

## پروژه چت تحت شبکه با Python و Tkinter

### مقدمه

این پروژه یک چت باکس ساده برای ارتباط بین دو کاربر در یک شبکه محلی (LAN) است. برنامه از **socket** برای ارتباط شبکه‌ای و از **tkinter** برای رابط گرافیکی کاربر (GUI) استفاده می‌کند.

### ساختار پروژه

این پروژه شامل دو بخش اصلی است:

۱. **سرور – (server.py)** مسئول مدیریت ارتباطات و ارسال پیام بین کلاینت‌ها
۲. **کلاینت – (client.py)** رابط گرافیکی که کاربران می‌توانند پیام ارسال و دریافت کنند

### نحوه اجرا

#### ۱. اجرای سرور

روی سیستمی که قرار است سرور باشد، فایل **Server.py** را اجرا کنید:

پس از اجرا، سرور منتظر اتصال کلاینت‌ها خواهد بود.

#### ۲. تنظیم آی‌پی کلاینت

در سیستم کاربر دوم، مقدار **SERVER\_IP** در فایل **ChatBox.py** باید برابر با آی‌پی سیستم سرور تنظیم شود:

آی‌پی سیستم سرور را اینجا قرار دهید:

```
SERVER_IP = "192.168.xxx.xxx"
```

```
SERVER_PORT = 12345
```

برای پیدا کردن آدرس سرور در CMD سیستمی که قرار است سرور اصلی باشد عبارت **ipconfig** وارد کنید و مقدار **IPv4** را قرائت کنید.

بعد از ذخیره تغییرات، فایل **ChatBox.py** را اجرا کنید:

کلاینت باید بدون مشکل به سرور متصل شود.

### ویژگی‌ها و عملکرد

- ارسال و دریافت پیام بین دو سیستم در شبکه محلی

- نمایش مداوم پیام‌ها در چت‌باکس
- استفاده از **socket** برای ارتباط بین سرور و کلاینت
- رابط کاربری با **tkinter** برای نمایش پیام‌ها

توضیح مختصر از کد

### کد سرور (Server.py)

در کد سرور، از **socket** برای ایجاد یک اتصال شبکه‌ای استفاده شده است. سرور منتظر اتصال از طرف کلاینت‌ها می‌ماند و پس از آن پیام‌ها را دریافت کرده و به تمامی کلاینت‌های متصل ارسال می‌کند. این کد از پورت 12345 برای ارتباط استفاده می‌کند.

### کد کلاینت (ChatBox.py)

در کد کلاینت، رابط گرافیکی با **tkinter** برای ارسال و دریافت پیام‌ها ایجاد شده است. پیام‌ها به سرور ارسال شده و از آن دریافت می‌شوند تا در پنجره چت نمایش داده شوند. کلاینت به آدرس IP سرور متصل می‌شود و از همان پورت برای ارتباط استفاده می‌کند.

در این کد، از دو رشته (thread) استفاده شده است:

- یکی برای دریافت پیام‌ها از سرور و نمایش آن‌ها در **chat box**.
- دیگری برای ارسال پیام از ورودی متنی (text entry) به سرور.

### نتیجه‌گیری

این پروژه یک پیاده‌سازی ساده از چت تحت شبکه است که می‌توان آن را گسترش داد. امکان اضافه کردن قابلیت‌هایی مانند پشتیبانی از چندین کاربر، رمزنگاری پیام‌ها، و ارسال فایل‌ها وجود دارد.