We will scan the subdomain and determine whether it is up or down in this article. Every minute, the script should automatically check the status and update it on the screen in a tabular format.

Pre requirements: Install the below mentioned modules before start your script.

pip install requests --> request module, which makes it simple for us to send HTTP requests to websites in order to obtain information.

pip install tabulate --> helps us to get the output in tabular format.

Note: Try using the pip install requests --user command to install the modules if you encounter any errors during installation.

URL structure: mostly url contains fore parts

1.Protocol

2. sub-domain

3. Domain name

4. Top level domain

Let's create a text file within a subdomain that lists submains that you can find online at random, sample file mentioned below.

subdomain.txt

mail

www

blog

localhost

ftp

smtp

dev

Execution:

* The subdomain.txt file, which contains a list of subdomains, should first be created.

* Import the required modules has mentioned above.

* Make a subdomain scanning function and pass the subdomain and domain name as parameters.

* You can now concatenate the subdomain with the domain name for each subdomain in the list using a for loop, and store the result in a variable called url.

* create the URL by f string with protocol, subdomain, and domain name.

```
url = f"https://{subdomain}.{domain_name}"
```

* Use request.get() function that help us to sending get request from the url, it's recomended to use try catch block method to avoid program crash.

* If we get the status code as 200 then that service is up or else considered as down,that program will continuosly print the status whether up or down with every one minute interval of time.

```
status = "up" if response.status_code == 200 else "down"
```

* use tabulate to get the result in tabel format.

Program:::

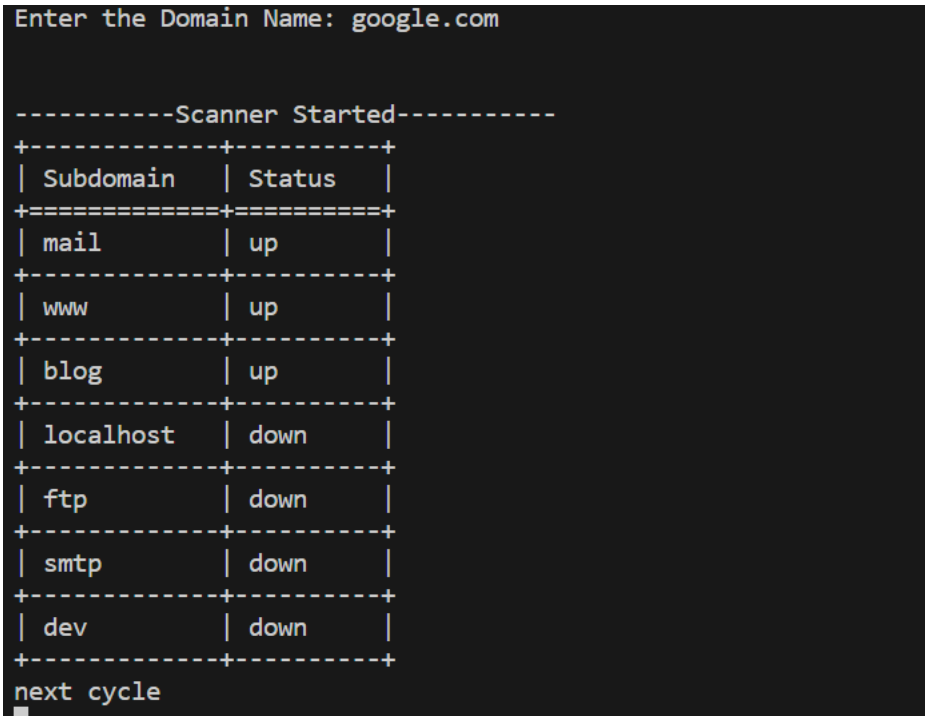
subdomain_check.py

```
import requests
import time
from tabulate import tabulate


# Function for scanning subdomains
def domain_scanner(domain_name, sub_domnames):
    print('-----------Scanner Started-----------')
    while True:
        results = []  # List to store results


        # Loop for getting URLs
        for subdomain in sub_domnames:
            url = f"https://{subdomain}.{domain_name}"  # Making URL by putting subdomain one by one
```

```
            try:
                # Sending GET request to the URL
                response = requests.get(url)

                # If the URL is valid (status code 200), store it as "up"
                status = "up" if response.status_code == 200 else "down"
            except requests.ConnectionError:
                # If URL is invalid, store it as "down"
                status = "down"

            # Append the subdomain and its status to the results list
            results.append({"Subdomain": subdomain, "Status": status})

        # Display the results in tabular format
        table = tabulate(results, headers="keys", tablefmt="grid")
        print(table)
        time.sleep(60) # Wait for 1 minute before checking again
        print("next cycle") #message after one itration


# Main function
if __name__ == '__main__':
    domain_name= input("Enter the Domain Name: ")  # Input the domain name
    print('\n')

    # Opening the subdomain text file
    with open('subdomain.txt', 'r') as file:
      sub_domain = file.read().splitlines()  # Reading the file and splitting lines to get subdomains
```

```
domain_scanner(domain_name, sub_domain) # function call
```

Output: After executing the code you can able to see the output as mentioned below.



Thank you!! Happy Learning!!!