

CN PROJECT REPORT

## **Proxy Server using Web Cache**

Group leader: 21K-4824 Asadullah Wagan

Group members: 21K-3281 Rana Wahaj Ahmed- 21K-3156 Syed Abdul Rehman

**Introduction:**

Our project entails the development of a Proxy Server system utilizing Python programming language. A proxy server acts as an intermediary between client applications and the internet. It intercepts requests from clients seeking resources from other servers and then forwards those requests to the appropriate server. The proxy server also caches responses from the server, thus improving performance by serving subsequent requests directly from the cache.

**Project Objective:**

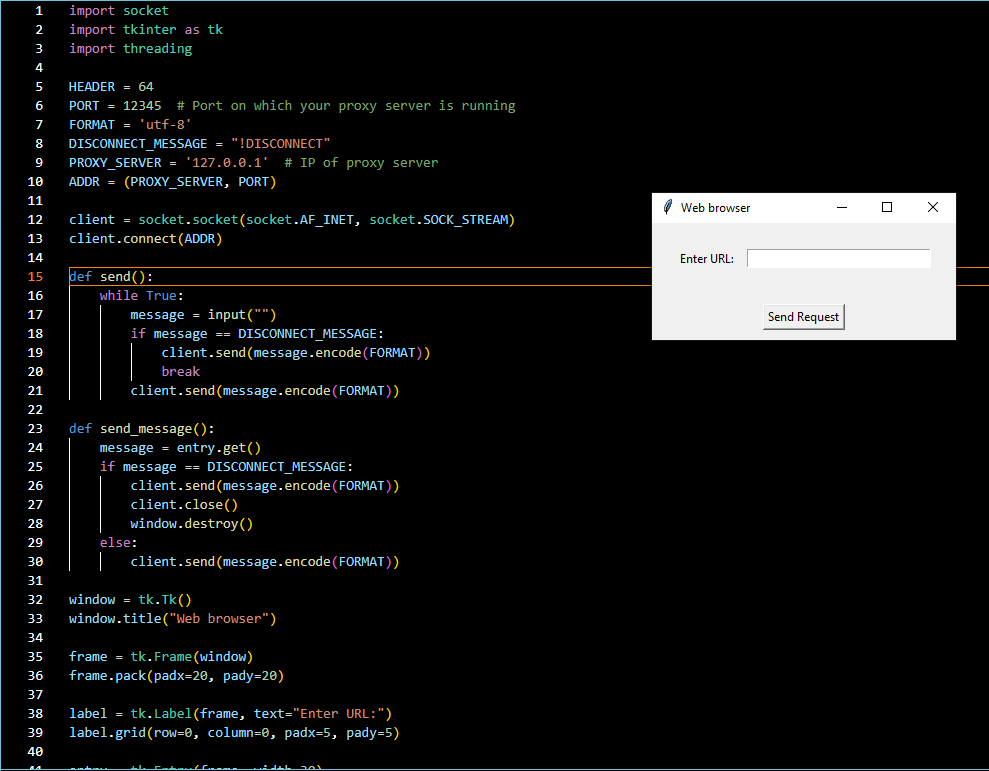
The primary objective of this project is to design and implement a proxy server system that facilitates efficient handling of client requests, caching of responses, and seamless communication with main servers.

**Project Components:**

1. **Client Code:** The client code establishes a connection with the proxy server and sends HTTP requests for desired resources. It utilizes the socket library for network communication and tkinter for the graphical user interface (GUI) to input URLs.
2. **Main Server Code:** This component represents the main server that fulfills client requests forwarded by the proxy server. It uses socket programming to handle incoming connections from the proxy server, processes the requests, and responds with the requested web content.
3. **Proxy Server Code:** The proxy server acts as an intermediary between the client and the main server. It intercepts client requests, forwards them to the main server, caches the responses, and returns the requested content to the client. The proxy server code utilizes threading for concurrent handling of client connections and caching to store frequently accessed web content.

**Functionality:**

* **Client Interaction:** The client interacts with the proxy server through a GUI where the user inputs URLs for desired web resources. Upon receiving the URL input, the client sends HTTP requests to the proxy server to fetch the corresponding web content.

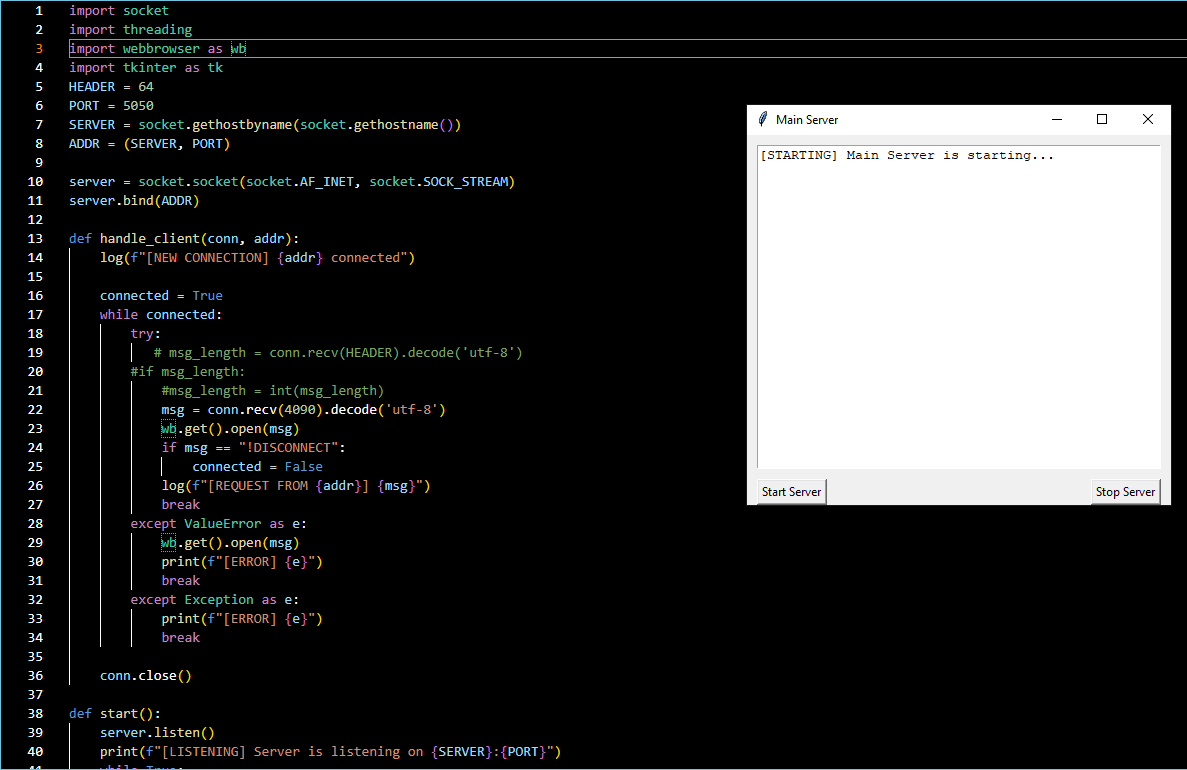


* **Proxy Server Operation:** The proxy server listens for incoming client connections and handles requests by fetching content from the main server or serving cached responses. It maintains a cache to store frequently accessed resources, thereby optimizing performance by reducing the need to fetch data from the main server repeatedly.

A screen shot of a computer

Description automatically generated

* **Main Server Communication:** The proxy server communicates with the main server to fulfill client requests. It forwards HTTP requests from clients to the main server and relays the responses back to the clients after caching.



**Conclusion:** In conclusion, the implementation of the proxy server system provides a robust solution for efficient web resource retrieval and caching. By leveraging Python's socket programming capabilities and GUI libraries like tkinter, we have developed a functional system capable of enhancing network performance and optimizing resource utilization.