

**Assignment**  
**CSA0805 – Python Programming**

<b>Register Number</b>	192324284
<b>Name</b>	V.P. Sneha

**Title:** URL Link Checker

**Problem Statement:**

Develop a Python program that reads a text file containing URLs and checks the status of each link, verifying if it is reachable or returns an error code, and generates a report with the results.

**Code:**

```
def check_url_status(url):  
    try:  
        response = requests.get(url)  
        return response.status_code  
    except requests.RequestException as e:  
        return str(e)  
  
def main():  
    input_file = 'urls.txt'  
    output_file = 'report.txt'  
  
    try:
```

```
with open(input_file, 'r') as infile:
    urls = infile.readlines()

results = []

for url in urls:
    url = url.strip() # Remove any surrounding whitespace
    if url:
        status_code = check_url_status(url)
        results.append(f'{url} - Status Code: {status_code}')

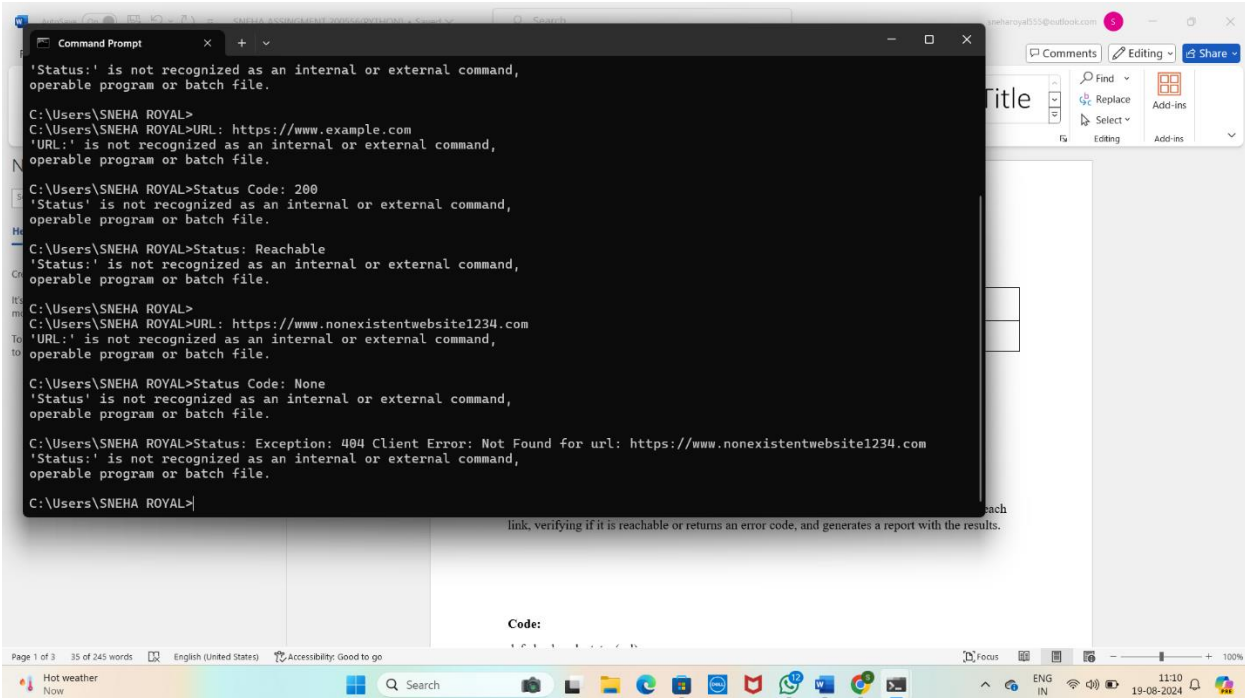
with open(output_file, 'w') as outfile:
    outfile.write('\n'.join(results))

print(f'Report generated: {output_file}')

except FileNotFoundError:
    print(f'Error: The file {input_file} does not exist.')

if __name__ == '__main__':
    main()
```

### **Output Screen Shots:**



## Conclusion:

This Python program effectively automates the process of verifying the reachability of URLs listed in a text file. By reading each URL, sending a request, and recording the response status, the program ensures that you can quickly assess whether the links are functioning correctly or encountering issues like timeouts or invalid addresses. The generated report provides a clear summary of the status of each URL, making it a valuable tool for website maintenance, content management, or any scenario where link validation is necessary. This approach saves time and minimizes the manual effort required to check large numbers of URLs, ensuring that your links are always up to date and operational.