language that allows two players to play against each other or a single player to play against an AI opponent.
- Implemented a **GUI** using **JavaFX** for user interaction, allowing players to easily place their pieces on the board and view the current state of the game.
- Designed an algorithm to check for win conditions after each move, ensuring that the game can detect when a player has won or when the game has ended in a draw.
- Incorporated an AI player using a **minimax** algorithm, enabling the computer to make intelligent moves based on the current state of the game. The AI player uses a recursive function to explore all possible future moves and chooses the move that maximizes its chance of winning while minimizing the chance of losing.