The Following is all the work I have done involving Computer Science and Software Engineering from most recent followed by all work in chronological order

Michael **Ramoutar**

(718) 593-1633 miker704@gmail.com **PORTFOLIO** **GITHUB** **LINKEDIN** **ANGEL**

SKILLS

React, Redux, Ruby, Rails, C++, JavaScript, Java, SQL, C, C#, Python, Rust, PostgreSQL, SQLite3, jQuery, Assembly x86, Mongoose, MongoDB, Node.js, Express.js, Sequelize.js, HTML5, CSS3, Bootstrap 5, Canvas, Git, Heroku, AWS, Webpack, Babel, OpenGl , Unreal Engine, Unity, Arduino

RECENT PROJECTS

Paint by Numbers (Mongoose, MongoDB, Express, React. Redux, Node.js) live | github

An application built using the MERN stack for creating, playing, and sharing Nonogram puzzles using drawn or uploaded images.

- Working as a team of 3 engineers, serving as the lead backend engineer, handling overall project structure, MongoDB management, user authentication, models, schema, validations, Axios requests, Redux Store, actions and reducers.
- Designed a complex framework of React components to compose the puzzle Board interface, which facilitates a myriad of stylish and responsive functionalities such as multi-select.
- Scripted robust functions for importing and converting images into pixilated Tile maps, whose RGB values were analyzed to produce playable, sharable puzzles and be parsed to a string to be saved to the database.

Centipede.js (JavaScript, DOM, HTML, CSS, Canvas) live | github

A recreation of the classic Atari 1981 game Centipede with a modern twist including more enemies, power ups and game mechanics.

- Utilized canvas to draw and render custom made assets to the screen infinitely till the player looses all lives.
- Used advanced mathematics to create unique AI algorithms for player tracking and movement for the Centipede and other entities in the game. Other algorithms include physics and collision detection, gravity, for certain objects in the game.
- Added a wide array of features not present in the original game include new enemies, power ups, and game mechanics.

$TR!F3 (Ruby on Rails, React, Redux, PostgreSQL, ActionCable, WebRTC, C++, AWS) live | github

A comprehensive full-stack clone of the popular communication app Discord and its functionalities including text, video & voice chat

- Utilized both React and Rails to produce a dynamic, responsive single-page application connected to PostgreSQL database.
- Integrated hundreds of modular React Components to compose the webpage into a seamless, responsive UI experience.
- Employed Web sockets using Action Cable and WebRTC allows users to live-chat including video & voice calls, create servers, and message each other all in live time.
- Created an API called CORE which uses Action cable and Redux to achieve full asynchronous functionality everywhere in app, any Redux action dispatched that involves multiple users, or visual changes will be transmitted live to each user.

The CORE (C, C++, Assembly x86, Java, OpenGL) – Capstone Project

A Graphics and Game engine based on id Software's Doom Engine used to build games while using OpenGL

- Create a graphics engine with texture mapping, level generation, & higher application performance in OpenGL.

OTHER PROJECTS

GPA Calculator (C++) Fall 2016

Class Project – Intro to Programming GPA Calculator

- GPA calculator using C++ a project for introduction to programming course.

Half Byte Packing Simulator (C++) Spring 2017

Class Project – Simulate Data Compression of a String using Half Byte Packing of EBCDIC and ASCII characters

- Simulate Data Compression of random strings using half byte packing
- Simulates both uncompressing and compressing of a string using half byte packing on either EBCDIC or ASCII characters.
- Allows a user to input their own string to be compressed or decompressed and choice of EBCDIC or ASCII characters encoding.

Messenger App (Java) Summer 2017

Messenger app that allows sending and receiving of messages via an entered IP address using web sockets and Java Swing Library.

- Connect to a user by entering their IP address
- Send and receive messages live through web sockets.
- GUI Styled and built using Java Swing library.

Snake (Java) Summer 2017

Recreation of the classic game snake in Java.

- Recreated the classic game snake in Java GUI styled with Java Swing and games assets drawn Java AWT.
- Features Key bindings, frame rate handling, and full features of the original game.

Windows Calculator (Python) Summer 2017

Recreation Windows Calculator using Python and Tkinter Library.

- Recreated Windows calculator using Python with GUI style and built using the Tkinter Library.
- Does basic mathematical computations.

Snake (Python) Summer 2017

Recreation of the classic game snake in Python.

- Recreated the classic game snake in Python using the curses library.
- Incorporates the full features of the original game.

Web Crawler (Python) Summer 2017

Created Web Crawler that browses websites returning bodies of text, and pulls images from a page via user inputted URL.

- Allows User to input URL of site to crawl.
- Downloads all images on a page if it has any.
- Returns sections of text matching a user inputted string from the crawled webpage.

Key Logger (Python) Summer 2017

Create a Key Logger that records user key strokes

- Reads and records user keyboard input.
- Saves read user key input to a file.
- Made versions to work on either Linux or Windows computers

Lexical Analyzer (C++, Python) Fall 2017

Created a lexical analyzer to scan a input file and identify C++ reserved words, variables, etc. that is able to parse the entire C++ standard library in a similar fashion to a real compiler

- Reads a user inputted file then writes a generated table of results in a new file.
- Using basic file reading and logic scans C++ code to identify reserved words, coding syntax, Lexemes and tokens of the C++ programming language in similar fashion of a real lexical analyzer base on complier design theory.
- Made C++ and Python versions

Address Book (X86 Assembly) Fall 2017

Created an address book app that records individuals information in assembly x86 language

- Reads and records user inputted information and saves it to a text file logging a series of user information.

GPA Calculator (X86 Assembly) Fall 2017

GPA Calculator coded in X86 Assembly Language 32 bit

- Reads and records user input calculates the GPA for all courses in a semester.

Computation Companion App (X86 Assembly) Fall 2017

Created a multi calculation App is Assembly including money, time conversion, numerical calculations including median and

discriminate, GPA calculator

- Converts x amount of cents to bills and coins.
- Calculates max, min, medium, harmonic mean and discriminate of 3 numbers.
- Converts x amount of seconds to normal time format of hours : minutes : seconds
- Reads and records user input calculates the GPA for all courses in a semester.

Huffman Coding Data Compression (X86 Assembly) Fall 2017

Created an Implementation of the Huffman Coding data compression algorithm in X86 Assembly

- Reads a user inputted file and returns a new file containing the compressed data.
- Uses a Minimum Priority Queue to store the nodes of each character scanned from the input file a frequency counter is added to each letter, sub trees with the lowest frequency are then dequeued and merged into a single tree with frequency summed up, while the last node is re-queued this process continue till end of file is reached

Address Book (C++) Winter 2017

Created an Address Book that logs users information using OOP in C++

- Reads and records user information such as name, phone, address, etc.
- Saves data to a file that is formatted to be read again to modify data.
- Implemented using with and without a Linked List

College Registrar System (C++) Winter 2017

Created a College Registrar System using OOP in C++

- Reads students records from a file writes to a file in a pseudo CSV format for insertion of existing records.
- Asks user for information applying to them in order to view their data such as their grades, course info, etc.
- Allows users to register their information, provide major, minor, number of courses they are taking, etc.

Snake (C++) Winter 2017

Recreated the Classic game Snake

- Implemented all the features in the original game, graphics and models are rendered using ASCII characters.

ASCII Game Engine (C++) Summer – Winter 2017

Created an Fps game engine that uses ASCII Characters to as textures to render as walls and floors

- Create levels buy filling a multi-array with ASCII characters which are then texture mapped to generate the games walls.
- Uses Physics algorithms to add gravity and collision detection
- Supports user key bindings to move, jump etc.
- Fully modular to render actually images and textures for graphics and be able to integrate a game built on top of the engine.
- Does not need heavy dependencies such as OpenGL

Huffman Coding (C++) Spring 2018

Re-implemented Huffman Coding project in assembly to C++

- Reads a file full of data compresses it and writes it to a new file.
- Uses a Minimum Priority Queue to store the nodes of each character scanned from the input file a frequency counter is added to each letter, sub trees with the lowest frequency are then dequeued and merged into a single tree with frequency summed up, while the last node is re-queued this process continue till end of file is reached

Connect Four (C#) Spring 2018

Created the game Connect four using C# and Windows forms

- Create Connect four games in C# with UI styled and built using Windows Forms.
- User plays against a computer player.
- AI algorithm made for the computer player to analyze player probability of next move and make calculated decision in tandem to following the games rules.

Notepad (C#) Spring 2018

Created a Text Editor using C# and Windows Forms

- Reads and writes user keyboard input to the screen using a gap buffer.
- UI styled and function added using windows forms.

Pac-Man (C#) Spring 2018

Recreated the classic game Pac-Man using C# and windows forms

- Recreated Pac-man in C# including A.I. and Path finding Algorithms of the ghosts.
- GUI built using windows forms

Mis-Pressed? (C#) Spring 2018

Created a Key Logger that records user key strokes and activates random commands mouse movement, key presses, and sounds.

- Reads and listens to user peripheral input, mouse clicks, typing etc after a set delay will activate a random behavior.
- Random behaviors are executed after a set delay based on either the most frequent or type of event that last occurred will trigger random key presses sometimes re-typing the users prior key inputs, erratic mouse movement and windows sounds.
- Configured to be hidden by task manager

Lexical Analyzer V2 (C++) Spring 2018

Recreation of the lexical analyzer project to assist in tutoring students currently assigned to the project but using much more advanced approaches, features reduced code base, faster performance and code that aligns to the level;

- Has the same functionality and features of the original project.
- Code base is massively reduced from a thousand lines of code to around 300 lines compared to original project.
- Uses regular expressions and the vector library with vector merging operations to increase performance massively.
- Dryer code base and massive time complexity gains from original project

Advanced Key Logger (C++, Power Shell) Summer 2018

Create a Key Logger that records user key strokes that hides from windows processes and encrypts information and sends said information via email using power shell scripts.

- Reads and records user keyboard input.
- Hides from windows task bar and masks itself from windows task manager, skips antivirus detection without using the necessary dependencies.
- Custom Encryption and Decryption of file containing logged keystrokes
- Efficient uses very little resources and memory
- Contains Power Shell scripts to send email encrypted logged files

MBRW *.* Virus (C++, X86 Assembly**)** Summer 2018

Created a Virus once triggered activates a process that overwrites the MBR rendering the machine unbootable

- Uses Windows API's and Libraries to access internal processes.
- Using C++ ability to use inline assembly rewrite the 512 bytes of information held in the MBR to null, which renders the operating system unbootable after shutdown.
- Tested successfully in virtual box for Windows 8.1 and 10

Laptop Mount (Blender) Summer 2018

Aid in Development and design of a 3-D printed mount for laptops that can attach to a soldiers body. (DOD Internship)

- Using Blender to design a mount that hold a laptop that can be strapped to the body of a soldier and allow hand free use without having to take their hands of their weapons.

Autonomous Drones – Navigation & Object Detection (Matlab, C++, Python, OpenCV, CNN, ROS) Summer 2018

Aided in Development, Testing and Implementation of Navigation Stack, object detection, and machine learning algorithms on UGVS. For autonomous movement of ground drones to detect objects of interest to enhance safety of ground soldiers and provides intelligence of combat environments. (DOD Internship)

- Using Husky UGV's implement navigation stack, object detection and machine learning algorithms to enable autonomous movement and detect objects of interest on the battlefield.
- Using OpenCV and CNN neural networks implemented on ROS (Robotic operating System) to enable object detection algorithms.
- Using lidars and Intel real sense cameras to generate and 3d map of the environment to be used by the navigation stack to allow the Husky UGV to move around unmanned and detect objects of interest and get information of distance between it and objects of interests/ objects and obstacles in the environment to calculate navigation around the environment.
- Under Project Lead navigation stack was implemented and test successfully by letting UGV leave the test room and navigate by itself around the lab floor moving past obstacles and moving around other employees back to the test room unmanned .
- Integrated Intel real sense camera to stream live video from the drone's computer to a remote computer.

Multi Clock (JavaScript, DOM, CSS, HTML) Fall 2018

Create a Clock App that Displays the current time live

- Uses either local time zone or system time zone to calculate the current time.
- Option to switch between an analog or digital clock.

Windows Calculator (JavaScript, DOM, CSS, HTML) Fall 2018

Created a basic Windows Calculator in the browser

- Clean Simple UI.
- Provides basic mathematical calculations align to the standard settings in Windows Calculator

Microsoft Paint Clone (JavaScript, DOM, CSS, HTML, Canvas) Fall 2018

Created a clone of Microsoft paint in the browser

- Using HTML Canvas allow users to draw images with their mouse cursor
- Allows the user to choose their own custom RBG colors in addition to default colors
- Variety of brushes and sizes to select and draw with similar to Microsoft Paint
- Users can save their drawings and download them as an image.

Messenger App (Java) Fall 2018

Created Messenger App in Java using modern Java FX GUI library and multi-threaded processing. (Class Project)

- Uses Multi-threaded processing.
- Saves messages to an output file.
- Uses Web Sockets to send and receive messages in real time
- Connects via entered IP address and port number to communicate with another user.
- Style using the modern Java FX library

Fitbit Clock Face App (JavaScript, Fitbit OS) Fall 2018

Created a Fitbit clock face app featuring tracking of users heart rate, time, and other health stats using Fitbit OS API

- Using JavaScript , Fitbit OS API and Fitbit dev tools platform created a watch app to display all track able health data of a user with minimum user input to view said data all on a single screen.
- Users can set and remove what data they want to appear on the watch face and place them in their desired position

Arduino **Game (**Arduino, C++) Fall 2018

Created a 2-d side scroller game to run on an Arduino board that would serve as a hand-held game

- Using Arduino IDE write code to be imported onto the micro controller of an Arduino board.
- Soldered electrical components including an LCD screen, buttons etc into a hand-held package.

Shared Books (React.js, Node.js, Express.js, HTML, CSS, Bootstrap, PostgreSQL) 2018-2019

Created Full stack app that is an Amazon clone rotated around selling, exchanging, and renting of college textbooks.

- Users can browse the app for college text books they need.
- Users can offer to pay, rent, or exchange a textbook for another textbook with other students.

- If users decide to rent or trade each other's text books a SHA code would be generated and exchanged with users when a trade has been confirmed to ensure users would return their books.
- App aims to assist college students in finding ways to get textbooks that are commonly used at their school, save them money and ensure reusability of textbooks

The CORE (C, C++, Assembly x86, Java, OpenGL, CMake) – Capstone Project Winter 2017 - Present

A Graphics and Game engine based on id Software's Doom Engine used to build games while using OpenGL (Capstone Project)

- Create a graphics engine with texture mapping, level generation, & higher application performance in OpenGL.
- Engine features multiple components that build a working game engine including sub engines dedicated to handling , graphics rendering, sound, input, physics, utility, all connected to a core engine running everything together in the game loop
- Uses C++ and OpenGl to generate a 3-d world, features lighting and shaders.
- Features custom level building and texture mapping user can import image files that can be used as textures to map out the game world.
- Users can draw out the level of the game using custom utils.
- Features Physics engine to handle gravity, collision detection, etc
- Can fully be used to implement a game using the engine
- Engine is built of ASCII Engine before fully transitioning to use OpenGL to become infinitely more powerful.
- Used as Capstone Project for a Software Engineering Course near end of 2018 with a Java version built with a game using the Java version of the engine

The CORE.java – (Java, OpenGL) Capstone Project Winter 2018

A Conversion of the Original Core Engine to Java for Capstone Project

- Using Java and OpenGl completed a near impossible task under tight time constraints to convert the original Core Engine from C++ to Java within less than a month
- This version of the Engine would go to build a Wolfenstein Clone

Wolfenstein Clone – (Java, OpenGL) Capstone Project Winter 2018

A Recreation of the first 3 levels of the classic game Wolfenstein 3-D using a custom built game engine converted to java

- Using the Java Version of the Core Engine Project built the first 3 levels of Wolfenstein 3-D
- Added A.I. Texture mapping, Collision Detection for the game itself on top of the engines assets.
- Turned in as a Final part of Capstone project for a Software engineering course as a proof to the practicality of the previous engines built.

Lexical Analyzer V3 (LEXY) (C++) Winter 2022 - Present

Recreation of the original Lexical Analyzer Projects from the past but using pure advance data structures and concepts aligned to complier design theory using grammars, syntax, parse trees, etc. to code a realistic and practical lexical analyzer used by compliers.

- Reads files and writes to a new file a table of generate results containing the lexemes and tokens contained in the read file.
- Uses Complier design theory and principles of programming languages to code a proper lexical analyzer, this includes the use of advanced data structures and algorithms and computer science theory.
- Uses no simple logical approaches or pre-made libraries or utilities such as regular expressions
- Uses Parse, syntax, abstract trees, left/most and right/most recursive grammars

HACKATHONS

Beacon (C++, Java, Android) Fall 2017

Create a basic concept where survivors can use their devices to connect to cell towers to aid rescue teams find survivors of earthquakes faster.

- When a natural disaster occurs, emergency services such as amber alert can send a code that can be typed to connect to any cell tower closest to them which sets up a beacon.
- Beacons can be used to help rescue teams find survivors faster by pointing them to the right location

The Watcher (C#, JavaScript, MySQL) Spring 2018

Created an app that notifies the user of the crime rate levels at their current location and provide a live update of directions to reach home in case of emergency

- Pulls data from local police precincts and averages the crime rates.
- Sends the user a notification about the area they are currently in.
- If a crime has been recently committed or if an area has a high crime rate send a live ongoing notification from Google maps to the users on the lock screen, providing directions home or options to change the route to their destination.

Android Hub (Java, Android, Kotlin, MySQL) Fall 2018

Created an Android version of the Blackberry Hub app to store all push notifications that appear in the notification bar

- Created a Basic Android App using Android Studio.
- Using API's from popular apps such as Gmail, Facebook, Twitter etc store all incoming push notifications and organize them in the app allowing the user to view them later in case they clear it from their notification bar.
- Ability to store notifications from recent phone and SMS messages
- Stored all push notifications in MySQL Database

EXPERIENCE

Associate Software Developer

Infosys, New York, NY July 2020 - March 2021

- Associate Software Developer Trainee, trained in both front/back end development in Java, JavaScript, Angular.js, Node.js, HTML, CSS, SQL, and Spring Framework, to develop, test, and maintain systems and applications of company clients.

Freelance Computer Science Tutor

New York, NY Winter 2020-Summer 2021

- Tutor College Students in various computer science topics, including various programming languages including Python, C, C++, C#, Java, etc.

Computer Science Teacher

RFCUNY, New York, NY Spring 2019

- Teach High school Students the basics of C++ and Python in an after school program designed to introduce the field of computer science to a younger audience

Computer Science and Mathematics Tutor

RFCUNY- York College, New York, NY Spring 2018 – Summer 2019

- Tutored college students ~15-30/week in Math, programming, data structures & algorithms, and computer science theory.

Full Stack Software Developer Student

RFCUNY- CUNY Tech Prep, New York, NY Summer 2018-Summer 2019

- Tutored Selected for full stack JavaScript training program, as one of 122 students out of 400+ applicants
- Learn in-demand technologies like React, Node + Express, and PostgreSQL
- Designed a web app to exchange books between college students reducing expenses on course materials
- Future updates include: integration with cunyfirst APIs where student ID will sync with their courses to identify needed course textbooks

Open Source Contributor, Programmer, Beta tester

ModDB.com, Online Summer 2017 – Summer 2020

- Build and contribute to community projects, including mods for the game Doom (1993) by enhancing texture mapping and implementing game features that will run on GZDoom source port.

Software Engineer Intern, Autonomous Drones & A.I.

United States Department of Defense - Army Research Labs, Baltimore, MD Summer 2018

- Designed, implemented, and tested navigation stack & object/detection algorithms on a UGV, which used CNN neural networks to detect objects/people in an area to tactically aid soldiers – (MATLAB, C++, Python, ROS)

EDUCATION

Software Development - *App Academy | Spring 2022*
Software Development – *CUNY Tech Prep | 2018 - 2019*
Computer Science BS – *CUNY – York College | 2014 – 2019*