

# ONLINE INVENTORY MANAGEMENT SYSTEM

## NAMES OF PARTICIPANTS:

1.SACHIN RAMESH KULKARNI   PES2UG21CS449

2.SAI LITHISH DEGAPUDI       PES2UG21CS456

## SERVER CODE:

```
import socket

# Define the protocol
PROTOCOL = {
    "browse": 1,
    "purchase": 2,
    "response": 3
}

# Define the server address and port
SERVER_ADDRESS = "localhost"
SERVER_PORT = 9999

# Define the inventory
inventory = {
    'iPhone 13': 59,
    'iPhone 12': 31,
    'iPhone 11': 8,
    'iPhone SE': 7,
    'iPhone XR': 4,
    'Galaxy S21': 50,
    'Galaxy S20': 45,
    'Galaxy S10': 32,
    'Galaxy Note 20': 56,
    'Galaxy A52': 120,
    'Pixel 6': 70,
    'Pixel 5': 40,
    'Pixel 4a': 37,
    'Pixel 3': 20,
    'Pixel 2': 3,
    'OnePlus 9 Pro': 98,
    'OnePlus 8 Pro': 76,
    'OnePlus 7T': 130,
```

```

        'OnePlus Nord': 54,
        'OnePlus 6T': 51,
        'Mi 11': 230,
        'Mi 10T': 170,
        'Redmi Note 10 Pro': 400,
        'Poco X3 Pro': 321,
        'Mi 9T': 211,
    }

# Define the server socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind((SERVER_ADDRESS, SERVER_PORT))
server_socket.listen(1)

print(f"Server listening on {SERVER_ADDRESS}:{SERVER_PORT}")

while True:
    # Wait for a client to connect
    client_socket, client_address = server_socket.accept()
    print(f"Client connected from {client_address}")

    while True:
        # Receive command from the client
        command = client_socket.recv(1024).decode().strip()

        if not command:
            # If no command is received, close the connection
            client_socket.close()
            print(f"Connection closed for {client_address}")
            break

        # Process the command
        command_parts = command.split()
        command_code = int(command_parts[0])

        if command_code == PROTOCOL["browse"]:
            # Browse the inventory
            response = "\n".join([f"{name},{quantity}" for name, quantity in
list(inventory.items())[:25]])
            client_socket.send(f"{PROTOCOL['response']}\n{response}".encode())

        elif command_code == PROTOCOL["purchase"]:
            # Purchase an item
            item_to_purchase = " ".join(command_parts[1:-1])
            quantity = int(command_parts[-1])
            if item_to_purchase in inventory:
                if inventory[item_to_purchase] >= quantity:
                    inventory[item_to_purchase] -= quantity

```

```

        client_socket.send(f"{PROTOCOL['response']}\nItem
purchased: {item_to_purchase} x
{quantity}\n{inventory[item_to_purchase]}".encode())
    else:
        client_socket.send(f"{PROTOCOL['response']}\nNot enough
quantity of {item_to_purchase} available,will restock soon.Apologize for the
inconvinience.".encode())
    else:
        client_socket.send(f"{PROTOCOL['response']}\nItem
{item_to_purchase} not found".encode())

```

## CLIENT CODE:

```

import socket

# Define the protocol
PROTOCOL = {
    "browse": 1,
    "purchase": 2,
    "response": 3
}

# Define the server address and port
SERVER_ADDRESS = "localhost"
SERVER_PORT = 9999

# Define the client socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect((SERVER_ADDRESS, SERVER_PORT))

def send_command(command):
    # Send command to the server and receive the response
    client_socket.send(command.encode())
    response = client_socket.recv(1024).decode()
    return response.split("\n")[1:]

# Browse the inventory
response = send_command(str(PROTOCOL["browse"]))
items = "\n".join(response).split("\n")
print("Brands available for purchase:")
for item in items:
    name, quantity = item.split(",")
    print(f"{name} ({quantity} available)")

# Purchase an item

```

```
item_to_purchase = input("Enter the brand to purchase:\t")
quantity_to_purchase = int(input("Enter the quantity to purchase:\t"))
command = f"{PROTOCOL['purchase']} {item_to_purchase} {quantity_to_purchase}"
response = send_command(command)
if response[0] == f"Item purchased: {item_to_purchase}":
    print(f"{item_to_purchase} purchased. Available quantity: {response[1]}")
else:
    print(response[0])
```