

## به نام خدا

مستند پروژه‌ی درس مبانی برنامه نویسی

نام پروژه: پیام رسان داخلی!

استاد مربوطه: استاد شیرین بقولی زاده

ارائه دهنده: محمد مهدی ابوترابی

شماره دانشجویی: ۹۸۱۰۵۵۵۷

دانشگاه صنعتی شریف



دانشگاه صنعتی شریف

## فازا کلاینت:

تابع سوکت: این تابع در حقیقت سوکت را ایجاد می‌کند و برای وصل کردن به سرور و درخواست دادن به آن لازم است آن را صدا کنیم.

تابع register: این تابع در حقیقت برای ثبت نام کاربر است و وقتی کاربر در تابع main گزینه ثبت نام را بزند این تابع فراخوانده می‌شود که در آن ابتدا username و password را از کاربر می‌گیرد و با استفاده از تابع‌های استرینگ استرینگ لازمه را برای فرستادن به سرور تولید میکند و سپس تابع وصل را صدا می‌زند که در ادامه آن را توضیح می‌دهیم. و پیام به سرور فرستاده می‌شود و جواب را هم از سرور می‌گیرد و سپس بر اساس successful یا error دادن سرور پاسخ مربوطه را بر روی صفحه می‌دهد و به منوی بعد می‌رود.

تابع vasi: این تابع در حقیقت برای فرستادن پیام به سرور و دریافت پاسخ سرور است که ورودی‌های آن چهار استرینگ است که اولی درخواست و دومی پاسخ سرور و سومی type و چهارمی content است که با استفاده از دستورات cJSON تایپ و کانتنت را از json ارسالی سرور استخراج میکند.

تابع vasi refresh: این تابع همان تابع وصل برای دستور رفرش است. از آنجا که در دستور رفرش جیسونی که سرور به ما بر می‌گرداند در بخش content آن آرایه‌ای از جیسون‌ها برگردانده می‌شود باید با استفاده از دستورات آرایه‌ای آن از آن پیام‌ها را دریافت کنیم.

تابع vasi channel members: از آن جا که در دستور رفرش جیسونی که سرور به ما بر می‌گرداند آرایه است برای آن هم از یک تابع وصل جدا استفاده می‌کنیم. که در آن فرستادن و دریافت پیام را از سرور دریافت می‌کنیم.

تابع login: این تابع در حقیقت برای ورود کاربر است و وقتی کاربر در تابع main گزینه ورود را بزند این تابع فراخوانده می‌شود که در آن ابتدا username و password را از کاربر می‌گیرد و با استفاده از تابع‌های استرینگ استرینگ لازمه را برای فرستادن به سرور تولید میکند و سپس تابع وصل را صدا می‌زند که در ادامه آن را توضیح می‌دهیم. و پیام به سرور فرستاده می‌شود و جواب را هم از سرور می‌گیرد و سپس بر اساس successful یا error دادن سرور پاسخ مربوطه را بر روی صفحه می‌دهد و به منوی بعد می‌رود.

تابع menuasli: این تابع همان منوی مربوط به چنل است که پیام‌های مربوط به آن را چاپ میکند.

تابع create channel: این تابع برای ساخت channel است. که در حقیقت درخواست مربوطه به ساخت کانال را ساخته و سپس تابع وصل را صدا می‌زند.

تابع join channel: این تابع برای جوین شدن به کانال است که کار آن مانند ساخت چنل است دقیقا و همانکار را انجام می‌دهد.

تابع `logout`: این تابع برای لاگ اوت است و پیام را با استفاده از توابع استرینگ می‌فرستد و سپس با استفاده از تابع وصل استرینگ ساخته شده را به سرور فرستاده و پاسخ مربوطه را دریافت می‌کند.

تابع `chatroom`: این تابع در حقیقت منوی چت و مسیج فرستادن این‌ها را نشان می‌دهد.

تابع `find_user`: این تابع در حقیقت برای پیدا کردن کاربر بین کاربرهای چنل است که پیام مورد نیاز برای سرور در آن تولید می‌شود و سپس هم تابع وصل فراخوانده می‌شود.

تابع `sendmessage`: این تابع برای فرستادن پیام است که در آن ابتدا پیام را از کاربر دریافت کرده و سپس استرینگ مورد نیاز را تولید کرده و تابع وصل را صدا کرده و به سرور این را می‌فرستد.

تابع `find_message`: این تابع پیام مورد نیاز کاربر را دریافت کرده و سپس پیام مربوطه را به سرور می‌فرستد و تابع وصل را صدا می‌زند.

تابع `refresh`: این تابع برای رفرش است که پیام مورد نظر را ساخته و این را به تابع وصل رفرش می‌فرستد و بقیه کارها در آن انجام می‌شود که در بالاتر آن را توضیح دادیم.

تابع `channel members`: این تابع استرینگ مورد نیاز سرور را تولید کرده و سپس تابع وصل کاربرها را فرا می‌خواند که بقیه کارها در آن انجام می‌شود که در بالاتر آن را توضیح دادیم.

تابع `leave channel`: این تابع نیز پیام مورد نیاز را تولید کرده و آن را به تابع وصل پاس می‌دهد. و بقیه کارها در آن انجام می‌شود و پیام مربوطه به سرور فرستاده و دریافت می‌شود.

تابع `arrow_play_menu1`: این تابع و دو تابع بعدی برای کار کردن با `arrowkey` ها برای جا به جایی در منو ها طراحی شده است که با دکمه های `W` و `S` می‌توانید در منو ها بالا پایین بروید.

در تابع `main` هم منوهای مربوط به آن صدا زده می‌شود و چاپ می‌شود.

## فاز ۲ سرور:

استراکت کاربران: این استراکت حاوی اطلاعات کاربران همچون نام کاربری، پسورد و نام کانالی که در آن عضو است و دیگر اطلاعات مورد نیاز همه کاربران ثبت نام شده است.

تابع سوکت: همان طور که در کلاینت یک تابع سوکت داشتیم این جا هم یک تابع سوکت داریم که در آن توسط سرور یک سوکت ساخته می‌شود و منتظر دریافت `request` از سمت کلاینت می‌ماند.

تابع `load users`: این تابع در ابتدای بالا آمدن سرور می‌رود از تمام فایل های جیسون که در دیتابیس برای کاربران ذخیره کردیم اطلاعات مورد نیاز را استخراج کرده و در استراکت های یوزرها قرار می‌دهد. که این کار با خواندن تمام فایل ها صورت می‌گیرد.

تابع `reuest`: این تابع بخش مهمی از برنامه است و درخواست کلاینت را دریافت کرده و بر اساس این که آن در خواست چی می باشد یکی از توابع مورد نیاز را صدا می زند که آن کار را انجام دهد.

تابع `find_massege`: این تابع می رود از میان پیام های کانال مربوطه می گردد و پیامی که حاوی این کلمه بوده است را برمی گرداند. و از فایل مربوط به آن کانال به تمام پیام ها دسترسی پیدا می کند.

تابع `find_user`: این تابع می گردد میبندد آیا همچین کاربری در بین کاربر های `register` شده وجود دارد یا نه. اساس کارش هم با باز کردن فایل آن است که اگر فایلی وجود نداشت و پوینتر مربوط به فایل آن `null` شد عدد 1 در غیر این صورت عدد 0 را بر می گرداند.

تابع `find_user2`: این تابع در بین کاربران کانال فعلی می گردد و میبندد آیا همچین کاربری در کانال وجود دارد یا نه که اگر وجود داشت پیام موفقیت آمیز و اگر نبود پیغام یافت نشدن را می فرستاد.

تابع `find_user_in_struct`: این تابع در استراکت مربوط به کاربران نام کاربر را جستجو کرده و اندیس «ا» کاربر را در آرایه ای از استراکت ها که داریم بر می گرداند.

تابع `find_channel`: همانند تابع `find_user` است اما در میان کانال ها. در حقیقت چک میکند میبندد آیا همچین کانالی قبلا ساخته شده یا نه. که این کار را با استفاده از باز کردن فایل مربوط به آن انجام می دهد. اگر پوینتر آن `null` شد و این کانال وجود نداشت عدد 1 در غیر این صورت عدد 0 بر می گرداند.

تابع `find_token`: در میان استراکت یوزرها می گردد و اندیس کاربر با توکنی که داده شده را بر می گرداند. و اگر همچین توکنی وجود نداشت عدد منفی یک را بر می گرداند.

تابع `create_token`: این تابع با استفاده از `rand` و `srand` یک رشته ی 32 تایی رندم از حرف و عدد درست میکند و به عنوان توکن فرد مورد نظر بر می گرداند.

تابع `register`: این تابع ابتدا یک فایل با نام و پسورد داده شده از جنس جیسون می سازد. سپس یک خانه از آرایه ی استراکت هم برای آن پر می شود. سپس جیسون مورد نظر را ساخته و آن را برای کلاینت می فرستد.

تابع `create_channel`: این تابع ابتدا در فولدر کانال ها یک فایل برای کانال جدید می سازد. سپس پیام اول را در آن با فرمت جیسون به این ترتیب که این یوزر این کانال را ساخت ایجاد می کند. و در نهایت پیغام این که با موفقیت این پیام ساخته شد در غالب جیسون برای کلاینت می فرستد.

تابع `join_channel`: این تابع می رود فایل مربوط به آن کانال را باز کرده و این کاربر را به جمع کاربران آن کانال اضافه کرده و فایل جیسون آن کانال را دوباره باز نویسی انجام می دهد. و در نهایت مانند دیگر توابع پیغام موفقیت آمیز بودن آن را برای کلاینت می فرستد.

تابع `send_message` : این تابع مسیج را از کاربر دریافت کرده و سپس آن را به فرمت جیسون در آورده و به آرایه ای از جیسون که در فایل مربوط به آن کانال ذخیره شده اضافه کرده و در نهایت پیغام موفقیت آمیز بودن آن را به فرمت جیسون در آورده و به کلاینت می فرستد.

تابع `refresh` : این تابع از فایل مربوط به کانال پیام ها را از آن جا که آخرین بار کاربر دیده تا انتها می فرستد. این که کاربر چند پیام دیده را به وسیله ی یک متغیر `int` در استراکت مربوط به کاربر نگه داشته می شود و هر بار بعد رفرش آپدیت می شود و در نهایت پیغام موفقیت آن به کلاینت در فرمت جیسون سند می شود.

تابع `channel_members` : این تابع نیز در استراکت کاربران می گردد و کاربرانی که در همپین کانالی عضو هستند را به صورت یک لیست در آورده و سپس این را به صورت فرمت جیسون به کلاینت بر می گرداند.

تابع `leave channel` : این کاربر یک پیغام لغت کاربر را در فایل مربوط به کانال به فرمت جیسون اضافه کرده و سپس در استراکت مربوط به آن نام آن کانال را از اطلاعات کاربر حذف می کند و پیغام موفقیت آمیز بودن را به کلاینت بر می گرداند.

تابع `logout` : این تابع توکن ساخته شده برای کاربر را از بین برده و کاربر را از لیست کاربرهای آنلاین خارج کرده و پیغام موفقیت را به کلاینت بر می گرداند.

تابع `send` : این تابع کار جیسون ساختن را برای اکثر توابع ما و ارسال آن را انجام می دهد. دو تا استرینگ دریافت می کند به عنوان تایپ و کانتنت و خودش جیسون مورد نظر را می سازد و آن را به کلاینت ارسال می کند.

در تابع `main` هم ابتدا کارهایی برای راه اندازی سرور انجام می شود و سپس اگر فولدر برای دیتابیس نداشتیم فولدرهای مورد نظر را هم می ساخت و مقدار دهی اولیه انجام می شود و سپس به درخواست های کلاینت گوش می دهد.

### فاز ۳:

استراکت `cjson` : این استراکت شامل استرینگ های `type, content, valuestring` و نوع و دو تا پویتر یکی `before` و `next` است.

تابع `cjson_createobject` : این تابع در واقع یک فضا برای این استراکت در نظر می گیرد و مقدار دهی اولیه به پویترها و استرینگ ها انجام می دهد. و خروجی آن یک `*cjson` است.

تابع `cJSON_CreateString` : این تابع یک استرینگ دریافت کرده و یک سیجیسون ساخته و در کانتنت آن این استرینگ را ریخته. و نوع آن را استرینگ قرار می دهد.

تابع `cJSON_AddItemToObject` : این تابع یک آیتم را که میتواند آرایه یا استرینگ باشد را به جیسون اضافه می کند. که برای هر کدام از این حالت ها به وسیله ی `link list` آیتم را به جیسون خود اضافه می کنیم.

تابع `cJSON_PrintUnformatted` : این تابع پوینتر سیجسون را به استرینگ تبدیل میکند. سه حالت دارد. یا نوع آن استرینگ است که تبدیل آن به استرینگ آسان است. یا نوع آن آرایه است که باید آیتم ها را به وسیله ی لینک لیست خواند و کنار هم قرار داد. یا جیسون عادی است که از بهم وصل کردن 4 استرینگ به آسانی به دست می آید.

تابع `cJSON_CreateArray` : این تابع تا مقدار زیادی شبیه درست کردن آبجکت است. در آن یک مقدار حافظه برای آن تخصیص می دهیم و نوع آن را آرایه در نظر می گیریم.

تابع `cJSON_AddItemToArray` : به وسیله ی `link list` در این تابع در انتهای آخرین عضو آرایه ی خود ایتیم جدید را اضافه کرده و با بقیه لینک میکنیم که البته با توجه به شرایط حالت های خاص برای آن در نظر گرفته می شود.

تابع `cJSON_GetArraySize` : این تابع با استفاده از شمردن کاما ها تعداد عضو های آرایه را بدست می آورد. البته همه ی حالاتی که چه آرایه از استرینگ چه آرایه از سیجسون ها باشد در آن در نظر گرفته می شود.

تابع `cJSON_Parse` : این تابع حالات مختلف ورودی را در نظر گرفته و بر اساس ورودی آن را به بخش های مختلف می شکند. چه آرایه باشد ورودی چه آبجکت های ساده درست کار می کند.

تابع `cJSON_GetObjectItem` : این تابع برای دریافت آیتم از آبجکت است که در لینک لیست به وجود آمده آن آیتم را جستجو می کند و آن را برمی گرداند در یک سیجسون که در خود تابع ساخته می شود.

تابع `cJSON_GetArrayItem` : این تابع بر اساس کاما اگر آرایه عادی باشد یا بر اساس کروشه اگر آرایه ای از سیجسون ها بود آرایه مورد نظر را می شکند و آیتم عدد مورد نظر را بر می گرداند.