بسم الله الرحمن الرحیم

محمد حسین حاجی سید سلیمان / گروه سه

فاز اول: Client

در ابتدا باید اشاره کنم که برنامه در یک حلقه بی‌نهایت قرار دارد و در این حلقه بر اساس متغیر status توابع مربوطه صدا زده می‌شود.

- تابع color

خروجی ندارد و در ورودی یک عدد دریافت می‌کند. بر اساس آن عدد رنگ نوشته‌ها پس از صدا زدن این تابع تغییر می‌کند.

- تابع menu1

این تابع نیز ورودی و خروجی ندارد. در ابتدا فهرست ابتدایی را به کاربر نمایش می‌دهد که شامل Register و Login است. بر اساس انتخاب کاربر، کد مربوط به هر قسمت اجرا می‌شود و در ادامه پاسخ مناسب به server فرستاده می‌شود. سپس پاسخ سرور دریافت شده و تحلیل می‌شود. اگر Register موفقیت آمیز بود، همین منو دوباره نمایش داده می‌شود تا کاربر Login کند. اگر Login موفقیت آمیز بود کاربر به منوی بعدی هدایت می‌شود.

- تابع menu2

تابع menu2 بدون ورودی و خروجی است. این منو پس از ورود موفقیت آمیز کاربر نمایش داده می‌شود. مشابه بخش قبل بر اساس انتخاب کاربر، هر یک کدهای مربوطه به این منو اجرا می‌شود. کاربر با انتخاب Create channel امکان ساخت کانال، با انتخاب Join channel امکان عضویت در کانال‌های موجود و با انتخاب Logout می‌تواند از حساب خود خارج شود. همچنین کاربر می‌تواند با انتخاب My current channel ببیند که آیا در کانالی عضو هست یا خیر و اگر عضو بود به آن کانال هدایت شود.

در هر یک از بخش‌های فوق، در صورت بروز خطا، پیغام مناسب نمایش داده می‌شود. خطاهای احتمالی شامل موارد زیر است:

I. درخواست کاربر برای ایجاد کانال و یا عضویت در کانال در حالی که در کانالی دیگر عضو است.
II. ایجاد کانال با نام‌هایی مشترک با کانال‌های موجود.
III. درخواست عضویت در کانالی که موجود نیست.

در صورت بروز هر یک از این خطاها، پاسخ مناسب نمایش داده می‌شود.

- تابع menu3

این تابع نیز ورودی و خروجی ندارد. در صورتی که کاربر با موفقیت در کانالی قرار بگیرد، به منوی آخر هدایت می‌شود. منو شامل امکان ارسال پیام، بررسی پیام‌های دیده نشده، مشاهده‌ی اعضای کانال و خروج از کانال است. مشابه بخش‌های قبلی بر اساس انتخاب کاربر، کد مربوطه اجرا می‌شود.

- تابع make_connect

برای ایجاد سوکت و اتصال به سرور که پیش از ارسال هر پیام صدا زده می‌شود.

فاز دوم: Server

در ابتدا باید اشاره کنم که برنامه در یک حلقه بی‌نهایت قرار دارد و در این حلقه بر اساس پیام ارسال شده توسط client آن را تحلیل کرده و تابع مربوطه را صدا می‌زند. در این توابع پاسخ مناسب ساخته می‌شود و سپس در main این پاسخ‌ها به client ارسال می‌گردد.

در ابتدا برنامه struct تعریف شده است که جهت نگهداری اطلاعات **کاربران آنلاین** استفاده می‌شود. بخش‌های مختلف این struct به شرح زیر است:

I. رشته‌ی onlineUser که همان نام کاربری فرد است.
II. رشته‌ی currentChannel که نام کانالی است که کاربر اکنون در آن عضو است.
III. عدد refreshNo که برای تعیین پیام‌های دیده نشده کاربر در کانال استفاده می‌شود.
IV. Boolean inChannel که به ما می‌گوید کاربر در کانالی عضو است یا نه.

- تابع checkToken

برای بررسی صحت توکن و اینکه آیا توکنی که از client فرستاده شده، در بین توکن کاربرهای آنلاین موجود است یا خیر. اگر خروجی 1- بود صحت توکن رد شده و در غیر این صورت شماره کاربر آنلاین از بین همه کاربران آنلاین برگردانده می‌شود.

- تابع checkLogin

این تابع در تابع login_Function استفاده می‌شود تا کاربری که Login است نتواند دوباره Login کند. ورودی آن نام کاربری است که می‌خواهیم وضعیت ان را تعیین کنیم و در خروجی 0 را به عنوان موفقیت و 1 را به عنوان عدم موفقیت (Login بودن کاربر) برمی‌گرداند.

- تابع randstring

برای ایجاد توکن استفاده می‌شود. در خروجی توکن تولید شده را می‌دهد و در ورودی طول دلخواه برای توکن وارد می‌شود.

- تابع register_Function

این تابع ورودی و خروجی ندارد. ابتدا بررسی می‌شود که آیا حساب کاربری با این نام وجود دارد یا خیر. اگر این نام قبلا استفاده شده بود، پاسخ "This name is not available" داده می‌شود. در

غیر این صورت فایلی برای این حساب در database ساخته شده و username و password را در آن یادداشت می‌شود.

- تابع login_Function

این تابع ورودی و خروجی ندارد. ابتدا بررسی می‌شود که آیا حساب کاربری با این نام وجود دارد یا خیر. اگر حسابی با این نام یافت نشد پاسخ "This name is not existing" داده می‌شود. در غیر این صورت فایل مربوط به آن حساب باز شده و password کاربر با password حساب موجود در database مقایسه می‌شود. اگر این دو رشته متفاوت بود پیام "Wrong password" داده شده و در غیر این صورت کاربر login شده و server به آن یک AuthToken ارسال می‌کند. پیش از این گفته شده ساخت توکن به کمک تابع randstring انجام می‌شود.

- تابع create_channel_Function

این تابع ورودی و خروجی ندارد. ابتدا توکن بررسی و در صورت صحت نام پیشنهادی کاربر بررسی می‌شود. اگر کانالی با این نام موجود نبود، کانال ساخته شده و فایل مربوط به آن در database ایجاد می‌شود. در این فایل اطلاعات اولیه کانال شامل پیام ایجاد کانال توسط کاربر و join شدن آن و همچنین نام کاربر در بخش subscribers نوشته می‌شود.

- تابع join_channel_Function

این تابع ورودی و خروجی ندارد. ابتدا توکن بررسی می‌شود. سپس نام کانال جهت موجود بودن در database چک می‌شود. در صورتی که مشکلی پیش نیاید؛ نام کاربر به بخش subscribers کانال اضافه شده و پیام join شدن او نیز به کانال افزوده می‌شود.

- تابع send_Function

این تابع ورودی و خروجی ندارد. مشابه بخش‌های قبلی ابتدا توکن بررسی می‌شود. سپس بررسی می‌شود که آیا کاربر در کانالی عضو هست؟ در صورت Error پیام مناسب ارسال می‌گردد و اگر مشکلی پیش نیاد فایل مربوط به آن کانال باز شده و پیام کاربر به database اضافه می‌شود.

- تابع refresh_Function

این تابع ورودی و خروجی ندارد. مشابه بخش‌های قبلی ابتدا توکن بررسی می‌شود و سپس بررسی می‌شود که آیا کاربر در کانالی عضو هست؟ در صورت Error پیام مناسب ارسال می‌گردد و اگر مشکلی پیش نیاید فایل مربوط به آن کانال باز شده و parse می‌شود. سپس پیام‌هایی که کاربر ندیده است (به کمک refreshNo) نمایش داده می‌شود. (به کمک تابع cJSON_GetArrayItem این کار انجام می‌گردد)

- تابع logout_Function

این تابع ورودی و خروجی ندارد. مشابه بخش‌های قبلی ابتدا توکن بررسی می‌شود و سپس بررسی می‌شود که آیا کاربر در کانالی عضو هست؟ اگر عضو بود؛ ابتدا از آن کانال خارج می‌شود. سپس اطلاعات آن از بین کاربران آنلاین حذف می‌شود.

فاز سوم: cJSON

- تابع cJSON_GetArraySize

ورودی این تابع آرایه‌ی مورد نظر از جنس *cJSON است و در خروجی آن تعداد اعضای آن داده می‌شود.

- تابع cJSON_AddItemToArray

ورودی اول این تابع آرایه‌ی مورد نظر از جنس *cJSON است و ورودی دوم آن چیزی است که قرار است به آرایه افزوده شود. این تابع خروجی ندارد.

- تابع cJSON_CreateArray

این تابع ورودی ندارد و تنها در خروجی یک آرایه‌ی خالی از جنس *cJSON به ما تحویل می‌دهد.

- تابع cJSON_Print

این تابع، ورودی از جنس *cJSON می‌گیرد و آن را به *char تبدیل می‌کند تا قابل چاپ باشد.

- تابع cJSON_Parse

ورودی آن *char و خروجی آن *cJSON است تا بتوان از آن در مراحل بعدی استفاده کرد.

- تابع cJSON_GetObjectItem

ورودی آن به ترتیب object مورد نظر و نام item است که می‌خواهیم آن را استخراج کنیم. در خروجی نیز آن item را به صورت *cJSON برمی‌گرداند.

- تابع cJSON_CreateObject

این تابع ورودی ندارد و در خروجی‌اش، object تولید شده را برمی‌گرداند. (که مشخصا خالی است.)

- تابع cJSON_AddItemToObject

در ورودی به ترتیب object را که می‌خواهیم به آن چیزی اضافه کنیم را می‌گیرد؛ سپس نام دلخواه برای item در object را می‌گیرد و در نهایت خود item را به شکل *cJSON دریافت می‌کند. خروجی این تابع void است.