

1. مورد اول این است که نام کاربری و رمز عبور باید بیش از 2 کاراکتر داشته باشند.

Client

- از کلاینت شروع میکنیم :

A. Registerme

این تابع اولین تابع کلاینت است که با سرور در ارتباط مستقیم است. با گرفتن نام کاربری و رمز عبور از کاربر انرا به شکل خاصی از رشته به سمت سرور میفرستد. اگر سرور اطلاعات را پذیرفت که برنامه با موفقیت وارد منوی بعدی میشود و در غیر اینصورت به منوی قبلی برمیگردد. (اعداد کد گذاری شده عامل تعیین منوی پیشروی کاربر هستند که در آخر تشریح میشوند).

B. Login

در این تابع نیز مثل تابع قبل با گرفتن نام کاربری و رمز عبور و فرستادن آنها برای سرور کار را پیش میبرد کارش بسیار نزدیک به تابع قبل است با این تفاوت که در این تابع رشته ای تحت عنوان token ساخته میشود که کد اعتباری هر کاربر آنلاین است در این تابع به کاربر تحویل داده میشود. البته ساخت آن توسط سرور شکل میگیرد. توابعی نظیر strcat برای ساخت رشته مورد تایید سرور هستند.

C. Createchannel

این تابع در منویی جدا از دو مورد اول وجود دارد. تابع برای تشکیل کانال است. نامی برای کانال و توکن کاربر را از وی گرفته و مورد بررسی قرار میدهد اگر مشکلی نبود پیام موفقیت را مثل توابع قبلی برمیگرداند که برنامه انرا تجزیه کرده و نتیجه را بما برمیگرداند. این تابع بطور خودکار ایجاد کننده کانال را در خود کانال عضو میکند (این کار متناظر سرور در هنگام اجرای این تابع است). در صورت موفقیت این تابع کاربر به منوی گفت و گو میرود و در غیر اینصورت در همان منو میماند.

D. Sendmessage

این تابع نیز پیام و توکن را از کاربر میگیرد و سپس با فرمت خاص به سرور میفرستد. بقیه فرایند این تابع دقیقا مانند توابع قبلی است.

E. Refresh

این تابع برای نشان دادن پیام هایی در کانال است که از آخرین بازدید شما در کانال مورد نظر ثبت شده است. در این تابع توکن تنهای داده ای است که کاربر انرا به سرور میدهد (البته عبارت refresh هم هست که جزء داده های کلاینت محسوب نمیشود). توی این تابع تقریبا سخت ترین کار و فرایند این پروژه انجام میگیرد که در آن پس از پارس کردن رشته جواب سرور باید لیست پیام ها و فرستنده پیام ها را نیز از رشته ای دیگر که درون رشته اولیه است بیرون بکشیم.

F. Logout

اما برگردیم به یک منو قبل تر از منویی که تابع قبلی در آن حضور داشت. تابع لاگ اوت هم به عنوان ورودی مثل تمام توابعی که در بالا ذکر شدند توکن تولید شده را میگیرد. در داخل تابع انرا به عبارت لاگ اوت میچسباند و برای سرور میفرستد و سپس جواب سرور را تفکیک میکند.

G. Leave channel

ورودی هم چون تمام توابع کلاینت که با سرور ارتباط مستقیم دارند تنها عبارت توکن تولید شده برای کلاینت است. انرا به عبارت leave میچسباند و سپس برای سرور ارسال میکند و در ادامه مثل بقیه توابع عمل میکند.

H. Member

در واقع همان channel list است. تابعی که باز هم توکن را ورودی میگیرد و سپس به عبارت channel members میچسباند و دوباره همان روند رفرش را در پی میگیرد با کمی تفاوت زیرا در این قسمت نباید ارایه از رشته ها را از جیسون دریافت شده جداسازی کرد. البته یک مقدار ساده تر میباشد تجزیه این جیسون.

- حال توابعی که یا با کاربر کاری ندارند یا با سرور

I. Create socket

این تابع در واقع اصلی ترین تابع بخش کلاینت میباشد. اتصال به سرور توسط این تابع شکل میگیرد و پس از هر رد و بدل شدن داده بین کلاینت و سرور سوکت کلاینت توسط close socket بسته میشود پس هر بار نیاز است که این تابع را صدا بزنیم.

J. Main

در این تابع تمام اتفاقات کلاینت انجام میگیرد. در ابتدا سوکت ساخته میشود و به سرور اتصال پیدا میکند و پس از آن هم یک لوپ بینهایت که تنها با تایپ `yes or Yes` از لوپ خارج میشود و در آن لوپ تمام توابع و در هر دور از آن هم یک سازنده سوکت پیدا میشود.

.K `Menu_conver`

شامل گزینه هایی از این قبیل میباشد : `create a channel – join a channel – log out` که با انتخاب هر کدام از گزینه ها در تابع `number` به کد آن تابع هم اسم گزینه تغییر میکند و با `return` به درون لوپ تابع اصلی برمیگردد و از طریق کد دوباره تابع مورد نظر را پیدا میکند.

.L `Asl_menu`

`Register,log in` و سوال اینکه آیا میخواهید در برنامه بمانید یا نه در این تابع به کلاینت نشان داده میشود و بقیه کار آن همانند تابع قبلی است.

.M `Talk_menu`

این تابع شامل این موارد است : `send message – refresh – channel members – leave channel` نحوه کار این تابع نیز همانند دو تابع قبلی است.

- فرم ارسال درخواست برای سرور از طرف کلاینت

"register <username>, <password>"

"login <username>, <password>"

"create channel <channel name>, <AuthToken>"

"join channel <channel name>, <AuthToken>"

"logout <AuthToken>"

"send <message>, <AuthToken>"

"refresh <AuthToken>"

"leave <AuthToken>"

- راهنمای اعداد کد گذاری شده در برنامه تحت عنوان `number` برای رفتن به تابع های مورد نظر کاربر

```
if(number==0)
```

```
    aslMenu(AuthToken);
```

```
if(number==1)
```

```
    registerMe(AuthToken);
```

```
if(number==2)
```

```
    login(AuthToken);
```

```
if(number==3)
```

```
    menu_conver(AuthToken);
```

```
if(number==4)
```

```
    sendMessage(AuthToken);
```

```
if(number==5)
```

```
    talk_menu(AuthToken);
```

```
if(number==6)
```

```
    join_channel(AuthToken);
```

```

if(number==7)
    refresh(AuthToken);

if(number==8){
    member(AuthToken);
}

if(number==9)
    leave_channel(AuthToken);

if(number == 10)
    logout(AuthToken);

if(number == 11)
    create_channel(AuthToken);

```

## 2. Server

- حال میرسیم به سنگین ترین بخش کار یعنی سرور در این بخش یعنی سرور من سرور دوم را برای فاز ۲ طراحی کردم و بطور مستقیم از سرور اول که برای فاز ۳ بود بهره بردم اما با این حال تمام کد های سرور فاز ۳ رو در تابع اصلی زدم و از تابع دیگری استفاده نکردم پس در این بخش به تشریح توابع سرور فاز ۲ میپردازم.

### (a) Registering

از نام این تابع هم پیدا است که هنگام ارسال درخواست رجیستر توسط کلاینت به میدان می آید. رشته حاوی درخواست های کاربر را تحت عنوان بافر به عنوان ورودی میگیرد. دیگر ورودی این تابع پوینتری است که به فایلی اشاره میکند که در تابع اصلی کاربرد آن فایل توضیح داده میشود. این تابع نام و رمز عبور را از رشته فرستاده شده توسط کاربر بدست می آورد و پس از آن در فایل ممبر چک میکند اگر مورد مشابهی بود پیام ارور را به فرمتی خاص برای کلاینت میفرستد اگر نه نیز نام و رمز را در فایل وارد میکند و پیام موفقیت میفرستد.

### (b) Login

این تابع نیز بافر و فایل ممبر را به عنوان ورودی میگیرد. ورودی دیگر این تابع فایلی با نام آنلاین است. این تابع نیز نام و رمز را مانند تابع قبل بدست می آورد و سپس پس از چک کردن ممبر آنرا به آنلاین اضافه میکند.

### (c) Create channel

ورودی های این تابع بافر و آنلاین و چنل لیست و چنل و مسیج اند که همه بجز بافر پوینتر به فایل اند. درخواست از کاربر را میگیرد و در صورت نبودن کانالی همنام پیام موفقیت آمیز و در غیر اینصورت پیام خطا باز میگرداند.

### (d) Listchannel

بافر، آنلاین و چنل ورودی ها هستند که بافر رشته است. این تابع با بررسی کانال اعلام شده تمام افراد آنلاینی را که در این کانال هستند را شناسایی میکند و با فرمت خاصی برای کلاینت میفرستد.

### (e) Joinchannel

این تابع نام کانال مورد نظر را از رشته جواب کلاینت استخراج میکند پس در فایل آنلاین بدنبال نام کاربر میگردد و سپس نام را به همراه نام کانال به فایل کانال اضافه میکند. البته که تابع چک میکند که نام استخراجی کانال در بین کانال های موجود باشد. در آخر پیامی در فایل مسیج نیز پیامی برای ثبت عضو شدن کاربر ثبت میشود.

#### Leave (f)

در این تابع از طریق توکن (از رشته فرستاده شده توسط کلاینت استخراج میشود.) در فایل آنلاین میتوان نام کاربری که این درخواست را داده بدست آورد و پس از آن اطلاعات مربوط به آن نام در فایل کانال را پاک میکنیم.

#### Log out (g)

این تابع نیز تا حد زیادی همانند تابع قبل عمل میکند با این تفاوت که اطلاعات را در فایل آنلاین حذف میکند.

#### Sending (h)

این تابع از طریق توکن کلاینت نام آنرا از فایل آنلاین پیدا میکند و سپس نام فرستنده پیام را با فرمتی خاص به پیام میچسباند و پس از آن از طریق نام کاربر و فایل کانال نام کانالی که کاربر در آن پیام گذاشته نیز یافت میشود.

#### Getting data (i)

صرفاً سرور را برای دریافت پیامی از کلاینت آماده میکند تا سوکت را متصل کند.

- بخش های دیگر سرور با نامگذاری های معنا دار و کامنت گذاری شرح شده اند.