# HomeNetCheck

by

Joshua Gyau, Aidan Knuth, & Max Sayers

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology



| | |
|---|---|
| *Joshua Gyau* | April 22, 2021 |
| Joshua Gyau | Date |
| *Aidan Knuth* | April 22, 2021 |
| Aidan Knuth | Date |
| *Max Sayers* | April 22, 2021 |
| Max Sayers | Date |
| *Ryan Moore* | April 22, 2021 |
| Ryan Moore, Faculty Advisor | Date |

University of Cincinnati

College of
Education, Criminal Justice, and Human Services

April 2021

# Table of Contents

# List of Illustrations

# Abstract

HomeNetCheck is a tool for users and support technicians to help better diagnose network issues which would otherwise be difficult to address remotely. The move to working from home spawned many technical issues for end users. In the past year an abundance of new network issues have arisen, HomeNetCheck will enable technicians to run tests and easily analyze data that could identify network issues. Information that is collected during the support will be stored for other technicians experiencing similar issues in the future.

# 1. Introduction

## 1.1 Problem

The move to remote work has created many challenges for tech support as they now support users remotely. One of the biggest problems being faced by employees transitioning to home is their network. According to a recent survey done by Waveform, “Considering more than half (57%) of the U.S. workforce is now working from home because of the COVID-19 outbreak, the 15.5% who face daily connectivity issues represent a large number of people. The Waveform report says more than 10 million U.S. employees working from home are suffering from poor cell signal coverage and daily internet connectivity issues.” Whether it is connecting to Virtual Private Network (VPN), joining video calls, or having slow speeds; the issues are very difficult to diagnose. Without being able to see network setups and relying on users to describe issues, it becomes very difficult to help users.

## 1.2 Solution

Our solution is intended to help the support technicians and users resolve and better diagnose their issues. With the users’ permission, the tech will be able to run multiple network analyses and tests with the results being then displayed on a simple to read web page. While running the tests, the technician will also ask the user questions about their physical network setup and provide them with documentation about best practices and set ups for a work from home network. Information that is collected during the support call is stored and can be referenced later by other technicians experiencing similar issues.

### 1.3 Project Goals

The main goal of HomeNetCheck is to provide remote support technicians with an easier way to diagnose and work with a user having network issues. This includes creating a package to run on a user system, running a database to store information, and creating a web interface for the data collected. Support technicians using HomeNetCheck will have access to all the information collected in a central, easy to read format.

### 1.4 Overview

The rest of this report will detail the process of creating HomeNetCheck and the technology behind it. The report includes goals and objectives, design objectives, technologies used, user profiles, use case diagram, budget, timeline, and problems encountered.

# 2. Discussion

## 2.1 Project Concept

The idea behind HomeNetCheck stemmed from the new work from home landscape that is now prevalent in the COVID-19 world. Working from home has created numerous issues for remote users and remote support technicians. With our diverse work experience in the past year, we have seen many different issues and problems arise from working entirely at home. Many support technicians must run multiple tests and analyses on a user's network if they are working with the user on troubleshooting. HomeNetCheck will package and install existing network analysis tools on a user's personal computer and then after the tools are ran the data collected will be uploaded to a database. That data will then be displayed for the support technician to easily comprehend and hopefully diagnose the issue.

## 2.2 Design Objectives

For our project we wanted to work on a solution to help with the COVID-19 pandemic and the problems generated from the shift to working from home. After multiple brainstorming sessions, we determined that home network issues are a widespread and common issue. Network issues can often be elusive with so many different factors to check and consider. HomeNetCheck aims to provide a single display of network statistics from the network of a user or client facing prolonged issues.

The tools and technologies we will be leveraging to complete HomeNetCheck will be Python, Advanced Installer, Amazon Web Services (AWS), Hypertext Processor (PHP), Hyper Text Markup Language (HTML), JavaScript, Cascading Style Sheets (CSS), Google Charts, MySQL, AngryIP Scanner, Advanced Port Scanner, and Homedale. By utilizing the AWS virtual

tools, we will have a database storing our network analysis data, user logins for the web page, and other system information from the user's computer. The web interface will be using Google Charts to display collected data within the AWS environment as well and be accessible by the affected user and the support technician. The scripting and coding languages will provide the backend support and allow a seamless transfer of data from the machine being analyzed and to the display.

## 2.3 Methodology/Technical Approach

### 2.3.1 Software Packaging and Installation

Using the AutoIT scripting language, we create an executable file that will reference other tools and scripts to collect and send data. AutoIT is designed to be a simple Windows Graphic User Interface (GUI) automation tool to run simple commands and interact with display windows so that a user will not need to. AutoIT uses its own coding language that uses simulated keystrokes, mouse movement, and window manipulation to automate tasks that other languages are not able to interface with. AutoIT was the driving factor in the running and minimizing of network tools, as well as the opening of the documentation page upon installation. The executable file is packaged within an Advanced Installer installation package which will automatically be run after it is run on the computer. The installer provides information about the product to the user, so they are aware of the technical uses for HomeNetCheck and the information it will collect. Advanced Installer provided the ability to run scripts as soon as the user is done with the install wizard.

### 2.3.2 Network Information and Data Transfer

The above-mentioned software installation will install multiple network analysis tools. The combination of those tools and command line-based tests, will result in system and network information. Once, those tools are installed, the support technician will run them and export or log them to a specific location readable by the scripts. With the use of a Python script, the data from the logs of the recently ran tests will be parsed through and inserted into a database automatically. Also, by utilizing command line-based tests the script will quickly insert simple network information to the database. There is an executable file that will perform the above tasks and enter the data to the database while giving the user and technician a progress bar while it runs.

### 2.3.3 Web-based Visualization

All data collected will be displayed on the HomeNetCheck site. The site was created using HTML, CSS, JavaScript and PHP. The home page will give a general overview of our project and its different functions. The Documentation tab will provide guides, tips and instructions on how to use HomeNetCheck and how to interpret the data. Admins will be able to log into the backend of the site with the Login tab. Once logged in, admins will be able to see the results from tests which will be displayed using Google charts.

## 2.4 User Profiles

The three main users for HomeNetCheck will be an end user experiencing network issues, a support technician, and an administrator. The end user is the person who is having network issues and the ones who are benefitted most by this product. The support technician will be the person assisting the end users with their network issues by using the tool. The

administrator is the person who is responsible for maintaining the data, scripts, and analyzation tools. Below are three tables detailing our user profiles.

**Table 1: End User -** The following table is the profile of the End User.

| End User Profile |
| --- |
| **Application:**<br>HomeNetCheck |
| **Potential Users:**<br>Person who is having network issues |
| **Software and Interface Experience:**<br>Accepting terms and conditions, installing base package |
| **Experience with Similar Applications:**<br>Microsoft Outlook<br>Microsoft Word<br>Google Chrome |
| **Task Experience:**<br>Basic computer knowledge, using the simple computer software and programs |
| **Frequency of Use:**<br>Use only when instructed by an administrator |
| **Key Interface Design Requirements:**<br>Basic interface without much interaction from the end user |

**Table 2: Support Technician –** The following table is the profile of the Support Technician.

| Support Technician User Profile |
| --- |
| **Application:**<br>HomeNetCheck Admin Interface |
| **Potential Users:**<br>Information Technology employees, support technician |
| **Software and Interface Experience:**<br>User will need to be familiar with HomeNetCheck Application and basic troubleshooting methods. Available information will be presented in multiple forms. |
| **Experience with Similar Applications:**<br>The support technician should have some experience with network errors and addressing technical problems. HomeNetCheck will be a unique experience compared to other Network Analysis tools. |
| **Task Experience:**<br>HomeNetCheck will have a unique user interface. Support users will be able to easily view data without needing expensive experience. |
| **Frequency of Use:**<br>HomeNetCheck is available for technicians whenever they experience a problem that cannot be addressed without examining network information. |
| **Key Interface Design Requirements:**<br>Design should provide necessary information in an easy to read data presentation. |

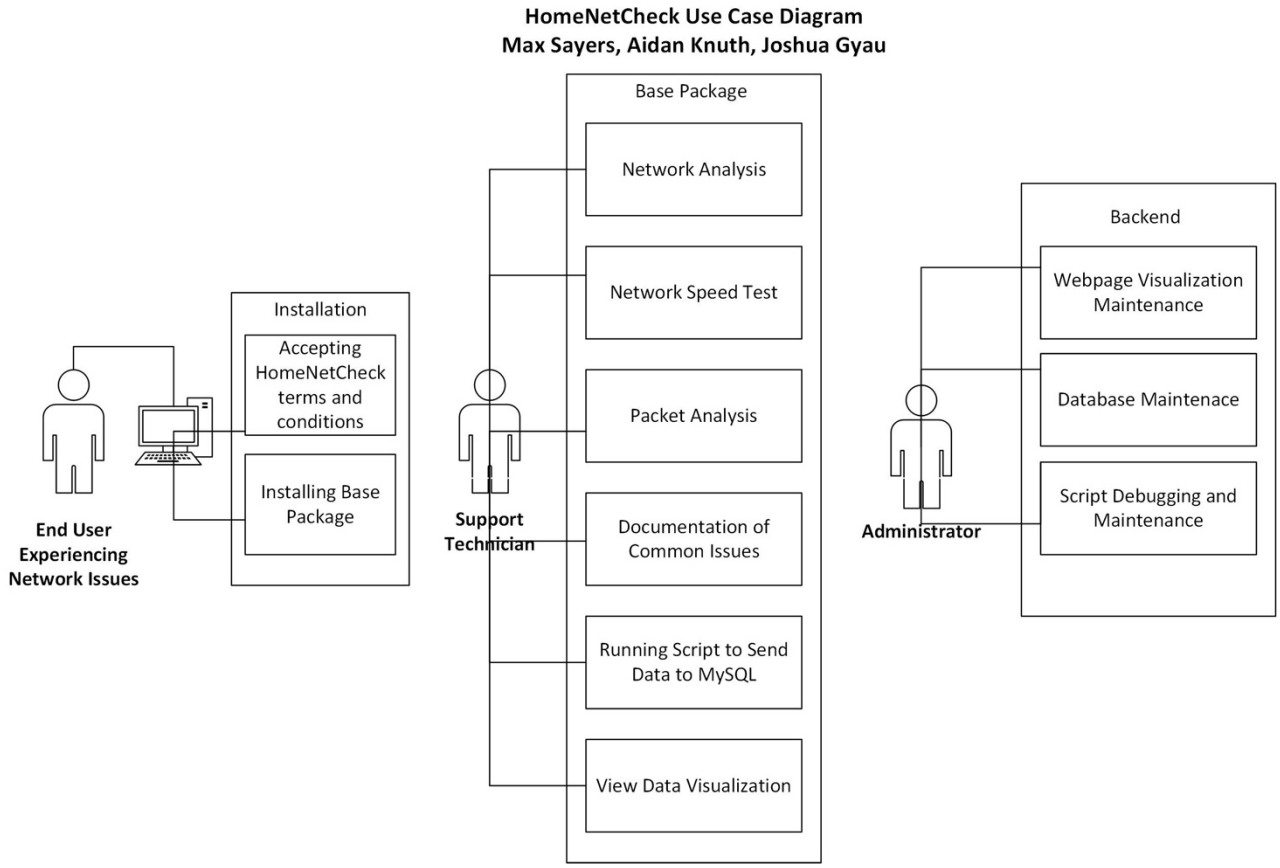
**Table 3: Administrator –** The following table is the profile of the Administrator.

| **Administrator User Profile** |
| --- |
| **Application:**<br>HTML, JavaScript, PHP, MySQL, Batch, Python, Automatic Installer, AWS, Google Charts |
| **Potential Users:**<br>Database administrators, Web developers, other administrators |
| **Software and Interface Experience:**<br>Administrator should be familiar with the scripting languages used in the product, MySQL and MySQL workbench, Google Charts for displaying the data. |
| **Experience with Similar Applications:**<br>The administrator should be familiar with the technologies used for the creation of the product (listed in the application section). |
| **Task Experience:**<br>For the area of expertise in which the administrator is in, they will need to be familiar with the applications in the application section:<br>Database Administrator – MySQL<br>Web Developer – HTML, JavaScript, PHP, Elastic Stack<br>General Administrator – Batch, Python, Automatic Installer. |
| **Frequency of Use:**<br>Any of the administrators will only have to act when there is an issue that requires their attention. For some examples, the data not storing correctly, data virtualization issues, scripting errors. |
| **Key Interface Design Requirements:**<br>The administrator will need to be able to access the software and back end that is HomeNetCheck. The administrator must be well versed in the language/software required for their specific role – database, web, scripting. |

## 2.5 Use Case Diagram

Our Use Case Diagram details how each user will interact with the functions and features of HomeNetCheck. We have associated the tasks with the user profile that will be responsible for the task. The diagram clearly shows the distinct actions and roles associated with HomeNetCheck.

**Figure 1: Use Case Diagram**



## 2.6 Testing

### 2.6.1 Testing Overview

The testing of HomeNetCheck was vital for the progression and continuation in developing our final product. We chose to test two different user roles, the Support Technician and the End User. Testing the role of the Administrator is not currently feasible with our group members as the only subjects for testing. Throughout the development of HomeNetCheck, we thoroughly tested each facet of the product. The support technician would be the person most involved in the use and function of HomeNetCheck. The end user is much less involved and usually accompanied by a support technician, but both user roles will provide insight into the user-friendliness of our product. Testing with individuals outside of our team who were unaware of the details and procedures of HomeNetCheck resulted in very valuable information.

### 2.6.2 Testing Methodology

Our approach to testing was focused on observing the effectiveness and practicality of HomeNetCheck. Due to the constant changes being made to our project throughout the first semester, each team member was responsible for testing the component that they were working on at the time. This allowed us to make lots of changes without the need for constant user testing. Once the tool was ready to be assembled, we moved on to outside users. To achieve the desired feedback, we chose to test how concise and accurate our instructions were to the eyes of a support technician.

Our method for testing was to have a friend, who is also studying Information Technology at the University of Cincinnati use their device as the test use-case, following the role of Support Technician. They will use their own device as the system being observed as if

they were troubleshooting its network issues. We will provide little verbal instruction and observe how effective the instructions on the site are.

### 2.6.3 Scope of Testing

Our testing mainly covered the use of our website and the documentation provided with the tool. HomeNetCheck is being built to run without much work need for the user at all. The user is able to download and run a single application. We wanted to ensure that any level of user can navigate the website and complete the installation. Users were provided only a single link and asked to download the tool for themselves.

The technician role requires a higher level of documentation and technological understanding. This test user was asked to simulate using the tool while a group member acts as a user with network trouble. The goal of this testing was to view the data of the user network from a different network.

### 2.6.4 Testing Objectives

The end goal of testing HomeNetCheck was to determine the usability and functionality in a pseudo remote support situation. There should be little involvement from the end user experiencing the issues, with much of the tasks being placed on the Support Technician assisting that use. Some of the main goals for testing:

a. Overall functionality (how well it works)
b. Support Technician can follow the instructions on the site
c. End User can follow documentation to complete the primary installation
d. Ensuring the website is easy to follow and offers a smooth experience
e. The gathered data can be easily read/interpreted by a technician

### 2.6.5 Test Logs and Procedures

Below is a table which includes the results of a user and technician test of the HomeNetCheck application.

**Table 4: Logs and Procedures**

| Test Number | User Role | Excepted Result | Actual Result | Pass /Fail | Reason | Date |
|---|---|---|---|---|---|---|
| 1 | End User | Success | Success | Pass | User was able to follow the instructions on the site and install the zipped package | 3/3/2020 |
| 2 | Support Technician | Success | Success | Pass | Support Technician was able to follow instructions for the entire process. | 3/5/2020 |

### 2.6.6 Testing Review

For our external testing, we chose to keep it simple. There were two users tasked with testing HomeNetCheck. The first test was done by a person in the role of 'End User'. This individual is someone will little technical knowledge/experience. They were successfully able to download the zipped package from our directions. In the ideal scenario, after the user has installed the zipped package is when the support technician will step in and complete the rest of the process. Our second user test was done entirely by a support technician. This person is a cyber security student at the University of Cincinnati. There were no problems following the instructions and gathering the data. They ran into a problem with the script that sends data to the database, as it would break and not send the data. After some investigation, it was due to the

‘signal.txt’ file being empty. We had to refactor the code and run it with a console window open to see which line it was failing at. After running the installed tools again, data was successfully collected and sent to the database.

With our testing internally, we found many different bugs that needed to be changed. For the Python script that sends data to the MySQL database, testing each individual section of the code as its own script was vital. With each tool, there were different files to be parsed (xml, txt) and displayed. Finding the best way to display what data we wanted was a time-consuming effort that required research and trial. Developing the script into an executable was similar yet more difficult to troubleshoot and test. Instead of a returned error log, the only feedback was that the script failed. Again, isolating sections of the code allowed us to identify where the problems were at. The internal-first testing approach allowed us to present a more refined and complete project for user testing with hopes of more feedback and recommendations being provided. Testing is vitally important for the success of any project and that hold true with HomeNetCheck.

### 2.7 Budget

This budget is the estimation of real-world costs for the technologies and labor required to create HomeNetCheck. Price of usage will vary from company to company due to the scale of the organization.

**Table 5: Budget**

| HomeNetCheck Spending Breakdown | | | | |
|---|---|---|---|---|
| **Product** | **Description** | **Amount/Hours** | **Unit Price** | **Total Price** |
| AWS EC2 | Windows Server hosted on AWS | 1 | $47.01/mo | $564.14/year |
| AWS RDS | MySQL RDS instance hosted on AWS | 1 | $24.82/mo | $297.84/year |
| Web Design | Labor Cost | 200 | $400 | $400 |
| Database Build & Design | Labor Cost | 200 | $400 | $400 |
| Script Automation and Writing | Labor Cost | 200 | $400 | $400 |
| Network Analysis Tools | Free tools used to gain information | 3 | $0 | $0 |
| Code & Code Editors | Virtual Studio Code, AutoIT, Advanced Installer | 3 | $0 | $0 |
| **Total (per year)** | | | | **$2,061.98** |

## 2.8 Timeline

The following table shows the working timeline that our group used to create our project throughout the Fall and Spring semesters. This timeline helped to keep the team on track.

**Table 6: Project Timeline**

| Task # | Task Name | Duration | Start Date | End Date |
|---|---|---|---|---|
| **1** | **Analysis and Research** | **30 days** | **8/24/2020** | **10/5/2020** |
| 1.1 | Contract | 7 days | 8/24/2020 | 8/31/2020 |
| 1.2 | WBS Creation | 14 days | 8/31/2020 | 9/14/2020 |
| 1.3 | Create Elevator Pitch | 14 days | 9/21/2020 | 10/5/2020 |
| **2** | **Project Design** | **7 days** | **10/5/2020** | **10/12/2020** |
| 2.1 | Web Page Design | 5 days | 10/5/2020 | 10/10/2020 |
| 2.2 | Theme Brainstorming | 2 days | 10/10/2020 | 10/12/2020 |
| **3** | **Environment Creation** | **7 days** | **10/5/2020** | **10/12/2020** |
| 3.1 | Server Creation | 3 days | 10/5/2020 | 10/8/2020 |
| 3.2 | Ensure All Members Can Access Resources | 4 days | 10/8/2020 | 10/12/2020 |
| **4** | **Development** | **41 days** | **10/12/2020** | **11/23/2020** |
| 4.1 | Python Script Development | 27 days | 10/20/2020 | 11/16/2020 |
| 4.1 | Database Creation & Maintenance | 35 days | 10/5/2020 | 11/9/2020 |
| 4.2 | HTML Page development | 42 days | 10/12/2020 | 11/23/2020 |
| 4.3 | Installation Package Creation | 14 days | 10/26/2020 | 11/9/2020 |
| 4.4 | CSS & JavaScript development | 42 days | 10/12/2020 | 11/23/2020 |
| 4.5 | Present Alpha Product | 1 day | 11/23/2020 | 11/23/2020 |
| **5** | **Finalization** | **125 days** | **12/8/2020** | **4/12/2021** |
| 5.1 | Testing | 125 days | 12/8/2020 | 4/12/2021 |
| 5.2 | Final Design Tweaks | 59 days | 1/8/2021 | 3/8/2021 |
| 5.3 | Small Feature Implementation | 28 days | 2/14/2021 | 3/14/2021 |
| 5.4 | Documentation | 15 days | 3/14/2021 | 3/29/2021 |
| 5.5 | Presentation Preparation | 15 days | 3/14/2021 | 3/29/2021 |
| 5.6 | Present Final Project | 1 day | 3/29/2021 | 3/29/2021 |
| 5.7 | Prepare for IT Expo | 14 days | 3/29/2021 | 4/12/2021 |
| 5.8 | IT Expo | 1 day | 4/13/2021 | 4/13/2021 |

## 2.9 Problems Encountered and Analysis of Problems Solved

This project has been difficult given the circumstances surrounding this semester. Our group was not able to meet in person because of the COVID-19 pandemic and have relied heavily on video tools and separation of duties. Working individually can be productive but working together as a whole group can allow for better communication. The Spring semester was similar to the Fall in that we proved that meeting virtual was effective.

One of the largest problems was coding everything we needed in Python. We had not had much experience with Python but with much research we were able to utilize it in the way we wanted. A few problems that were encountered were as follows: connecting to the MySQL database, parsing .txt and .xml files, utilizing various modules to gather information, and creating a progress bar. The progress bar was the most challenging. The final solution proved to be rather simple code wise – it involved adding a percentage as the script runs through each section of code. For example, once the speed test was done and the variables were defined, the progress bar increased by 10% as it was 1/10 of the way complete. The program was difficult to run without the console window being open because it was required by the PySimpleGUI module. The solution for that was to have it open and then minimize as soon as the script began. In the end, Python created various issues that were all resolved and made us better Python developers.

One of the main problems in the automation of running the network tools was the interaction between AutoIT and complex GUIs. One example is Homedale, which has a GUI that consists of one all-encompassing window and one smaller window. The smaller window presents data in graphical form and has multiple tabs along the top. These tabs change the information being presented on the graph and provide different data. In order to pull this data from Homedale, the user is required to right click inside the graph and select the save option. The

AutoIT script was not able to simulate this action, and the data still needs to be pulled manually by a technician. With more time the script could be improved and tested until the tools require no action at all. This level of automation would drastically improve the usability of the tool and completely remove the necessity for technical knowledge or instruction.

The biggest problem that was encountered with the Visualization of HomeNetCheck was figuring out best tool to use. Some of the options we explored were Kibana, which is part of the Elastic Stack, and JS Charts. We chose Google Charts in the end because that was the tool that fit our needs for the project. Also, the template we used for the layout of our site cause us some minor issues. One example is that the CSS used unordered list tags to create the navigation bar for the site. This created a problem when it was time to add the documentation content to the site due to not being able to use add bullets, numbers etc.

### 2.10 Possible Improvements for the Future

If we had to create HomeNetCheck all over again with more time to work, there are a few things we would like to change. The biggest feature we would like to add is the automation of the data collection from the installed analysis tools. This proved to be very difficult since each program had its own way of exporting and saving results. If we were to get more time, we would want to investigate some different tools that gather more information and log them differently. With information being logged differently, or automatically, we could utilize a centralized logging system to gather all the logs from the different applications. Another feature we would like to add would be some customization for each organization that would use HomeNetCheck. It would be very useful for each organization to be able to pick and choose what they see and how they see it, most of this would be attainable with Google Charts visualization but would take much more time.

Some suggestions that we have had from others include ensuring the process is easy enough for end users to follow, as well as having useful documentation for both the end users and more technical literate users. We worked hard to create valuable instructions for using HomeNetCheck, but as of right now, the process is tailored for the use of a support technician. With the file names requiring specific locations and extensions, it just isn't simple enough for everyone. With centralized and automated logging, the experience for the end user could blossom. The other documentation we provide is resourceful for both end users and more advanced users. On multiple documentation pages, there are sections with basic and advanced levels. Currently, there are no plans for continuation of HomeNetCheck but we would be open to the possibility if the right situation presents itself.

# 3. Conclusion

## 3.1 Lessons Learned

This project brought success as well as failure. Our team learned that not every idea was viable and changed the topic of our project early in the semester. The project we landed on was more concentrated but required a larger range of technology. HomeNetCheck integrates many tools and technologies into a single project and uses a web site to present information to the user. On a general level we have interacted with many of these features, but the combination of different technologies requires creative thinking and problem solving. Implementing and connecting these various features provided us with more knowledge and information about best practices and methods. We applied what we learned from the Fall semester and improved on our product for the Spring. Ultimately, we are happy with the final product of HomeNetCheck.

## 3.2 Skills Developed

We were not familiar with the AutoIT language before this project but needed to find a way to interact with directly with the Windows Interface. Learning a new language is always challenging and required lots of trial and error to accomplish our specific goals.

To go along with the AutoIT language, additional Python knowledge was required. We had little experience in Python but had never utilized it in this specific way. We never really knew the power of Python before this project, it proved to be such a versatile language and allowed us to complete everything we wanted from file parsing, MySQL connections, Graphical User Interface creation, modules to allow speed tests, ping tests, and more. With enough research, Python can be utilized to complete just about any task or automation. Adding onto Python knowledge, the use of py2exe also was very useful. Py2exe was used to load the script

into an executable file that can be ran on any computer, even ones without Python installed. This feature is invaluable for the use case of an organization where it would be impractical to install Python on its own on each computer. Python was an integral part of HomeNetCheck and is an important skill that we were able to learn throughout the process.

We learned about many visualization tools, their strengths and weaknesses. We used Google charts for the first time while working on this project. We learned the different tools it has and the many different options it offers to visualize data. We also learned different ways of displaying data with data-tables and the different ways of filtering data.

## References

Guta, M. (2020, July 22). Nearly 16% of Those Newly Working from Home Have Connectivity Issues. Retrieved October 5, 2020, from https://smallbiztrends.com/2020/04/work-from-home-connectivity-issues.html