**INDUSTRIAL TRAINING REPORT**

ON

**PYTHON PROGRAMMING**



FROM

**INDIAN SCHOOL OF ETHICAL HACKING**

**[](https://www.isoeh.com/)**

TOPIC:

**DEVELOPING CHAT APPLICATION IN PYTHON**

PREPARED BY:

**PUJA NANDI**

**3rd year, Department of Computer Science and Engineering**

**Hooghly Engineering and Technology College, Pipulpati**

**University Roll Number: 17600117008**

**CONTENTS**

1. ABSTRACT
2. THE CODE
3. THE SERVER SCRIPT
4. THE CLIENT SCRIPT
5. THE RESULT
6. LIBRARIES USED
7. TOOLS USED
8. ACKNOWLEDGEMENT
9. REFERENCES

**ABSTRACT**

This project demonstrates – How to set up a simple Chat Room server and allow multiple clients to connect to it using a client-side script. The **Chat Application** is very common today offered either via a **web application** or **mobile application**. Learning to write a **Chat Application** is good for understanding many **network communication** concepts and can be useful to build other **network applications**. **Chat Application**provides communication between two parties i.e. **sender** and **receiver**.

The **sender** is someone who initiates and send a message to other known as **receiver**; receiver at other end receives the message. The role of **sender** and **receiver** is not fixed and keep exchanging during communication, so in simple words, at a point, someone who sends the message is a **sender** and who receive the message is called **receiver**.

Generally, in real-world communication is done directly using voice in an ideal situation (distance between communicating parties, identification of parties) where **sender** speak out and the intended **receiver** respond after listening. So, what is the **medium of communication**here? Indeed, it is the air which helps our voice to travel to the **receiver** and successful communication depends upon air (high wind and long-distance can cause trouble). In online or digital communication, the role of air is played by network channel (coaxial cable, fiber optics, etc.) and communication is controlled by a **server**. A **server** is a program which regulates the communication between **sender** and **receiver**.

So, to create a **Python** **Chat Application**, one has to write a **server** program and **client** program/s (**sender** and **receiver**). Suppose, two parties Alice and Bob want to chat with each other and ask you to develop a chat application then being a developer you have to write a **server** **program** and a **client** **program**

**THE CODE**

THE SERVER SCRIPT

**Server program** has all the logic to control and regulate the **Chat**, so most of the chat logic is implemented with a **server** program. So first step of communication is to identify the users, how server do this? In network communication, users are identified by a **socket** which is nothing but a combination of **IP address** and **port address**. So, for human understanding, Alice and Bob will be chatting but for a **network**, it is two **sockets** process which is sending and receiving bytes.

Steps involved in this process is as follows:

1. Create socket
2. Communicate the socket address
3. Connect to client
4. Receive the message
5. Decode the destination user and select the socket
6. Send a message to the intended client
7. Keep repeating step 5 & 6 as per users wish

**THE SERVER SIDE CODE**

import socket

import sys

import time

##end of import###

###init ###

s=socket.socket()

host=socket.gethostname()

print ('server will start on host:',host)

port=8080

s.bind((host,port))

print("")

print("server done binding to host and port successfully")

print("")

print("server is wating for incomming connection")

s.listen()

conn,addr=s.accept()

print(addr,"has connected to the server and is now online..")

print("")

while 1:

message=input(str(">>"))

message=message.encode()

conn.send(message)

print("message has been sent..")

print("")

incomming\_message=conn.recv(1024)

incomming\_message=incomming\_message.decode()

print("")

print("client:",incomming\_message)

THE CLIENT SCRIPT

**Client** script is run by the user, so the same **client** code will be run by a different user but each will have a separate **socket** so they will have their unique **communication channel**. Client script uses to be thin because it has very less work i.e. it only connect with the **server** and send and receive messages.

The steps involved in **client script** are:

1. Create a unique client socket per instance/user
2. Connect to the server with given socket address (IP and port)
3. Send and receive messages
4. Repeat step 3 as per configuration

**THE CLIENT SIDE CODE**

import socket

import sys

import time

s=socket.socket()

host=input(str("please enter the host name of the server:"))

port=8080

s.connect((host,port))

s.close()

print("connected to chat server")

while 1:

incomming\_message=s.recv(1024)

incomming\_message=incomming\_message.decode()

print("")

print("server:",incomming\_message)

message=input(str(">>"))

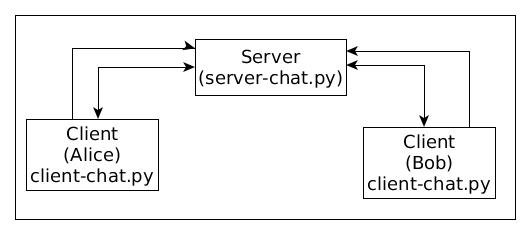
message=message.encode()

s.send(message)

print("message has been sent..")

print("")

THE FLOW DIAGRAM OF THE APPLICATION



**OUTPUT**

**LIBRARIES USED**

Python socket

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection

Python sys

The sys module provides functions and variables used to manipulate different parts of the Python runtime environment.

Python time

Python has defined a module, “time” which allows us to handle various operations regarding time, its conversions and representations, which find its use in various applications in life.

**TOOLS USED**

Coding language: Python3

Editor: Anaconda

**ACKNOWLEDGEMENT**

I would like to express our special thanks of gratitude to my instructor **Mr. Subhendu Bhadra** as well as The Indian School of Ethical Hacking, who gave me the golden opportunity to do this wonderful project on the topic **Developing chat application in python**, which also helped me in doing a lot of Research and I came to know about so many new things & I am really thankful to them.

Secondly, I would also like to thank my friends & teachers who helped me a lot in finalizing this project within the limited time frame.

**REFERENCES**

* *[https://www.python.org/](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwj6sKKN7sTjAhUEbisKHRG1DBoQFjAAegQICBAC&url=https%3A%2F%2Fwww.python.org%2F&usg=AOvVaw0QREvGsjwHKp2GtoYvs1JH)*
* *[https://www.w3schools.com/python/](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=6&cad=rja&uact=8&ved=2ahUKEwj6sKKN7sTjAhUEbisKHRG1DBoQFjAFegQIAhAB&url=https%3A%2F%2Fwww.w3schools.com%2Fpython%2F&usg=AOvVaw3DDMaeRvoPZxdqsRK3yoWF)*
* *[https://www.geeksforgeeks.org/python-programming-language/](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=8&cad=rja&uact=8&ved=2ahUKEwj6sKKN7sTjAhUEbisKHRG1DBoQFjAHegQIBhAB&url=https%3A%2F%2Fwww.geeksforgeeks.org%2Fpython-programming-language%2F&usg=AOvVaw16lCHt-9nlAhg8u2e92eD8)*
* *[https://www.tutorialspoint.com/python/](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=18&cad=rja&uact=8&ved=2ahUKEwj6sKKN7sTjAhUEbisKHRG1DBoQFjARegQIARAB&url=https%3A%2F%2Fwww.tutorialspoint.com%2Fpython%2F&usg=AOvVaw2QBXg0qSlQEOe4IQcNFvbb)*