For my assignment I had to program an application that deals with common problems and come up with a solution for it. Since websites are always hosted on a server, I decided to write a program that checks if the website is online and if it is telling the user which status code the response was. I also investigated the current and emerging trends in web development to see if my program was using the latest technology.

While AI is rather new it has become very popular over the last couple of years and is very helpful for debugging and generating sample code. I've personally have used it to debug issues that google wouldn't have been able to resolve. I've used it for very difficult issues that I couldn't work out what the cause was. It has cut the time I spent debugging in half. I've also used it for suggestions and sample code. You must understand how to code as you must modify it and change it to work with your specific project in mind. There is GitHub Copilot which is built into VS code and Visual Studio.

Progressive applications which include offline access push notifications and super-fast loading people are more likely to spend time on a website longer if they are progressive applications.

Serverless Architecture there are many services like AWS or Microsoft Aszure that allow you to deploy website code and projects without having to configure the server manually which you pretty much had to do a couple of years ago.

Web assembly, while it's hard to understand, is very useful because it has a very massive speed increase compared to JavaScript web assembly, which is many times faster than JavaScript which makes it perfect for applications that need to be fast.

Web3 & Decentralized Applications are a very new thing and is quite people there are things that involve blockchains kind of like how bitcoin works that provide security and user control. Token based authentication and smart contracts are very popular too.

https://www.hostinger.com/tutorials/web-development-trends

https://graphicdesignjunction.com/2025/01/future-of-web-development-trends/

Not being able to tell if a website is offline or misconfigured is a very common problem that I decided to solve by writing this application, being able to tell helps with web development and allows the developer to have information they usually wouldn't have access to.

To solve this problem, I wrote a python script that checks if a domain name is online or offline and it uses the requests library to get the response and tell the user how long it took to respond, the status code and if it's online or offline. I also validated and checked the URLs to make sure they are validated correctly.

I decided to choose python as every Linux server comes with it by default and I used the requests framework which allows you to get the status code and responses from a website easily. This was perfect for my solution. I also used the validators framework to validate the string to make sure it is a valid format. Lastly, I used argparse to handle the command line arguments and give feedback to the user.

While researching how to develop this application I looked at how others wrote their code and how they checked if a website was online and I improved on it and made it more advanced and easier to use. I used the following resources.

https://stackoverflow.com/questions/1949318/checking-if-a-website-is-up-via-python

https://learningactors.com/learn-how-to-create-a-website-status-checker-in-python/

I decided to choose python for this project because python is included on all servers by default while NodeJS isn't so it would run on any machine running Unix or Linux which saves the developer time from having to install and set up a complete NodeJS installation. The most popular Linux distribution is ubuntu and python 3 comes with ubuntu by default. I'm more comfortable with python and it's easier to write small scripts compared to JavaScript. I decided to use requests as that is the easiest library in python to do it. The other libraries I could have used were HTTP. Client socket or aiohttp. I didn't choose socket as it doesn't give http status and only checks via DNS. Aiohttp is too complicated to implement, and it was only one URL, so I didn't need it to be asynchronous. Lastly HTTP. Client is quite limited.

https://stackoverflow.com/questions/11057926/what-are-the-list-of-os-that-have-python-preinstalled-and-which-versions

https://unix.stackexchange.com/questions/24802/on-which-unix-distributions-is-python-installed-as-part-of-the-default-install

https://askubuntu.com/questions/1396920/does-ubuntu-20-04-come-with-the-python-standard-library

https://www.speakeasy.com/blog/python-http-clients-requests-vs-httpx-vs-aiohttp

https://docs.aiohttp.org/en/stable/client_reference.html

When creating and designing this program I had to consider the ethical and technical issues of creating a program that checks if a website is online and comes across the following possible issues.

Ethical Issues.

Privacy

The library I used to request the information at could unintentionally gather Ip addresses headers and cookies.

Permissions.

Sometimes the owner of a server doesn't want the server to be pinged so that could be considered a violation of permission.

Straining

If the web server is small, it could easily bring the server down and many people using this application at once could pretty much take down a small server unintentionally/

Terms of Service.

It could be a violation of the terms of service to connect and request the data this application needs.

Technical Issues.

Incorrect information.

There might be times when the application incorrectly assumes it is offline or online when it isn't.

Bans and triggering firewalls.

While my program was written in a way where it only requests the information once if it wasn't it could request data from the server nonstop and end up with thousands of requests getting the Ip address of the user banned.

https://www.forskningsetikk.no/en/guidelines/social-sciences-and-humanities/a-guide-to-internet-research-ethics/

https://dev.to/manageengineapm/challenges-in-website-monitoring-48en

When designing the application I had a basic idea of how to do it but the idea I had was more complicated then it had to be I was thinking about doing a DNS level check see if there are A records and if it can connect to the A record I learned all I had to do was use the request library and gather information from the function do some if statements for the status codes and tell the user what the result of connecting to the server is. My ability to plan and design a program has gotten better when developing and implementing my ideas. I would think of everything I need to see if there is an easier and simpler way to accomplish it and how to plan stuff better by thinking more about the issues I'm dealing with.

What could I have done differently if I did this again in the future? I would have made the application handle multiple URLS using something like threading and I could have encrypted the information from the program and decrypted it and loaded it back into the program and showed the users what was stored in status.txt. I also could have added a command line argument that tells you specific things about the server such as the current IP address.

I was going to add a GUI using Tkinter or GTK but due to time constraints and how soon the assignment was due I couldn't implement it and I did think about it and it wouldn't make too much sense to have a Gui if it's just one URL being checked.

Installation and program documentation

You first need to download the application and put the source code into a local folder, it's just the main.py script and you will need to set up a virtual environment and install all the dependencies and libraries.

I used the linux subsystem for windows and the commands to install the dependencies and virtual environments are the following.

```
Python -m venv websitechecker
```

Then we activate the virtual environment we use the following command.

```
Source venv/bin/activate
```

We then install the requirements using the text file with the libraries in them no need to install of them one by one.

```
Pip install -r requirements.txt
```

To use the application, you need to provide the script then the arguments you need to use the –-url argument and include a valid domain name that has to be in the format of http:// or https://

```
Main.py –url https://jessicaamy.com
```

If you don't use a command line argument and just use the application it will run a loop and keep asking you for a domain and check if that domain is okay.