# Tyler Weiss

IT Support and Full-Stack Computer Programmer

## Contact

43 Crescent Hollow Drive
Sewell, NJ 08080
609-970-2619
tylerhweiss@gmail.com
www.tylerhweiss.com

## Education

### Temple University

College of Science and Technology
Philadelphia, PA
BS in Computer Science
GPA: 3.48
Dean's List (GPA 3.86) Spring 2021

### Clearview Regional High School

Mullica Hill, NJ
Graduated with Honors
Member of National Honor Society
Varsity athlete in soccer, wrestling, lacrosse

## Employment

**IT Tech Support**
August 2020 – May 2023

I provided on-demand, in-person technical support for professors while they live-streamed classes from on-campus classrooms. This included general application support for Microsoft Office (Word, PowerPoint etc.) and Windows support and troubleshooting of audio and video within the classroom environment. This included assisting with room/office cameras, projectors, and speakers, as well as resolving issues related to PC-based cameras and speakers as well as

## Relevant Experience

### Portfolio Website

I designed and developed a responsive website to showcase my coding ability utilizing HTML, CSS, JavaScript, and Bootstrap. For enhanced security against potential attacks, I implemented Content Security Policy (CSP) headers and reCAPTCHA for secure form submissions. For server-side functionality, I used Node.js and Express.js, and I used an SMTP server for email connectivity. I demonstrated strong full-stack development skills, API integration, error handling, and secure data management with environment variables while completing this project.

Tools used: HTML, CSS, JavaScript, Bootstrap, Node.js, Express.js

### Spotify Playlist Sorter Website

I created a Flask web application that sorts user playlists using HTML, CSS, and JavaScript. It accesses and retrieves user-specific playlist data from their Spotify account via the Spotify Web API. The Spotify Playlist Viewer project incorporates a Python script with the ability to arrange their playlists dynamically based on date released, track name, and artist name. The application can either update or create a new playlist based on user preference. To organize the playlist data in real time without requiring server-side processing, the sorting capability is done using client-side JavaScript.

Tools used: Python, HTML, CSS, JavaScript, APIs

### Mobile Application

I've worked on a team to develop a multi-page Android application called Round that creates an environment for students and social groups to easily organize each other's schedules for availability and share contact information. Firebase is used to store all the users with their credentials. The app can create a group and share a group code via a QR code. Someone else in that app can then join the group. You can sync

other peripherals like configuring multiple monitors. I also configured Zoom meetings and troubleshot A/V and screen-sharing issues. At the end of each day, I secured the equipment and locked classrooms and lecture halls. I was trusted with keys to several campus buildings.

**Pioneer Pipe**
June 2017 – Current

I helped manage the main yard along with inventory in the garage. I unloaded delivery trucks with pipe equipment and delivered them to various sites. I also kept construction equipment clean.

# Tools and Languages

- C/C++
- Python
- Java
- HTML
- CSS
- JavaScript
- Node.js
- Express.js
- APIs
- Bootstrap
- Git/GitHub
- SQL
- Flutter/Dart
- Agile development
- Unix/Linux
- VS Code
- IT Services

# Activities

**Boy Scouts of America**
Earned rank of Eagle Scout, December 2018

**Alpha Kappa Lambda Fraternity**
January 2020 – May 2023
Vice President of Risk Management
Fall and Spring 2021

your Google calendar events in the app, and the app will compare all other group members to see when each group member is free next. My team utilized agile development methodologies to manage and deliver our project to meet milestone deadlines.

Tools used: Flutter, Dart, Android, Firebase

## Network Security and Authentication Simulator

I developed multiple client and server applications that simulated various secure network communication protocols. This included writing a DES encryption algorithm from scratch, public key encryption, and implementing a basic Kerberos ticket server.

Tools used: Python, Client Socket Programming

## Unix Shell Project

I've developed a custom Unix shell that offers seamless functionality for starting programs, traversing directories, and using commands like "ls" and "cd" that I developed myself. Input and output redirection to files, pipes, search paths, and batch file execution are additional features supported by the shell. For added efficiency, I have also added the ability for parallel processing.

Tools used: C/C++, Unix, Process Management, Makefiles, Streams, Forks, File I/O

## Spellchecker Project

I've developed a spellchecking service that utilizes socket connections to accept and validate one or more words. This service supports numerous clients and handles connections in parallel, guaranteeing peak performance even under high usage. I set up extensive logging features to keep track of all activities, assuring transparency and facilitating quick troubleshooting. I also modified a Python client to add more testing options, enabling a thorough and effective testing procedure.

Tools used: C/C++, Python, Unix, Sockets, Threading, File I/O