

بنام خدا

اعضا: سوده زارعی، فاطمه فرهادی، اشکان عظیمی

گزارشکار درباره توضیح نحوه کد

این قسمت از گزارشکار تنها مربوط به خط های کدی است که نوشته شده اند و بیشتر بر روی تغییرات بوجود آمده بر روی انها متمرکز است.

تغییرات بر روی بخش کد سرور:

اولین تغییر که راه دیگری برای پیاده سازی کردن ان نبود، ثبت راحت تر نام و آی پی های کلاینت هاست. بنابر این: در کد، لیستی که شامل آی پی های کلاینت ها در سرور بود، به دیکشنری تبدیل شده و بعد یک دیکشنری دیگر برای نام کلاینت ها نیز در نظر گرفتیم.

```
# -----Our Changes-----
# Changing the list for clients to a dictionary
clients = {}
# Adding another dictionary for their names when needed to address them and their ports
client_names = {}
# -----
```

که چون فقط در کلاینت اسم ها ذخیره میشد، مجبور به تغییری در کلاینت شدیم که اسم را در لحظه ورود، به سرور میدهد.

```
25 else:
26     client.send(name.encode('utf-8')) # Send the user's name to the server
```

سپس در سرور، تنها تغییر دیگری که نیاز بود در سرور ایجاد شود، تشخیص private بودن پیام هاست.

```
# -----Our Changes-----
if message.startswith("PRIVATE"): # changing the chat if it goes for a private
    _, recipient, private_message = message.split(":", 2)
    if recipient in clients:
        clients[recipient].send(f"PRIVATE: {name}: {private_message}".encode())
# -----
```

تغییرت بر روی بخش کد کلاینت:

اولین تغییر که فرستادن اسم هر کلاینت به سرور بود، توضیح داده شد. که این بخش قبل از نشان داده شدن صفحه اصلی چت قرار میگيرد.

```
def receive_message():
    """
    Function to receive messages from the server.
    """
    while True:
        try:
            message = client.recv(1024).decode() # Receive a message from the server
            if not message:
                break
            # -----Our Changes-----
            if message.startswith("PRIVATE"):
                _, sender, private_message = message.split(":", 2)
                if sender in private_windows:
                    private_windows[sender].display_message(f"{sender}: {private_message}")
                else:
                    private_windows[sender] = PrivateChatWindow(root, sender)
                    private_windows[sender].display_message(f"{sender}: {private_message}")
            else:
                message_box.config(state=tk.NORMAL)
                message_box.insert(tk.END, message + '\n') # Display the received message in the message box
                message_box.config(state=tk.DISABLED)
                message_box.yview(tk.END)
            # -----
```

قسمت تغییرات بعدی کد در قسمت دریافت کردن پیام توسط کلاینت است، که اگر private باشد، در چت مخصوص به فرستنده با گیرنده آن نشان داده میشوند.

```
private_windows = {}
#-----Our new function-----
def start_private_chat():
    # Function to create a private chat with another user.
    recipient = simpledialog.askstring("Private Chat", "Enter the name of the user you want to chat with:", parent=root)
    if recipient and recipient != name:
        if recipient not in private_windows:
            private_windows[recipient] = PrivateChatWindow(root, recipient) # Calling the function to open a new window
        else:
            private_windows[recipient].deiconify()

def send_private_message(recipient, message):
    # Function to send a private message to another user.
    if message:
        formatted_message = f"PRIVATE: {recipient}: {message}"
        client.send(formatted_message.encode('utf-8')) # Send the private message to the server
#-----Our Class-----
```

بخش بعدی پنجره های چت های private را شمارش میکند و بعد دو فانکشن بوجود آوردن صفحه چت private جدید و فرستادن پیام های private است.

```
#-----Our Class-----
class PrivateChatWindow(tk.Toplevel):
    # Class to open a new chat window for each new chat between new people.
    def __init__(self, master, recipient):
        super().__init__(master)
        self.recipient = recipient
        self.title(f"Private Chat with {recipient}")

        self.message_frame = tk.Frame(self)
        self.message_frame.pack()

        self.scrollbar = tk.Scrollbar(self.message_frame)
        self.scrollbar.pack(side=tk.RIGHT, fill=tk.Y)

        self.message_box = scrolledtext.ScrolledText(self.message_frame, height=20, width=50, wrap=tk.WORD, yscrollcommand=self.scrollbar.set)
        self.message_box.pack(side=tk.LEFT, fill=tk.BOTH)
        self.message_box.config(state=tk.DISABLED)

        self.message_entry = tk.Entry(self, width=50)
        self.message_entry.pack()
        self.message_entry.focus()

        self.message_entry.bind("<Return>", self.send_private_message)

    def send_private_message(self, event=None):
        """
        Function to send a private message.
        """
        message = self.message_entry.get()
        if message:
            send_private_message(self.recipient, message)
            self.display_message(f"You: {message}")
            self.message_entry.delete(0, tk.END)

    def display_message(self, message):
        """
        Function to display a message in the private chat window.
        """
        self.message_box.config(state=tk.NORMAL)
        self.message_box.insert(tk.END, message + '\n')
        self.message_box.config(state=tk.DISABLED)
        self.message_box.yview(tk.END)
#-----
```

تغییر بعدی، ارائه کلاس صفحه چت private با فانکشن های فرستادن پیام و مشاهده پیام ها است.

```
# -----Our Buttons -----  
private_chat_button = tk.Button(root, text="Start Private Chat", command=start_private_chat)  
private_chat_button.pack()  
# -----
```

تغییر آخر برای طراحی دکمه مربوط به ساخت صفحه چت private است.