به نام خدا

توضیحاتی در خصوص توابع استفاده شده در فایل سرور

دسته اول) این دسته شامل توابعی می‌شود که مستقیماً پیام را به کلاینت ارسال نمی‌کنند.

تابع چت:

```
void chat ();
```

این تابع بعد از اتصال به کلاینت صدا زده می‌شود و ورودی و خروجی ندارد.

این تابع هر بار یک درخواست از کلاینت دریافت می‌کند و با توجه به آن تابع مناسب را صدا می‌زند. برای مثال اگر درخواست کلاینت افزودن یک نفر به کانال باشد، تابع fun_join را صدا می‌زند. یعنی کارکرد این تابع دریافت یک پیام از کلاینت و هدایت پیام به تابع مناسب است. (دقت کنید که رشته دریافتی global تعریف شده و نیازی به ارسال آن به عنوان ورودی به تابع نیست.)

تابع تکرار:

```
int fun_tekrar ( int n , char user[]);
```

این تابع برای بررسی موجود بودن یا نبودن یک نام کاربری یا نام کانال استفاده می‌شود. به این صورت که برای بررسی نام کاربری عدد ۱ به عنوان مولفه اول و نام کاربری مورد بررسی به عنوان مولفه دوم به تابع داده می‌شود و این تابع همه‌ی فایل ها را در دایرکتوری مورد نظر چک می‌کند. در صورت موجود بودن عدد صفر و در غیر این صورت عدد ۱ را به عنوان خروجی برگشت می‌دهد. برای بررسی اسم کانال، باید عدد ۲ را به عنوان مولفه اول و اسم کانال را به عنوان مولفه دوم به تابع ارسال کنیم. (توجه کنید که دلیل این نوع چک کردن، نام گذاری فایل‌های حاوی اطلاعات کاربران با همان نام کاربران است. همچنین فایل های حاوی اطلاعات کانال نیز با اسم همان کانال ذخیره می‌شوند.)

تابع بررسی کاربر آنلاین:

```
int  fun_user ( cJSON *a , char user[] , int n);
```

این تابع دو کاربرد دارد. یکی برای بررسی آنلاین بودن یک کاربر و دیگری برای بدست آوردن نام کاربری و کانال یک کاربر آنلاین. به عنوان مولفه اول باید جیسونی که حاوی اطلاعات کاربران آنلاین است را به تابع ارسال کنیم.

حال اگر بخواهیم آنلاین بودن یک نام کاربری را متوجه شویم باید به عنوان مولفه دوم آن نام کاربری و به عنوان مولفه سوم عدد ۱ را وارد کنیم. در صورت آنلاین بودن فرد (با جستجو در کاربران آنلاین) عدد صفر و در غیر اینصورت عدد یک برگشت داده می‌شود. برای بدست آوردن اطلاعات یک کاربر آنلاین با توجه به authtoken باید در مولفه دوم آن توکن و در مولفه سوم عدد ۲ وارد شود. در صورت پیدا نشدن کاربر عدد ۱ برگشت داده می‌شود. در غیر اینصورت اطلاعات آن کاربر وارد تعدادی متغیر سراسری می‌شود تا بعداً بتوانیم از آنها استفاده کنیم. همچنین عدد صفر برگشت داده می‌شود.

تابع ساخت توکن:

```
void randauth ();
```

هنگامی که کاربری لاگین می‌کند، در صورت موفقیت، سرور باید یک توکن برای آن ایجاد کند تا در دفعات بعد از آن استفاده کند. این کار را با استفاده از این تابع انجام می‌دهیم. به اینصورت که با استفاده از تابع srand و خود تابع rand ۳۰ کاراکتر تصادفی از اعداد و حروف کوچک انگلیسی ایجاد می‌کنیم و آن را در متغیر سراسری auth ذخیره می‌کنیم.

دسته دوم) هرکدام از توابعی که در ادامه توضیح آن‌ها خواهد آمد مربوط به یکی از درخواست های کلاینت از سرور می‌باشد. یعنی برای هر درخواست کلاینت از سرور یک تابع جداگانه تعریف کرده‌ایم. این توابع بعد از انجام یک سری عملیات پاسخ را به کلاینت ارسال می‌کنند.

تابع ثبت نام:

```
void fun_register ();
```

اگر کلاینت درخواست ثبت نام شخصی را به سرور ارسال کند، تابع چت این تابع را صدا می‌زند. این تابع با بررسی نام کاربری و رمز عبور که آن ها را از رشته دریافتی بدست می‌آورد، تکراری بودن آن را بررسی می کند و درصورت عدم تکرار کاربری با این مشخصات ایجاد می‌کند و پیام موفقیت را به کلاینت می‌فرستد. در صورت تکراری بودن نام کاربری هم پیام خطای متناسب با آن را ارسال می کند.

تابع لاگین (ورود):

void fun_login ();

اگر کلاینت درخواست لاگین شخصی را به سرور ارسال کند، تابع چت این تابع را صدا می‌زند. این تابع با بررسی نام کاربری و رمز عبور که آن ها را از رشته دریافتی بدست می‌آورد، تکراری بودن آن را بررسی می کند و درصورت عدم وجود نام کاربری، پیام خطای متناسب با آن را ارسال می کند. در صورت آنلاین بودن کاربر نیز پیام خطا ارسال می‌کند. در غیر اینصورت رمز عبور وارد شده را با رمز عبور ذخیره شده مقایسه می‌کند و در صورت تطبیق یک توکن ایجاد می‌کند و آن را برای کلاینت ارسال می‌کند و کاربر را به کاربران آنلاین اضافه می‌کند.

تابع logout (خروج):

void fun_logout ();

اگر کلاینت درخواست خروج یک شخص را به سرور ارسال کند، تابع چت این تابع را صدا می‌زند. این تابع ابتدا توکن را بررسی می‌کند در صورت آنلاین نبودن آن پیغام خطا را برای کلاینت ارسال می‌کند. در غیر اینصورت کاربر را از بین کاربران آنلاین حذف می‌کند و پیغام موفقیت را برای کلاینت می فرستد.

تابع ساخت کانال:

void fun_createch ();

اگر کلاینت درخواست ساخت یک کانال را برای سرور بفرستد، تابع چت این تابع را صدا می‌زند. این تابع ابتدا نام کانال را از رشته دریافتی متوجه می‌شود. سپس به کمک تابع "بررسی کاربر آنلاین" نام کاربری و درست بودن توکن را متوجه می‌شود. سپس در صورت عدم تکراری بودن نام کانال، کانالی با این نام ایجاد می‌کند و کاربر را به عضویت آن در می‌آورد. سپس اطلاعات کانال را در فایلی ذخیره می‌کند.

تابع ورود به کانال:

void fun_join ();

اگر کلاینت درخواست عضویت در یک کانال را برای سرور بفرستد، تابع چت این تابع را صدا می‌زند. این تابع ابتدا نام کانال را از رشته دریافتی متوجه می‌شود. سپس به کمک تابع "بررسی کاربر آنلاین" نام کاربری و درست بودن توکن را متوجه می‌شود. سپس در صورت عدم وجود کانال، پیغام خطا را به کلاینت ارسال می‌کند. درغیر اینصورت کاربر را به عضویت آن کانال در می‌آورد. سپس اطلاعات کانال را بروز می‌کند.

تابع خروج از کانال:

void fun_leave ();

اگر کلاینت درخواست خروج از یک کانال را برای سرور بفرستد، تابع چت این تابع را صدا می‌زند. ابتدا به کمک تابع "بررسی کاربر آنلاین" نام کانال کاربر و درست بودن توکن را متوجه می‌شود. اگر توکن اشتباه باشد یا کاربر در کانالی عضو نباشد پیغام خطای نظیر آن را برای کلاینت ارسال می‌کند. درغیر اینصورت کاربر را از آن کانال خارج می‌کند و پیغام موفقیت را برای کلاینت می‌فرستد.

تابع ارسال پیام:

void fun_send ();

اگر کلاینت درخواست ارسال پیام را برای سرور بفرستد، تابع چت این تابع را صدا می‌زند. این تابع ابتدا توکن و پیام را از رشته دریافتی بدست می‌آورد. سپس با استفاده از تابع "بررسی کاربر آنلاین" نام کاربری و اسم کانال این کاربر را بدست می‌آورد. اگر توکن اشتباه باشد یا کاربر در کانالی عضو نباشد پیغام خطای مربوطه را برای کلاینت ارسال می‌کند. در غیر اینصورت این پیام را به لیست پیام های آن کانال اضافه می‌کند و پیغام اجرای موفق را برای کلاینت می‌فرستد.

تابع مشاهده پیام:

void fun_refresh ();

اگر کلاینت درخواست مشاهده پیام ها را برای سرور ارسال کند، تابع چت این تابع را صدا می‌زند. ابتدا این تابع توکن را از رشته دریافتی استخراج می‌کند و سپس با استفاده از تابع "بررسی کاربر آنلاین" درست بودن توکن، نام کاربری، اسم کانال و تعداد پیام های دیده شده توسط کاربر را بدست می‌آورد. اگر توکن اشتباه باشد یا کاربر در هیچ کانالی نباشد پیغام خطای مناسب برای کلاینت ارسال می‌شود. در غیر اینصورت پیام های دیده نشده توسط کاربر از این کانال را در لیستی قرار می دهد و برای کلاینت می‌فرستد.

تابع جستجو در پیام ها:

void fun_search ();

اگر کلاینت درخواست جستجو در پیام ها را برای سرور ارسال کند، تابع چت این تابع را صدا می‌زند. این تابع شبیه تابع قبل عمل می‌کند با این تفاوت که یک کلید واژه را از رشته دریافتی استخراج می‌کند و هر پیامی را که دارای آن بود، وارد آرایه می‌کند. در این جا از تابع strstr استفاده کردیم که یک رشته را در رشته دیگر پیدا می‌کند.

تابع اعضای کانال:

```
void fun_members ();
```

اگر کلاینت درخواست مشاهده اعضای کانال را برای سرور ارسال کند، تابع چت این تابع را صدا می‌زند. این تابع مانند تابع "مشاهده پیام ها" از تابع "بررسی کاربر آنلاین" برای پیدا کردن کانال کاربر استفاده می‌کند. باز هم مثل حالات قبل اگر توکن اشتباه باشد یا کاربر در کانالی نباشد پیغام خطای مربوطه را برای کلاینت می‌فرستد. در غیر اینصورت لیست اعضای کانال را از فایل مربوط به آن کانال استخراج می‌کند و برای کلاینت می‌فرستد.

تابع جستجو در اعضای کانال:

```
void fun_find ();
```

اگر کلاینت درخواست جستجو در اعضای کانال را برای سرور ارسال کند، تابع چت این تابع را صدا می‌زند. ابتدا این تابع از رشته دریافتی نام کاربری مورد جستجو و توکن را دریافت می‌کند. سپس این تابع مانند تابع "اعضای کانال" عمل می‌کند با این تفاوت که اگر در بین اعضای کانال آن کاربر را یافت پاسخ مثبت برای کلاینت می‌فرستد و در غیر اینصورت پاسخ منفی برای کلاینت می‌فرستد.