Varun Parikh

✉ varun.parikh@uwaterloo.ca
www.vrnprkh.dev
✉ parikh.var@gmail.com
www.github.com/vrnprkh
☎ +1-587-429-4785

Technical Skills

Programming Languages: Python, C/C++, VHDL, HTML, CSS, JavaScript, TypeScript, SQL, Java, Scala, ARM
Tools: Git, React.js, Node.js, Express.js,LaTeX, Pandas, UNIX, Flask, GCP, DigitalOcean, Arduino, Azure

Experience

AI Solution Developer - Microsoft Azure Waterloo Accelerate Program Apr. 2023 - Aug. 2023

- Led design to develop a solution to drastically minimize the impact of financial fraud against older adults
- Developed advanced scam call detection using Azure AI speech-to-text and a custom machine learning model
- Received AZ-900 and AI-900 Certifications

Private Tutor Sep. 2020 - Jun. 2022

- Tutored 10 university level students for math and CS courses including Calculus 1 and 2,
- Created detailed lesson plans to help students succeed; grades increased up to 40% from midterms to finals

Projects

DocUrCode: Web tool made in react that automatically writes descriptions and documentation for your code

- Used OpenAI API with GPT-3.5 to generate line by line code descriptions with customizable levels of detail
- Developed front-end using React; implemented syntax highlighting, window resizing, and dynamic code selection
- Created back-end using Express.js and hosted server on Google Cloud with secure back-end to safely hide API key
- **Technical Skills:** React, Node.js, Express.js, Google Cloud, HTML, CSS, TypeScript, JavaScript

Interactive Chess Board: Interactive chess board made as a teaching tool for beginners

- Used multiplexers to process data from hall sensors multiple times per second to accurately track piece positions
- Identifies pieces when picked up, and highlights all legal moves, and indicates when an illegal move was made
- Designed an algorithm to process piece positions and moves, while tracking held pieces, only using binary sensor data
- **Technical Skills:** Python, pygame, pyfirmata, Arduino

VrnHDL: Easy-to-use markup language for generating complicated digital circuit diagrams

- Implemented syntax parsing, detailed error messages, and overlap minimizing wire rendering algorithm
- Allows for advanced looping and conditionals to create complicated diagrams with very little code
- Made in Python, using the PIL library for image rendering; hosted on website built using Flask
- Created detailed unit and integration tests to ensure bug free syntax parsing and image rendering
- **Technical Skills:** Python, PIL, Flask, HTML, CSS, JavaScript, jinja, DigitalOcean

FactorySplitter: Web tool that generates a graph to evenly splits conveyor belts used in factory games

- Implemented a graph generating algorithm that creates an optimal method to balance loads
- Algorithm splits 1 node into n-outputs, by splitting nodes evenly up to a given amount, or merging 2 nodes
- Output rendered in Graphviz to generate a detailed graph with labeled nodes and distinctions between node types
- **Technical Skills:** JavaScript, HTML, CSS, Graphviz

8-Bit CPU: An 8-Bit CPU using the Harvard Architecture created in Minecraft

- Designed a custom 16-bit instruction set allowing for arithmetic, conditional branching and memory management
- Developed a basic assembly language along with an assembler in C++ to efficiently write complex programs
- Test programs included calculating the Fibonacci sequence, and finding the square root of a number

OnitamaAI: AI that plays the board game Onitama; which beat experienced human players more than 96% of the time

- Made in Python, implemented using a minimax algorithm with alpha-beta pruning
- Implemented a custom metric to evaluate a given board position efficiently and accurately

Education

University of Waterloo Waterloo, ON
Bachelor of Software Engineering (Honours) Candidate, with Co-op (84.29% GPA) Sep. 2022 – Present