

4 Darou Crescent, Brampton, Ontario
437-226-9367
jonkhan738@gmail.com

Jonathan Javed Khan

Objective:

Motivated university student looking for an entry-level job, eager to learn and work diligently

SKILLS

Proficient in Python programming
Basic skills in HTML
Basic skills in CSS
Basic skills in JavaScript
Excellent written and verbal communication skills
Strong foundation in computer science principles
Advanced mathematical skills
Keen attention to detail
Excellent critical thinking abilities
Friendly and approachable demeanor
Effective task prioritization and time management
Collaborative team player
Skilled in building positive relationships with colleagues

EDUCATION

University of Toronto, Mississauga — *Computer Science, Mathematics, and statistics degree*

Started: September 8th, 2022

Philopateer Christian College, Mississauga — *High school diploma*

Received: June 16, 2022

EXPERIENCE

City of Mississauga's One Million Trees Program

April 27th, 2022

- Helped the City of Mississauga plant one million trees
- Helped in clean up
- Helped in supervision

University of Toronto Global Engineering Challenge

July 2021-August 2021

- Worked in a team to set code that predicts wildfires
- Performed data analysis to find methods to prevent future wildfires

Projects

Portfolio Website

This website portfolio uses HTML CSS and JavaScript to serve as a dynamic showcase my skills, experiences, and projects. It features a responsive design with a user-friendly navigation menu, making it accessible on various devices. The website provides valuable information about my background, including my education and proficiency in programming languages. Additionally, it highlights my projects, offering visitors the opportunity to explore my work through GitHub and live demos. With a focus on professionalism and aesthetics, this website effectively communicates my qualifications and ambitions

Spell Checker:

The code demonstrates a simple spelling checker application using Python's tkinter library for the graphical user interface (GUI) and NLTK for natural language processing. The program creates a tkinter window with a text input area where users can type or paste text. As the user types or modifies the text, the program checks for misspelled words by comparing them against a list of English words from the NLTK corpus. Misspelled words are highlighted in red within the text widget.

Key Features:

- The program initializes a tkinter window and a ScrolledText widget for text input.
- It uses NLTK to download a list of English words.
- The program defines a check method that is called whenever a key is released within the text widget.
- Within the check method, it counts the number of spaces in the text and compares it to the previous count to detect word input.
- It cleans and tokenizes the input text, highlighting misspelled words in red by applying tkinter tags to the text widget.

Text Summarizer

The code demonstrates a Python program that performs extractive text summarization using the NLTK library. It tokenizes input text, removes stopwords and punctuation, calculates word frequency, assigns scores to sentences based on word frequency, selects the top-ranked sentences, and combines them to create a summary. The program aims to extract key information from a given text, condensing it into a shorter, more concise form.

Key word finder

The code demonstrates a Python program that extracts keywords from a given text using the NLTK library. It tokenizes the input text, removes stopwords and punctuation, calculates word frequencies, and assigns TF-IDF (Term Frequency-Inverse Document Frequency) scores to words. The program then ranks the words based on their TF-IDF scores and selects the top N keywords as the output.

Key Features:

- The program tokenizes the input text into words and converts them to lowercase.
- It removes stopwords (common words like "and," "the," "in") and non-alphanumeric characters.
- Word frequencies are calculated using the Counter class from the collections module.
- A simplified TF-IDF score is calculated for each word (assuming a fixed IDF for all words).
- Keywords are ranked by their TF-IDF scores and the top N keywords are selected as the output.