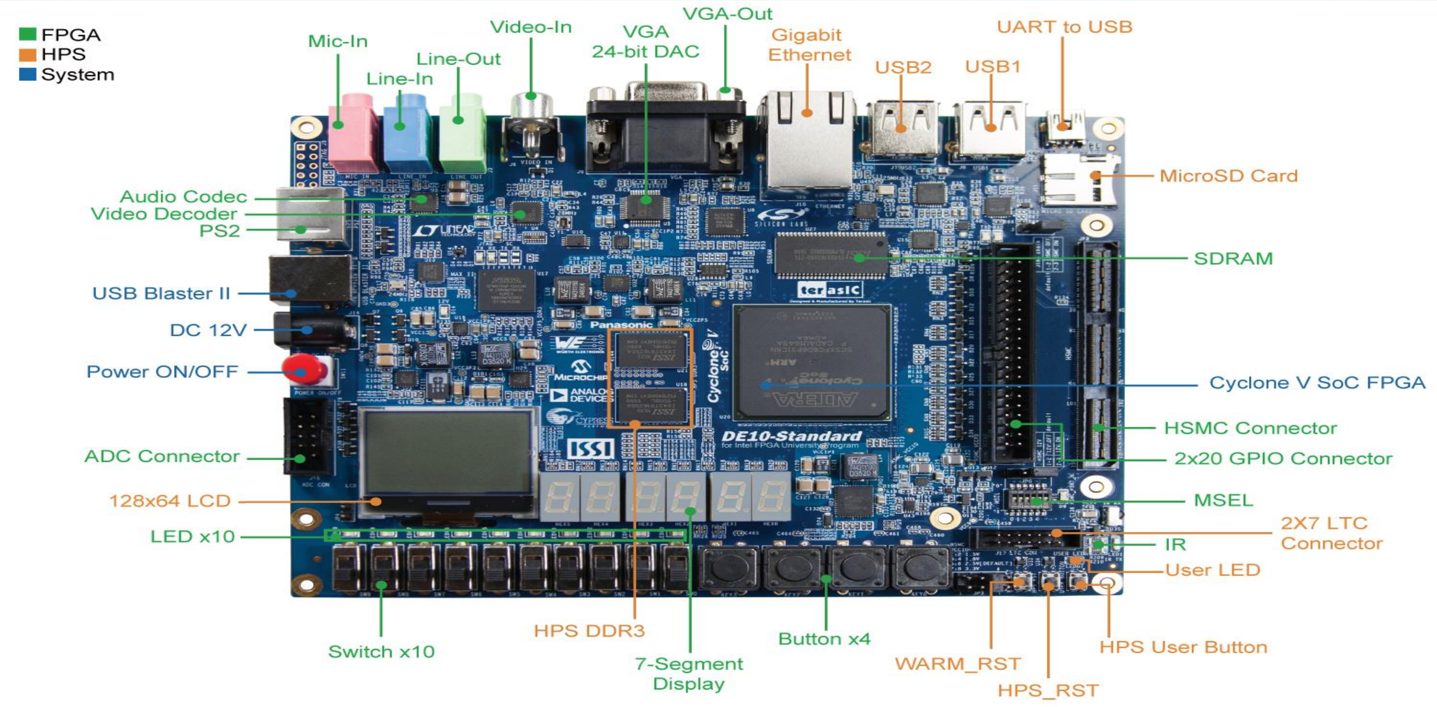
CIE WORKSHOP

SoC - Track

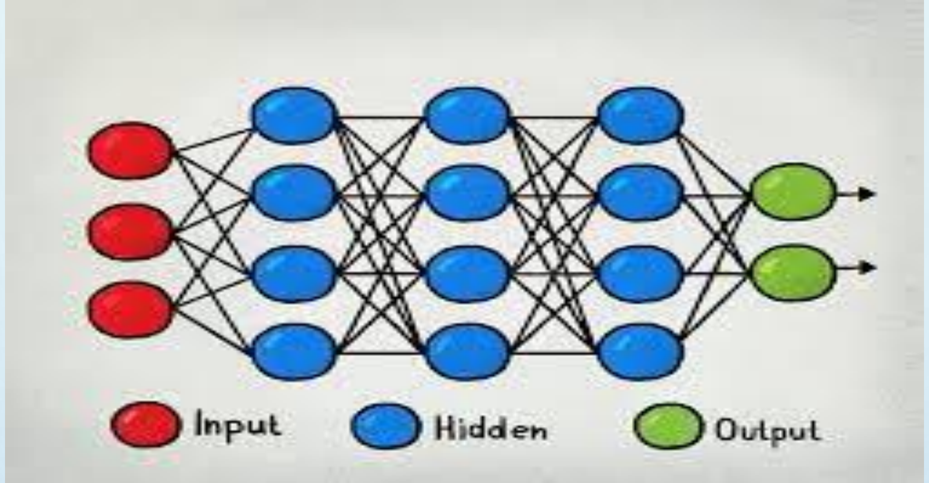
WHAT DID WE LEARN ?

DE-10 BOARD



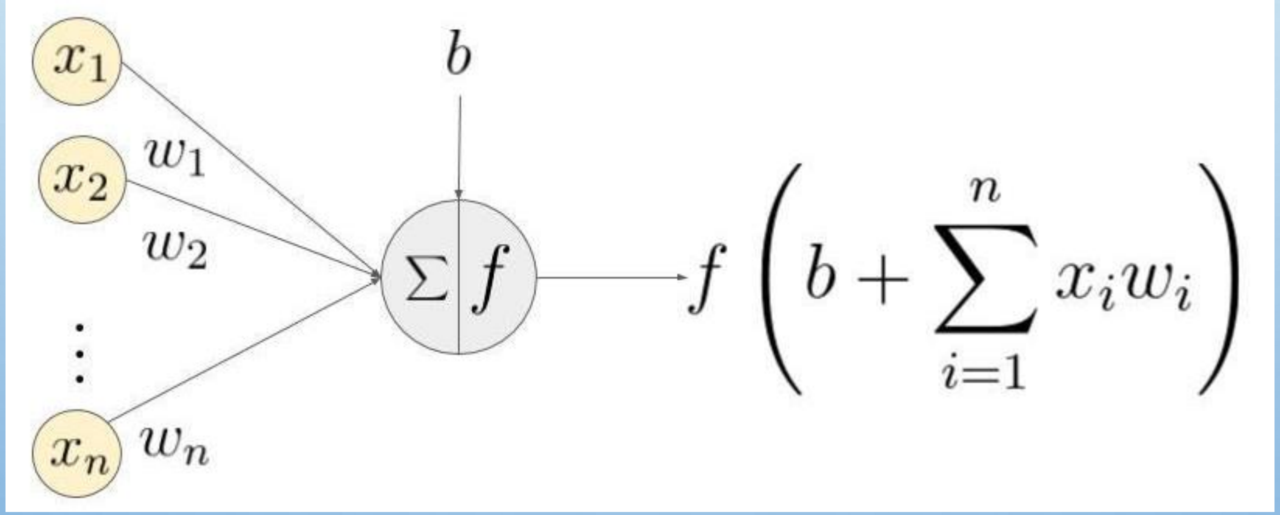
- The DE10-standard board is a popular development board designed by Terasic based on the intel (formerly Altera) cyclone V FPGA. It offers a versatile platform for prototyping and implementing various digital logic and embedded systems projects
- **We downloaded some software packages to work with DE 10 board**
- **Board configuration, ex: SoC “10101”**
- **FPGA connection between peripheral parts**

NEURAL NETWORK



- **Introduction about image processing and opencv library**
- **We learnt about neural network and image processing such as image detection and image generation.**
- In image processing, a neural network refers to a computational model inspired by the structure and functioning of the human brain's neural networks. Neural networks are designed to learn and recognize patterns in data, making them particularly useful in image processing task

- Overleaf(latex) which helps to follow the format to publish research paper and also for capstone project report
- DE 10 standard board their peripherals connected to HPS and HPS installed on board
- Executing procedure on chip and execution of program on chip
- Learnt some mathematical functions of neural network



CLI PROXY SERVER USING MULTI-THREADING

WHAT IS A PROXY SERVER?

- A proxy server is an intermediary server that acts as a gateway between a server and many clients.
- When a client makes a request to access a resource, it sends the request to the proxy server instead of directly contacting the destination server.
- The proxy server then evaluates the request and, depending on its configuration, may perform various functions :

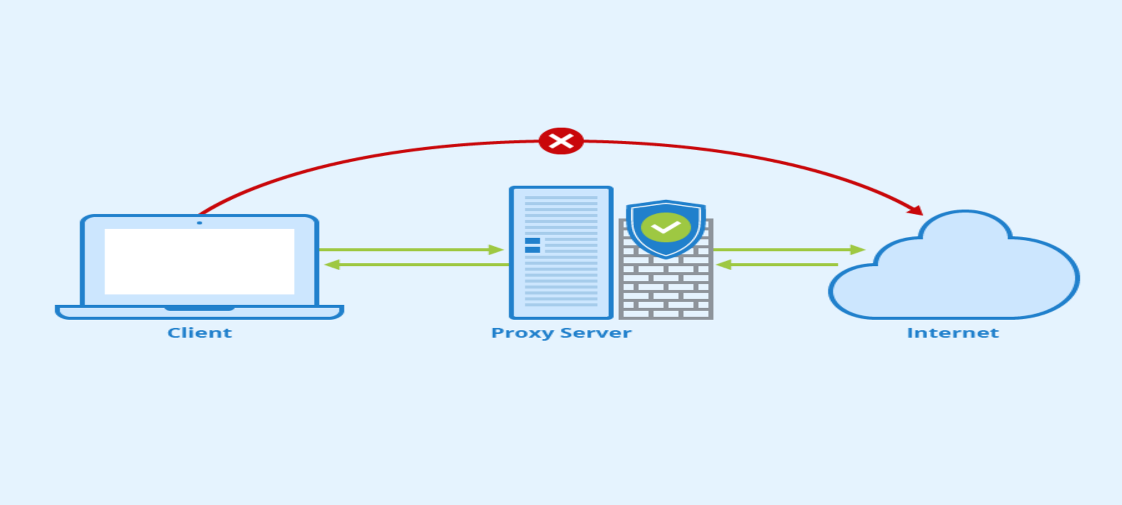
caching

Anonymity and privacy

Content filtering

Load balancing

Security and firewall protection



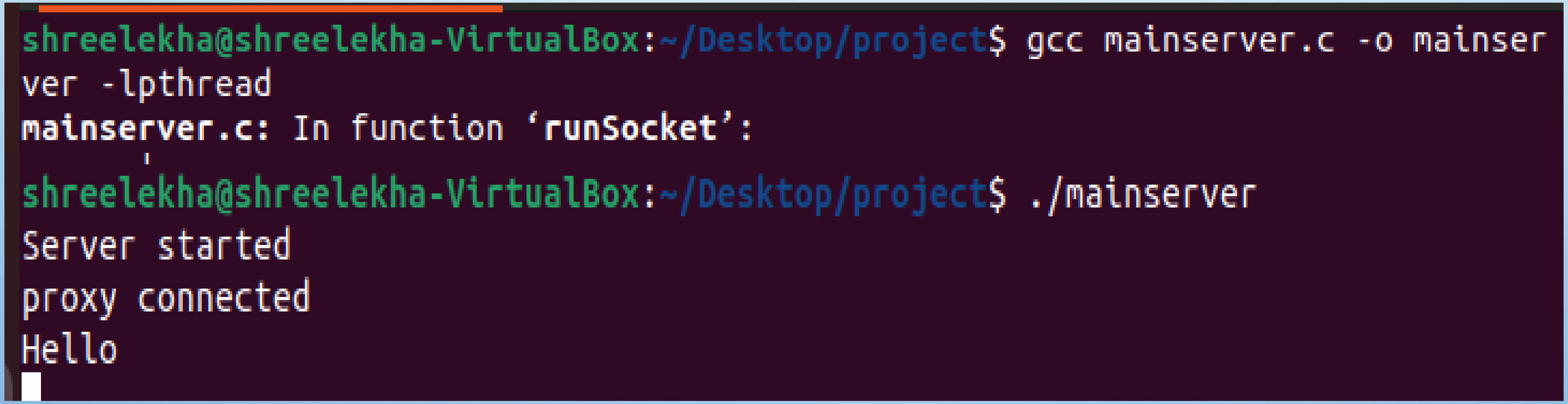
STEPS INVOLVED IN IMPLEMENTING A PROXY SERVER ON THE DE-10 BOARD

- Initialize the DE-10 board and set up the necessary network interfaces.
- Accept proxy details from the client, including the proxy IP address and port, through a socket connection.
- Store the client-provided server details (IP address and port) in variables within the DE-10 board.
- Create a socket on the DE-10 board at the specified proxy port and start listening for incoming client connections continuously on that port.

- For each successfully accepted client connection, create a new thread within the DE-10 board and pass the client details (client file descriptor) and server details (IP address and port) to that thread. Let's call this thread the "client thread."
- Inside each client thread, establish a socket connection to the main server using the provided server details (IP address and port). This client thread will be responsible for communication between the client and the main server.
- Meanwhile, the main server, running independently from the DE-10 board, should accept client connections and create its own thread for each client. These threads are known as "server threads" and facilitate communication between the main server and the clients.
- The client thread within the DE-10 board and the server thread within the main server will handle the communication between the respective clients and the main server, ensuring traffic differentiation.

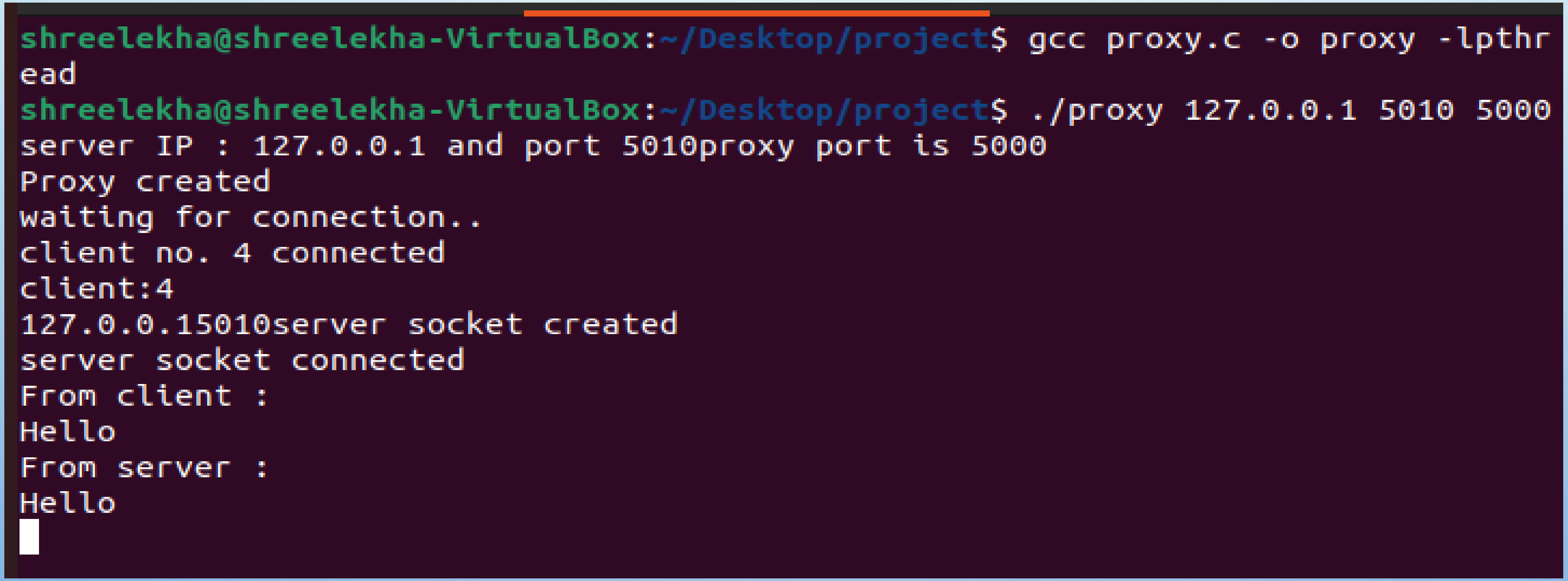
OUTPUT

MAIN SERVER CODE



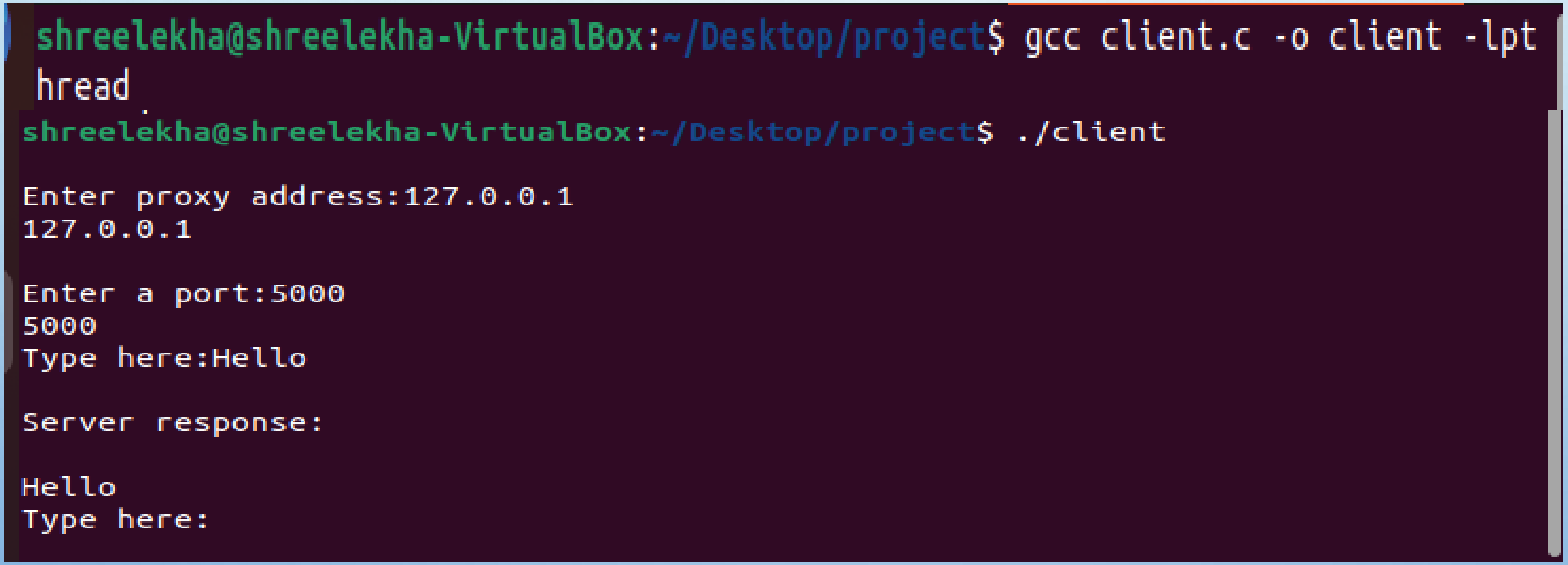
```
shreelekha@shreelekha-VirtualBox:~/Desktop/project$ gcc mainserver.c -o mainser
ver -lpthread
mainserver.c: In function ‘runSocket’:
shreelekha@shreelekha-VirtualBox:~/Desktop/project$ ./mainserver
Server started
proxy connected
Hello
```

PROXY SERVER CODE



```
shreelekha@shreelekha-VirtualBox:~/Desktop/project$ gcc proxy.c -o proxy -lpthr
ead
shreelekha@shreelekha-VirtualBox:~/Desktop/project$ ./proxy 127.0.0.1 5010 5000
server IP : 127.0.0.1 and port 5010proxy port is 5000
Proxy created
waiting for connection..
client no. 4 connected
client:4
127.0.0.15010server socket created
server socket connected
From client :
Hello
From server :
Hello
```

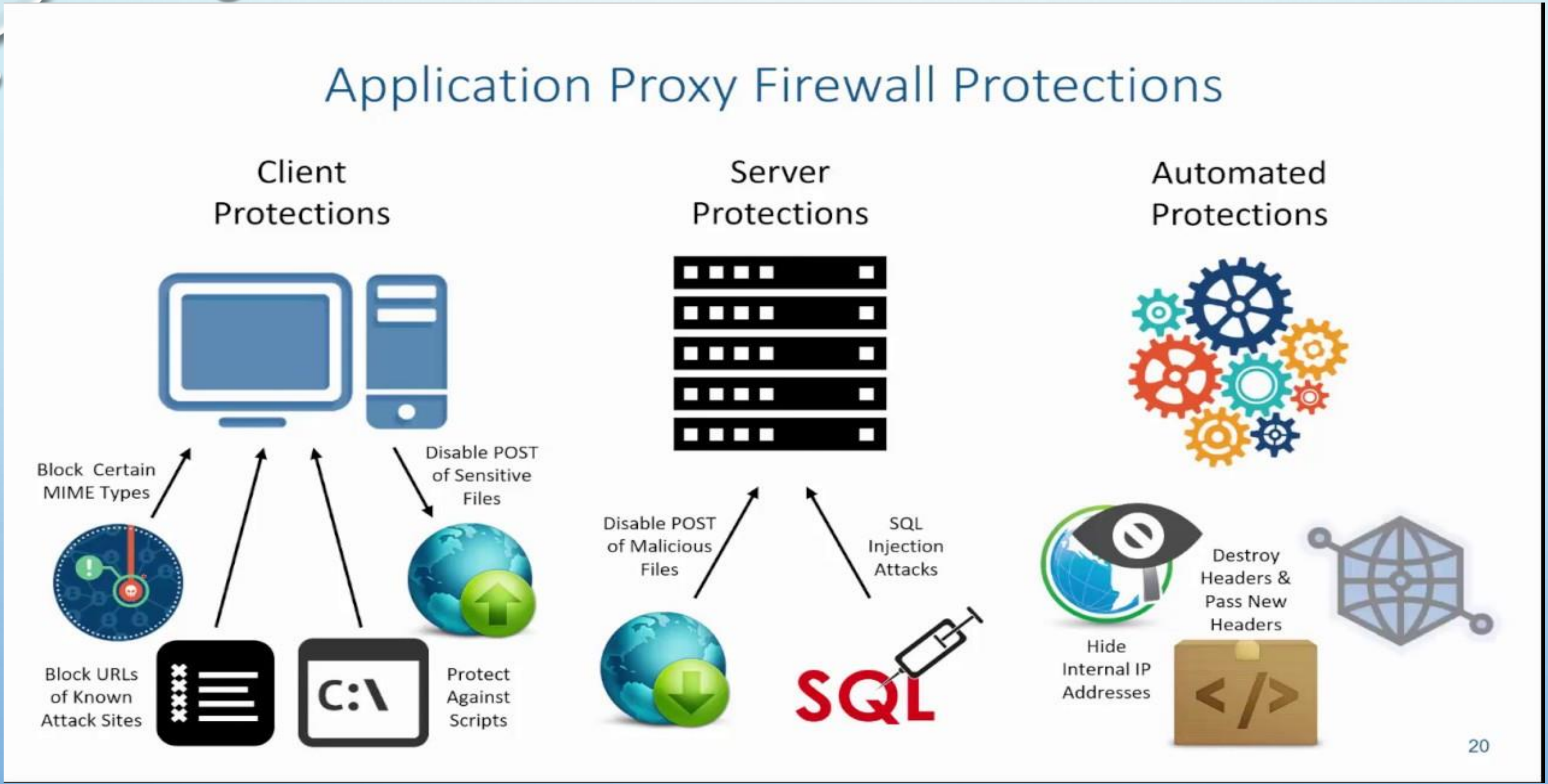
CLIENT CODE



```
shreelekha@shreelekha-VirtualBox:~/Desktop/project$ gcc client.c -o client -lpt
hread
shreelekha@shreelekha-VirtualBox:~/Desktop/project$ ./client

Enter proxy address:127.0.0.1
127.0.0.1

Enter a port:5000
5000
Type here:Hello

Server response:

Hello
Type here:
```

APPLICATIONS AND FUTURE WORK

- **Implementing request pipelining or multiplexing**: When there are multiple requests between the client and server, we can implement request pipelining or multiplexing. These techniques allow multiple requests to be sent concurrently over a single connection, reducing the overall latency and improving throughput.
- **Optimize network protocols:** Evaluate the network protocols being used for communication and consider optimizing them for better performance. For example, you could switch to a more efficient protocol like HTTP/2 or QUIC, which can reduce latency and improve overall network efficiency



Application Proxy Firewall Protections

Client Protections

Block Certain MIME Types

Block URLs of Known Attack Sites

Protect Against Scripts

Disable POST of Sensitive Files

Server Protections

Disable POST of Malicious Files

SQL Injection Attacks

Automated Protections

Hide Internal IP Addresses

Destroy Headers & Pass New Headers

- **VPN (Virtual Private Network):** VPN’s use proxies to establish secure communication channels between clients and remote servers. VPN proxies encrypt network traffic, providing users with privacy and anonymity while accessing the internet. They are commonly used in organizations to enable secure remote access to internal resources.
- **Email proxies:** Email proxies are used to filter incoming and outgoing email traffic. They perform tasks like spam filtering, virus scanning, content filtering, and enforcing email policies. Email proxies help organizations protect their email infrastructure and users from malicious or unwanted content.

THANK YOU

TEAM MEMBERS

N Shreelekha

S Sricharan

Rajith M

Vaishnavi C K