

توابع مهم مورد استفاده :

• `int main()` :

در تابع `main` ابتدا تابع `srand(time(o))` اجرا می‌شود که در ادامه برای تولید `AuthToken` استفاده می‌شود. سپس تابع `readFiles()`، `makeServerSocket()` و `recieveRequest()` اجرا می‌شوند که در ادامه نحوه عملکرد این توابع توضیح داده خواهد شد (تابع `recieveRequest()` در یک `loop` بینهایت قرار گرفته است).

• `void readFiles()` :

در این تابع اطلاعات مربوط به کاربران و کانال‌ها از فولدر `Resources` خوانده می‌شود و در آرایه‌های مربوط به نام کاربران و کانال‌ها اسم کاربران و کانال‌ها ذخیره و سپس چاپ می‌شود. در صورتی که هر یک از فولدرهای `Resources`، `Channels` یا `Users` وجود نداشت، این فولدرها ایجاد می‌شوند.

• `bool fileExists(DIR dr*, char filename[])` :

این یک تابع کمکی است که بررسی می‌کند آیا در دایرکتوری `dr` فایلی به اسم `filename[]` وجود دارد یا نه. در صورت وجود `true` و در غیر این صورت `false` برگردانده می‌شود.

• `void makeServerSocket()` :

در این تابع سوکت مربوط به `Server` ساخته می‌شود، متغیرهای گلوبال `int server_socket` و `struct sockaddr_in server` مقداردهی می‌شوند و سرور به ترتیب در وضعیت‌های `bound` و `listening` قرار می‌گیرد و آماده دریافت پیام از `Client` می‌شود.

• `void recieveRequest()` :

در این تابع ابتدا سوکت مربوط به `Client` با تابع `makeSocket()` ساخته می‌شود و سپس با استفاده از تابع `recv()` رشته‌ای ارسال شده توسط `Client` دریافت می‌شود نوع درخواست تعیین می‌شود و به درخواست مربوطه در توابع مشخصی پاسخ داده می‌شود. سپس سوکت `Client` بسته می‌شود.

- void makeSocket() :

برای ساختن سوکت مربوط به Client از این تابع استفاده می‌شود. متغیر های گلوبال `int client_socket` و `struct sockaddr_in client` در این تابع مقداردهی می‌شوند و درخواست `client` قبول می‌شود.

- int reg(char buffer[]) :

ورودی پیام ارسالی از طرف Client است که در `username` و `password` ابتدا از آن استخراج می‌شود. با کمک `userExists()` وجود کاربری با `username` وارد شده بررسی می‌شود. در صورتی که `username` از قبل موجود بود، خطای مربوطه ارسال می‌شود و مقدار `REG_ALREADY_EXISTS` برگردانده می‌شود. در غیر این صورت کاربر مورد نظر با استفاده از تابع `createUser()` ایجاد می‌شود، پیام موفقیت آمیز بودن ارسال می‌شود و مقدار `REG_SUCCESSFUL` برگردانده می‌شود.

- bool userExists(char username[]) :

در این تابع `username` در بین آرایه `username` ها جست و جو می‌شود. در صورتی که وجود داشته باشد `true` و در غیر این صورت `false` برگردانده می‌شود.

- void createUser(char username[], char password[]) :

در این تابع کاربری با `username` و `password` مورد نظر با فرمت `cJSON` در فولدر `Users` ایجاد می‌شود و همچنین به آرایه کاربران اضافه می‌شود.

- int login(char buffer[]) :

ابتدا از `buffer`، `username` و `password` استخراج می‌شود. سپس وجود کاربری با این نام بررسی می‌شود. در صورت عدم وجود خطای مربوطه ارسال می‌شود و مقدار `LOGIN_NOT_EXISTS`، `return` میشود. در غیر این صورت وضعیت `online` بودن کاربر در آرایه `loggedin` بررسی می‌شود. اگر کاربر در حال حاضر آنلاین بود، خطای مربوطه ارسال می‌شود و مقدار `LOGIN_ALREADY_LOGGEDIN` برگردانده می‌شود. اگر رمز ورودی اشتباه بود خطای مربوطه ارسال می‌شود و مقدار `LOGIN_WRONG_PASSWORD` برگردانده می‌شود. در غیر این صورت `AuthToken` برای کاربر ارسال می‌شود، وضعیتش آنلاین می‌شود و `LOGIN_SUCCESSFUL` برگردانده می‌شود.

- `int userindex(char username[]) :`

اندیس کاربر با نام `username` را در بین آرایه کاربران می‌یابد و آن را `return` می‌کند.

- `int userindex_auth(char auth[]) :`

اندیس کاربر با `AuthToken` ورودی در بین آرایه توکن‌ها جست و جو می‌شود و `return` می‌شود.

- `bool checkPass(char username[], char password[]) :`

کاربر با `username` مورد نظر در فولدر `Users` جست و جو می‌شود. در صورتی که رمز ورودی با رمز کاربر مورد نظر یکسان بود، `true` و در صورتی که یکسان نبود `false` برگردانده می‌شود.

- `bool makeAuth(int index) :`

یک `AuthToken` تصادفی به طول ۳۲ حرف ساخته می‌شود (حرف‌های آن شامل حروف کوچک و بزرگ انگلیسی، - و _ می‌باشد). در صورتی که `AuthToken` مشابهی در آرایه توکن‌ها وجود داشت، `false` برگردانده می‌شود و در غیر این صورت `AuthToken` ساخته شده در اندیس `index` از آرایه توکن‌ها کپی شده و `true` برگردانده می‌شود.

- `int create_channel(char buffer[]) :`

نام کانال و `AuthToken` ابتدا استخراج می‌شود. در صورتی که `AuthToken` درست نبود، `AUTH_INVALID` برگردانده می‌شود. وجود کانال با `channelExists` بررسی می‌شود. در صورتی که کانال مورد نظر وجود داشت خطای مربوطه ارسال می‌شود و مقدار `CRCH_ALREADY_EXISTS` برگردانده می‌شود. در غیر این صورت کانال مورد نظر و اطلاعات مربوط به آن با فرمت `cJSON` در فولدر `Channels` ایجاد می‌شود، وضعیت کاربر در آرایه `inchannel` در اندیس‌های مربوطه `true` میشود، پیام ساخته شدن کانال در کانال ارسال می‌شود، پیام موفقیت آمیز بودن برای کاربر ارسال می‌شود و مقدار `CRCH_SUCCESSFUL` برگردانده می‌شود.

- `bool channelExists(char channel[]) :`

در صورتی که در آرایه کانال‌ها، کانالی با نام ورودی وجود داشت `true` و در غیر این صورت `false` برگردانده می‌شود.

• `int channelindex(char channel[]) :`

اندیس کانال با نام ورودی در آرایه کانال ها یافته و برگردانده می شود.

• `bool authValidity(char auth[]) :`

توکن ورودی در آرایه توکن ها جست و جو می شود. در صورت وجود توکن `true` و در غیر این صورت `false` برگردانده می شود.

• `int join_channel(char buffer[]) :`

ابتدا نام کانال و `AuthToken` استخراج می شود. در صورت غلط بودن `AuthToken` مقدار `AUTH_INVALID` برگردانده می شود. در صورتی که کانال مورد نظر وجود نداشت، خطای مربوطه ارسال می شود و مقدار `JOCH_NOT_EXISTS` برگردانده می شود. در غیر این صورت کاربر مورد نظر در کانال عضو شده، وضعیت کاربر در آرایه `inchannel` در اندیس های مربوطه `true` میشود، پیام ورود به کانال در کانال ارسال می شود، پیام موفقیت آمیز بودن برای کاربر ارسال می شود و `JOCH_SUCCESSFUL` برگردانده می شود.

• `int logout(char buffer[]) :`

ابتدا `AuthToken` استخراج شده و صحت آن بررسی می شود. در صورت اشتباه بودن آن `AUTH_INVALID` برگردانده می شود. در غیر این صورت وضعیت آنلاین بودن او در آرایه `loggedin` بررسی می شود. اگر `false` بود، `LOGOUT_FAILED` برگردانده می شود و اگر `true` بود، `false` می شود، پیام موفقیت آمیز بودن ارسال می شود و `LOGOUT_SUCCESSFUL` برگردانده می شود.

• `int send_msg(char buffer[]) :`

ابتدا `AuthToken` و پیام موردنظر استخراج می شود. اگر توکن غلط بود `AUTH_INVALID` برگردانده می شود. در صورتی که کاربر در کانالی عضو نبود، خطای مربوطه ارسال می شود و `MSG_NOT_IN_CHANNEL` برگردانده می شود. در غیر این صورت پیام موردنظر در کانال مربوطه ارسال می شود. به این صورت که کل فایل `cJSON` مربوط به کانال خوانده می شود، پیام در غالب `cJSON` اضافه می شود و فایل `cJSON` قدیمی با فایل جدید `overwrite` می شود. سپس پیام موفقیت آمیز بود برای کاربر ارسال می شود و `MSG_SUCCESSFUL` برگردانده می شود.

• `int channelindex_auth(char auth[]) :`

اندیس کانالی که کاربر با توکن `auth` در آن عضو است را برمی گرداند. در صورتی که کاربر در کانالی عضو نبود ۱- برگردانده می شود.

• `int refresh(char buffer[]) :`

ابتدا `AuthToken` استخراج شده و صحت آن بررسی می شود. در صورتی که درست نبود `AUTH_INVALID` برگردانده می شود. در صورتی که کاربر در کانالی عضو نبود، پیام خطای مربوطه ارسال می شود و `RE_NOT_IN_CHANNEL` برگردانده می شود. در غیر این صورت `refresh` انجام می شود. به این صورت که اطلاعات کانال کاربر از فایل ها خوانده می شود و کل پیام ها کانال و تعداد آنها (`size`) استخراج می شود. سپس با استفاده از آرایه `msgindex` آخرین پیام خوانده شده توسط کاربر، در کانال مورد نظر مشخص می شود و از آن پیام تا آخرین پیام ارسال شده به صورت لیستی برای کاربر ارسال می شود و مقدار `RE_SUCCESSFUL` برگردانده می شود.

• `int channel_members(char buffer[]) :`

ابتدا `AuthToken` استخراج شده و صحت آن بررسی می شود. در صورتی که درست نبود `AUTH_INVALID` برگردانده می شود. در صورتی که کاربر در کانالی عضو نبود، پیام خطای مربوطه ارسال می شود و `CHMEM_NOT_IN_CHANNEL` برگردانده می شود. در غیر این صورت در آرایه `inchannel` در اندیس مربوط به کانال کنونی فرد، افراد موجود جست و جو شده و نام این افراد به صورت لیست برای کاربر ارسال شده و `CHMEM_SUCCESSFUL` برگردانده می شود.

• `int leave(char buffer[]) :`

ابتدا `AuthToken` استخراج شده و صحت آن بررسی می شود. در صورتی که درست نبود `AUTH_INVALID` برگردانده می شود. در صورتی که کاربر در کانالی عضو نبود، پیام خطای مربوطه ارسال می شود و `LEAVE_NOT_IN_CHANNEL` برگردانده می شود. در غیر این صورت پیام خروج فرد از کانال در کانال ارسال شده، وضعیت در کانال بودن او `false` شده، پیام موفقیت آمیز بودن برای کاربر ارسال می شود و `LEAVE_SUCCESSFUL` برگردانده می شود.

• `char *concat(int count, ...)` :

یک تابع کمکی که تعداد دلخواهی `string` را به هم می‌چسباند و آدرس `string` نهایی را برمی‌گرداند.

