به نام خدا

پروژه پایانترم

مبانی برنامه سازی

اپلیکیشن چت

گروه استاد بقولی زاده

سازنده: امیرعلی ابراهیم زاده

شماره دانشجویی: 98105546

دانشگاه صنعتی شریف

زمستان 98

Table of Contents

About the Project **3**

Client **4**

Register 4

Login 4

Create Channel 5

Join Channel 5

Logout 5

Send a message 6

Refresh 6

Viewing Channel's Members 6

Leave Channel 6

Server **7**

Register 7

Login 7

Create Channel 7

Join Channel 8

Logout 8

Send a message 8

Refresh 8

Viewing Channel's Members 9

Leave Channel 9

درباره پروژه

در این پروژه برنامه یک چت اپلیکیشن ساده نوشته است، در این چت اپلیکشن می توان کانال ساخت و در آن به نوشتن پیام پرداخت و نام کاربران را بررسی کرد. سرور این اپلیکیشن توانایی ذخیره اطلاعات کاربرها، کانال ها و پیام های رد و بدل شده میان آن ها را دارد و به طور همزمان چند کاربرد می توانند login باشند.

برای ساده کردن فرآیند نوشتن برنامه، پروژه به دو قمست client و server تقسیم شده و در دو فایل جدا کدهای این دو بخش نوشته شده است. کتابخانه cJSON نیز در این پروژه برای ساخت، رد و بدل کردن و ذخیره رشته های مورد استفاده در پروژه استفاده شده است.

در ادامه به شرح جزئیات دو قسمت server و client و نحوه کار آنها می پردازیم.

فاز یک: Client

در این پروژه کلاینت نقش واســـط بین server و کاربر را دارد، به طوری اطلاعات را دریافت کرده، آن را بصورت یک رشته برای server ارسال نموده و پاسخ سرور را به صورت یک رشته JSON دریافت می کند، سپس رشته را تجزیه کرده و اطلاعات درون آن را به کاربر نمایش می دهد.

ســاختار برنامه در این بخش به گونه ایســت که تابع main، تابع chat را صــدا زده و در ادامه این تابع تمام اعمال از جمله دریات و پردازش رشــته و ارتباط با ســرور را انجام می دهد. تابع چت پس از تعریف متغیرها بلافاصــله وارد یک while بزرگ شــده که شــرط آن همیشــه true اســت. در هر بار اجرا شــدن حلقه از کاربر می خواهیم که یا register و یا login کند.

```
Account Menu:
1: Register
2: Login
```

Register

درصورتی که کاربر گزینه register را انتخاب کند، وارد یک حلقه می شود و در حلقه نام کاربری و رمز عبور اکانت ثبت نامی را وارد می کند و ســـپس از کاربر دوباره درباره register یا login پرسیده می شود. این حلقه تا زمانی که کاربر گزینه login را انتخاب نکند ادامه می یابد و کاربر می تواند بینهایت با با اســـامی مختلف ثبت نام کرده و client اطلاعات او را برای ذخیره شدن به server می فرستند.

Login

در صورت انتخاب login، نام کاربری و رمز عبور کاربر گرفته شده و client این دو را در رشته ای به هم متصل کرده و برای server می فرستد. در صورتی که نام کاربری و یا رمز عبور نا درســت باشــد client پس از دریافت پیغام خطا از server دوباره تابع chat را صدا می زند تا کاربر به منوی اصلی برگردد. در صورت اعلام درستی نام کاربری و رمز عبور، برنامه یک توکن که شــناســه کاربران آنلاین اســت از server دریاف کرده و کاربر را به منوی بعدی می برد.

Create Channel

هنگامی که کاربر با این منو برخورد می کند برنامه وارد یک حلقه while همیشه صادق می شود. در صورتی که گزینه create channel را انتخاب کند، برنامه دوباره وارد یک حلقه while همیشه صادق می شود و نام کانال مورد نظر کاربر را دریافت می کند و رشته ای مرکب از عبارت create channel، توکن کاربر و نام کانال ایجاد می کند و برای server ارسال می کند. سپس دوباه منوی بالا نمایش داده می شود و تا وقتی که کلاینت انتخابی جز create channel نداشته باشد کاربر در حلقه while دوم باقی می ماند.

Join channel

کاربر از حلقه create channel خارج می شود و در صورتی که گزینه join channel را انتخاب کرده باشد شرط مربوط به این گزینه برقرار شده و نام کانال و توکن و عبارت join channel را در قالب یک رشته می سازد و برای server می فرستد. اگر server تشخیص دهد که کاربری با این توکن و یا کانالی با این نام وجود ندارد، برای کاربر پیام خطا می فرستد و در صورت وجود کانال و درستی توکن، وارد یک حلقه while می شود که شرط درستی آن، موفقیت آمیز بودن فرآیند عضویت در کانال است. این حلقه حاوی تمام دستورات مربوط به کانال عضو شده است که در ادامه توضیح داده خواهد شد.

Logout

برنامه وارد یک حلقه while می شود و توکن و عبارت logout را برای server می فرستد و تا زمانی که server موفقیت فرآیند را اطلاع ندهد به این کار ادامه می دهد. در صورتی که Logout موفقیت آمیز باشد تابع chat دوباره صدا زده می شود و برنامه از ابتدا شروع به کار می کند.

```
You joined the channel!
What do you want to do?
1: Send a message
2: Refresh
3: Viewing channel's members
4: Leave channel
```

Send a message

منوی بالا پس از عضویت در کانال نشان داده می شود و کاربر در صورت انتخاب گزینه send a message، عبارتی را وارد می کند که برنامه آن را به وسیله تابع gets دریافت می کند. برنامه رشته ای حاوی عبارت send، پیام کاربر و توکن کاربر به وسیله تابع sprintf می سازد و برای server می فرستد.

Refresh

در صورت انتخاب این گزینه، برنامه رشته ای را حاوی کلمه Refresh و توکن کاربر می سازد و برای server می فرستد و server رشته ای در قالب json برای client می فرستد که توسط توابع json که client در اختیار دارد، رشته ارسال شده را تجزیه کرده و پیام های ناخوانده را به کاربر نشان می دهد.

Viewing channel’s members

برنامه به وسیله تابع sprintf رشته ای حاوی عبارت channel members و توکن کاربر می سازد و برای server می فرستد و رشته json را برای client ارسال می کند و client رشته را تجزیه کرده و قسمتی که حاوی اسامی اعضا است را چاپ می کند.

Leave channel

در صورت انتخاب این گزینه، برنامه رشته ای را حاوی کلمه Leave و توکن کاربر می سازد و برای server می فرستد و server رشته ای در قالب json برای client می فرستد، برنامه در صورتی که رشته دریافتی از server اجرای عمل را موفقیت آمیز اعلام کند، از کانال خارج شده، به منوی دوم می رود.

فاز دوم: Server

در این بخش از برنامه، تابع main حاوی یک while با شرط همیشه صادق است، تمام اعمال در این حلقه انجام می شود. این حلقه در هر دور، یک socket می سازد، به client متصل می شود و رشته ای از آن دریافت کرده و سپس اعمال را انجام می دهد، پاسخی ارسال می کند و socket را می بندد و دوباره منتظر پیامی از یک کاربر می شود. Server اطلاعات مربوط به کاربران آنلاین از جمله نوبت آنلاین شدن شان، توکن و نام آنها را در یک structure ذخیره می کند و برای هر کانال نیز شماره ای در نظر می گیرد و به همراه نام آن کانال در یک structure ذخیره می کند. پس از ایجاد socket، client رشته ای می فرستد و server با بررسی حروف اولین کلمه آرایه دریافتی، تشخیص می دهد که رشته حاوی چه دستوری است.

Register

Server با تجزیه رشته دریافتی نام کاربری و رمز عبور مد نظر کاربر را استخراج کرده و آدرسی از یک فایل از نوع txt. می سازد و سعی می کند فایلی با این نام را بخواند، اگر موفق شد که یعنی کاربری با این نام وجود دارد، پس برای client پیام خطا می فرستد. اگر فایلی با این نام وجود نداشت چنین فایلی را می سازد و رمز عبور کاربر را در آن ذخیر می کند.

Login

Server با تجزیه رشته دریافتی نام کاربری و رمز عبور مد نظر کاربر را استخراج کرده و آدرسی از یک فایل از نوع txt. می سازد و سعی می کند فایلی با این نام را بخواند، اگر موفق شد که یعنی کاربری با این نام قبلا register کرده و موجود است، اگر رمز نادرست باشد که خطا می فرستد و اگر درست باشد، به وسیله تابع rand رشته ای از اعداد را بعنوان توکن برای client در قالب jcon می فرستد و سپس نام کاربری و توکن کاربر را در structure ذخیره کرده و در متغیری از structure کاربر آنلاین که channel_num نام دارد، 1- را ذخیره می کند بدین معنی که کاربر آنلاین در هیچ کانالی نیست. در صورتی که فایلی با نام Login شده یافت نشد، پیغام خطا ارسال می کند.

Create channel

Server رشته ای با نام گفته شده و با پایان txt. می سازد و سعی می کند فایلی با این نام را بخواند، اگر فایل با این نام وجود داشت که یعنی کانالی با این نام ساخته شده، پس پیغام خطا می فرستد. در غیر این صورت چنین فایلی را می سازد و موفقیت عملیات را به کاربر اعلام می کند. server توکن دریافتی را با توکن کاربران آنلاین مقایسه کرده و شماره آن را میان کاربران ذخیره می کند و از طریق این شماره، نام کاربری ای که در structure ذخیره شده را دریافت کرده و در فایل کانال می نویسد که این کاربر کانال را ساخته است.

Join channel

Server اگر نتوانست فایلی با نام کاربر بسازد، خطا برمی گرداند، اگر توانست، به کاربر موفقیت آمیز بودن فرآیند را اعلام می کند. سپس به وسیله توکن شماره کاربر و به وسیله نام کانال شماره کانال را می یابد و شماره کانال را در structure کاربر ذخیره می کند. server محتویات فایل کانال را در یک رشته ذخیره می کند، سپس اعلام اضافه شدن کاربر در کانال را به این رشته می چسباند.

Send a message

Server توکن را از انتهای رشته جدا کرده و نام کاربر را می یابد. سپس فایل کانال را باز کرده و اطلاعات آن را به رشته ای تعریف شده، به وسیله توابع json اضافه می کند. نام و پیام کاربر را به رشته اضافه کرده و در فایل با نام کانال کل رشته را ذخیره می کند.

Viewing channel’s members

Server توکن دریافتی را با توکن کاربران آنلاین مقایسه کرده و کاربر را پیدا می کند. شماره کانالی که در آن عضو است را ذخیره می کند و سپس نام هر کاربری که آن شماره کانال را در structure خود دارد به صورت json در یک رشته ذخیره می کند و json را به client می فرستد.

Refresh

Server کاربر و کانال را از طریق توکن پیدا می کند و محتوای فایل کاربر را در آرایه ای ذخیره می کند. در صورتی که به جز رمز عبور و نام کانال های دیگر در فایل ذخیره نشده باشد، کل محتوای کانال را برای کاربر می فرستد و سپس نام کانال و تعداد کاراکترهای

فایل کانال را در انتهای فایل کاربر ذخیره می کند. اگر نام کانال در فایل کاربر ذخیره شـده باشد، تعداد کارکتر خوانده شده تواسط کاربر را ذخیره می کند و تمام کاراکترهایی که اندیس آرایه شـان بیشـتر از تعداد کاراکتر های خوانده شـده اسـت را ذخیره کرده، پس انجام عملیات هایی بر روی رشته آن را به صورت جیسون در می آورد و برای کاربر می فرستد و سپس دوباره تعداد کاراکترهای کانال را پس از آخرین پیام ها می خواند و دوباره در فایل کاربر ذخیره می کند.

Leave channel

Server با استفاده از توکن فرستاده شده، شماره کاربر را پیدا می کند. هر کاربر یک شماره کانال دارد که حاوی شماره کانالی است که در آن عضو است، شماره کانال کاربر Leave کرده به 1- تغییر می کند که به معنی نبودن در هیچ کانالی اسـت. به وسـیله json به کاربر اعلام می شـود که خروج موفقیت آمیز بوده، همچنین خروج کاربر در فایل مربوط به کانال ذخیره می شود.

Logout

شـــماره کاربر آنلاینی که logout کرده از طریق توکن او به دســـت می آید و اطلاعات مربوط به او در structure پاک می شـــود. ســـپس با اســـتفاده از یک حلقه for اطلاعات کاربر آنلاین i + 1 را در اطلاعات کاربر i ام ذخیره می کند که i از شماره کاربر خروجی آغاز می شـود. پیامی در رابطه با Logout سـاخته و به client فرسـتاده می شـود. تعداد کاربران آنلاین یکی کم می شود.