

به نام خدا

## پروژه پایان‌ترم درس برنامه‌نویسی پیشرفته

نیم‌سال دوم ۱۴۰۱-۱۴۰۰

۱. تمامی فایل‌های کد را به همراه فایل متنی که در قالب pdf است (مورد سوم را بخوانید) به صورت یک فایل آرشیو zip (zip != rar) که به قالب زیر نام‌گذاری شده است، بارگذاری نمایید.

AP-FinalProject-LastName1-LastName2-StudentNumber1-StudentNumber2.zip

۲. در سوال‌هایی که ورودی و خروجی مطلوب آن‌ها مشخص شده است، رعایت نحوه ورودی گرفتن و نمایش خروجی اهمیت بسیاری دارد. دقیقاً همان‌طور که از شما خواسته شده است ورودی‌ها را خوانده و خروجی‌ها را تولید کنید. (سوالات دستی تصحیح می‌شوند اما لازم است که از فرمت ذکر شده تبعیت نمایید)

۴. در صورت مشاهده هرگونه تقلبی، طبق موارد گفته شده در قوانین درس برخورد خواهد شد.

۵. در صورت وجود هرگونه ابهام می‌توانید از طریق آدرس ایمیل «ap.winter2022@gmail.com» با تدریس‌یاران در ارتباط باشید.

۶. امکان آپلود با تاخیر به هیچ‌وجه وجود ندارد.

مهلت تحویل: تا سه‌شنبه ۲۱ تیر ۱۴۰۱ ساعت ۲۳:۵۹ شب

## پیش گفتار

برای انجام این پروژه در اولین فرصت هم‌گروهی مناسب پیدا کنید! این پروژه برای گروه‌های حداکثر دو نفره تعریف شده است. انجام دادن این پروژه به صورت تکی نمره اضافه نخواهد داشت. هم‌گروهی شما می‌تواند هر یک از دانشجویان این درس باشد و محدودیت مشترک بودن گروه درس و کارگاه وجود ندارد.

استفاده مناسب از مفاهیم تدریس شده ضروری است. طراحی خوب و منطقی کلاس‌ها و اینترفیس‌ها باید متناسب با اصول برنامه نویسی شی‌گرا باشد. رعایت اصول پنهان سازی اطلاعات (information hiding)، سلسله مراتب ارث‌بری جهت استفاده مجدد از کدها (code reusability)، استفاده از چند ریختی و سایر نکات تدریس شده الزامی است. سعی کنید پیش از شروع پیاده‌سازی، تعداد و نام کلاس‌ها، فیلدها و متدهای مورد نظر برنامه را تحلیل کرده و مطابق با تحلیل و طراحی انجام‌شده، برنامه را پیاده‌سازی کنید. مستندسازی به کمک Javadoc، کامنت‌گذاری و رعایت اصول کدنویسی خوانا برای همه کلاس‌های پیاده‌سازی شده الزامی است.

برنامه را پیش از بارگذاری به خوبی تست و اشکال زدایی کنید! همه کلاس‌ها و متدهای موجود در برنامه‌ها کاملاً مورد بررسی و آزمون قرار بگیرند تا از درستی عملکرد برنامه اطمینان حاصل کنید. حالت‌های مختلف ورودی توسط کاربر باید بررسی شود و در صورت لزوم، پیغام خطای مناسب نمایش داده شود.

در قسمت‌های مختلف پروژه باید خطاهای مختلف بررسی شوند و در قبال آن رفتار مناسبی از برنامه دریافت شود. پس شما در صورت نیاز باید برای تمامی قسمت‌ها عملیات Exception Handling را برای استثناهای Checked و همچنین در مواقع لزوم برای Unchecked پیاده‌سازی کنید.

در این پروژه شما می‌بایست کد خود را در گیت (به صورت private) نیز قرار دهید. توجه شود صرف قرارگیری و آپلود پروژه در گیت کفایت نمی‌کند و شما باید کامیت‌های مناسب و مستمر داشته باشید. (همچنین تقسیم کار بین شما و هم‌تیمی‌تان به صورت درست و منطقی انجام شده باشد)

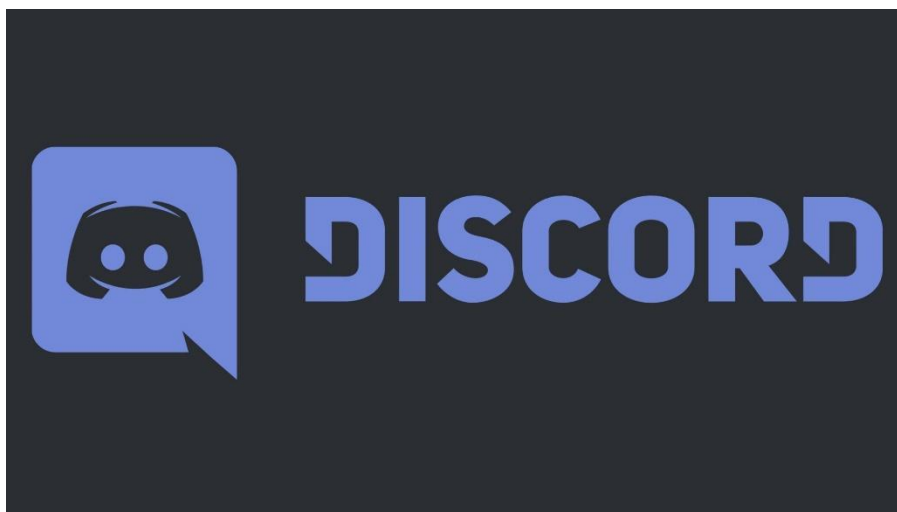
حداکثر نمره قابل کسب از بخش‌های امتیازی ۲۵ نمره است. در نتیجه در صورت کسب کامل نمره بخش‌های اصلی و بخش‌های امتیازی نمره ۱۲۵ را دریافت خواهید کرد. دقت کنید این بدان معنی است که اگر بخشی از نمره اصلی را از دست دهید، امکان رسیدن به حداکثر نمره ۱۲۵ وجود ندارد.

## صفحه

## فهرست

۵	صفحه ورود و ثبت نام
۶	صفحه پروفایل کاربری
۶	تعیین وضعیت
۱۲	ساخت سرور (متفاوت با سرور backend)
۱۳	تنظیمات سرور
۱۴	افزودن کاربران به سرور
۱۵	نمایش کاربران موجود در سرور
۱۵	حذف کردن کاربران از سرور
۱۵	تغییر نام سرور
۱۶	افزودن کانال به سرور
۱۷	انواع کانال
۲۰	واکنش
۲۰	سنجاق کردن
۲۱	پخش آهنگ (۱۰ امتیاز)
۲۲	برقراری تماس (۵ + ۵ امتیاز برای خصوصی و گروهی)
۲۲	نمایش وضعیت در حال نوشتن کاربران (۵ امتیاز)
۲۳	ذکر کردن و نام بردن از یک کاربر (۵ امتیاز)

## دیسکورد<sup>۱</sup>



**دیسکورد** یک نرم‌افزار برقراری تماس تلفنی در بستر شبکه برای سیستم عامل‌های ویندوز<sup>۲</sup>، آی‌اواس<sup>۳</sup>، اندروید<sup>۴</sup>، لینوکس<sup>۵</sup> و مک‌اواس<sup>۶</sup> و همچنین قابل اجرا در مرورگر وب است.

دیسکورد برای رایانه‌های شخصی به‌طور ویژه برای استفاده در هنگام بازی ساخته شده‌است، این نرم‌افزار از ویژگی‌هایی چون زمان پاسخگویی کم، سرورهای چت رایگان، تماس صوتی و تصویری بین دو یا چند کاربر، به اشتراک گذاری صدا و تصویر رایانه‌ی شخصی و سرورهای اختصاصی برخوردار است.

در این پروژه قصد داریم یک نسخه ساده شده از این نرم‌افزار را پیاده‌سازی کنیم. برای آشنایی بیشتر می‌توانید آن را از طریق این لینک دانلود و نصب نمایید. تمرکز پروژه میان‌ترم بر روی منطق و در واقع «back-end»، و تمرکز پروژه پایان‌ترم بر روی پیاده‌سازی محیط گرافیکی (GUI) برای آن است.

شما باید برای این پروژه دو نرم‌افزار توسعه دهید. یک مورد تحت عنوان «سرور»، برای ارائه خدمات و مورد دوم «کلاینت»، نرم‌افزاری که کاربران از آن استفاده می‌کنند (تاکنون شما تنها با مورد دوم سر و کار داشته‌اید).

---

<sup>1</sup> Discord

<sup>2</sup> Windows

<sup>3</sup> iOS

<sup>4</sup> Android

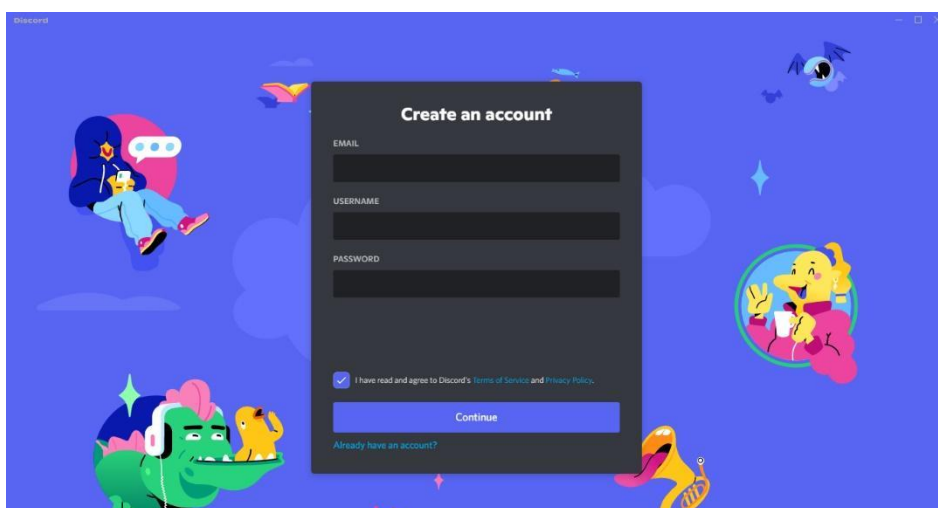
<sup>5</sup> Linux

<sup>6</sup> macOS

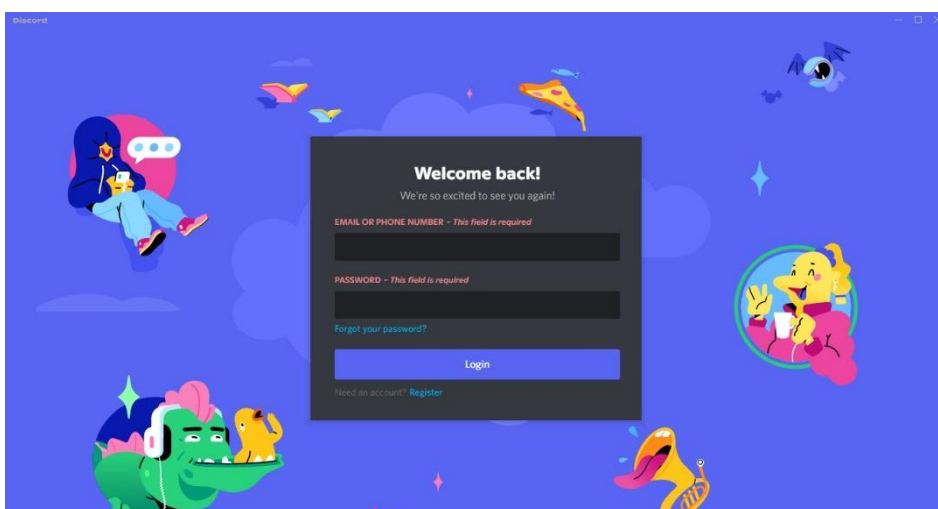
## حساب کاربری

### صفحه ورود و ثبت نام

در ابتدای اجرای برنامه باید صفحه‌ای نمایش داده شود که در آن کاربر با وارد کردن نام کاربری و رمز عبور بتواند به صفحه اصلی خود وارد شود. همچنین در این صفحه باید گزینه‌ای وجود داشته باشد که صفحه ثبت نام را نمایش دهد و کاربرانی که قصد ثبت نام در برنامه را دارند باید مشخصات خود را در این صفحه وارد و درخواست ثبت نام بدهند، در صورت موفقیت آمیز بودن ثبت نام، صفحه ورود دوباره نمایش داده شود. همچنین در صفحه ثبت نام باید گزینه‌ای برای برگشتن دوباره به صفحه ورود وجود داشته باشد.



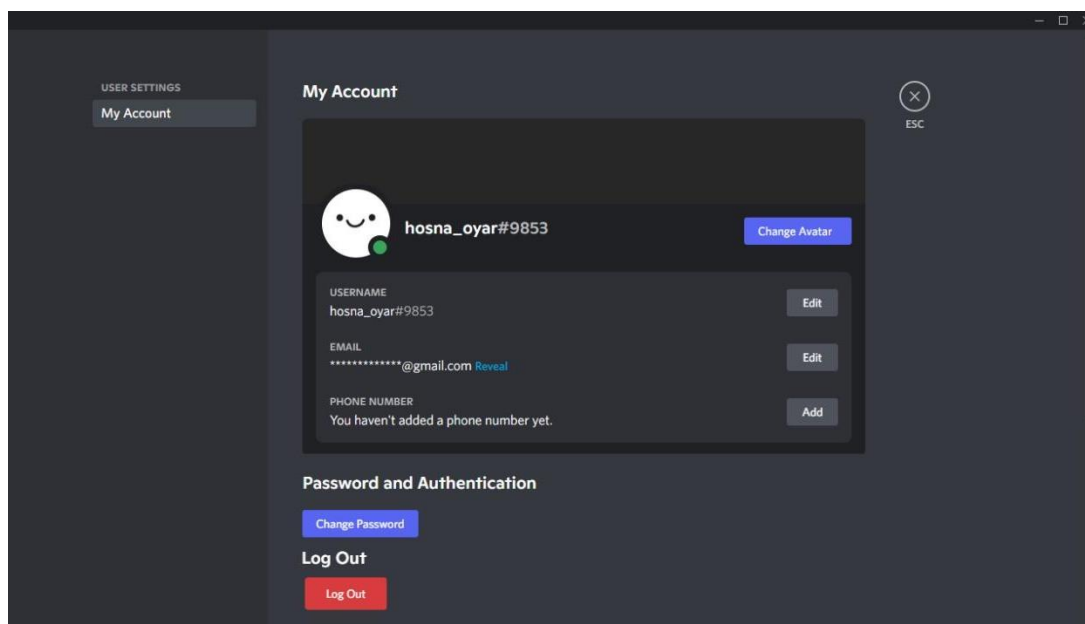
۱. صفحه ثبت نام



۲. صفحه ورود

## صفحه پروفایل کاربری

در این صفحه باید اطلاعات کاربر مانند نام کاربری، عکس پروفایل، ایمیل و شماره تلفن فرد نمایش داده شود. در این صفحه هر کدام از اطلاعات کاربر را می توانیم ادیت کنیم و همچنین کاربر باید بتواند عکس پروفایل خود را عوض کند. در این صفحه باید گزینه ای وجود داشته باشد که به کمک آن بشود از حساب کاربری فعلی خارج شد و به صفحه ورود بازگشت.

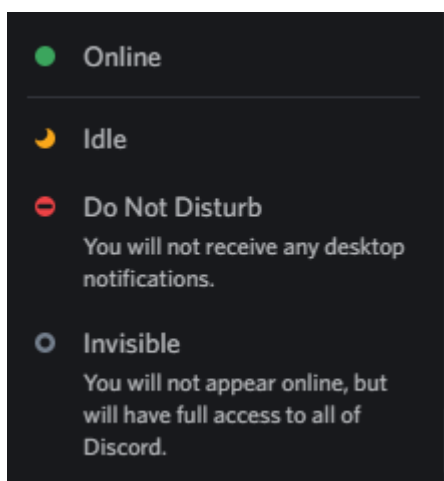


۳. صفحه حساب کاربر

## تعیین وضعیت

کاربر می تواند برای خود یک وضعیت<sup>۷</sup> تعریف کند:

- Online ○
- Idle ○
- Do Not Disturb ○
- Invisible ○

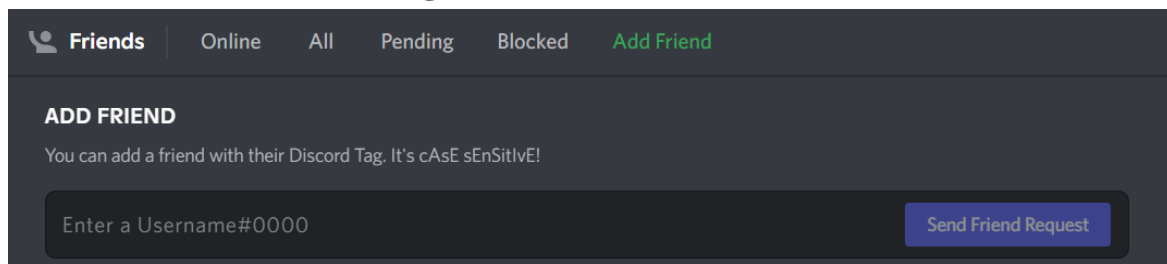


<sup>7</sup> Status

## درخواست دوستی (Friendship Request)

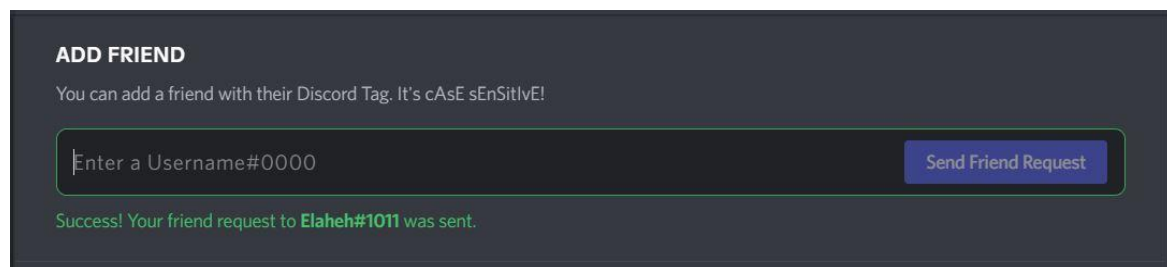
در این بخش کاربر باید درخواست دوستی ارسال یا دریافت کند. درخواست دوستی ارسالی می‌تواند لغو گردد و درخواست دوستی دریافتی می‌تواند قبول یا رد شود.

با کلیک بر روی گزینه Add Friend صفحه‌ی زیر نمایش داده می‌شود.



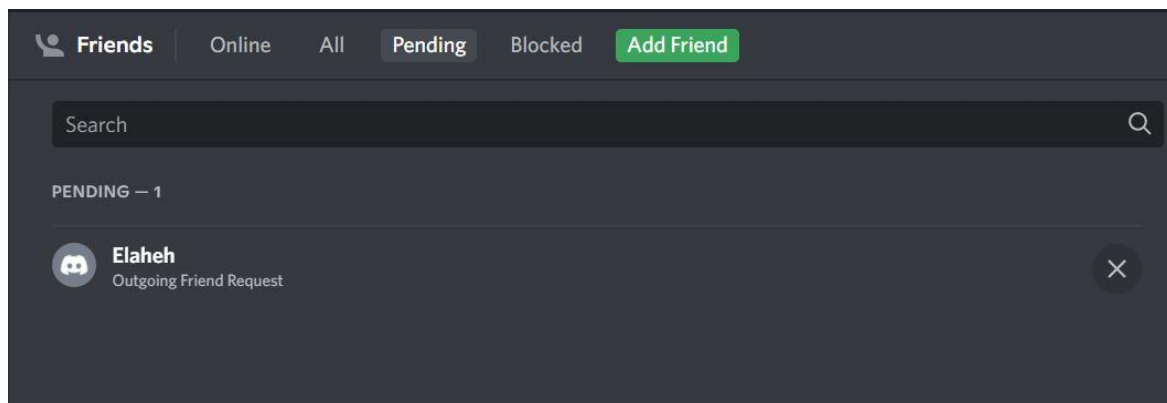
The screenshot shows the 'Add Friend' interface in Discord. At the top, there are tabs for 'Friends', 'Online', 'All', 'Pending', 'Blocked', and 'Add Friend'. Below the tabs, the text 'ADD FRIEND' is displayed, followed by the instruction 'You can add a friend with their Discord Tag. It's cAsE sEnSitivE!'. There is a text input field with the placeholder 'Enter a Username#0000' and a blue button labeled 'Send Friend Request'.

پس از این که کاربر نام کاربری مورد نظر را وارد کرد و بر روی گزینه Send Friend Request کلیک کرد، اگر نام نوشته شده درست باشد نتیجه به شکل زیر خواهد بود. جمله‌ی سبز رنگ نوشته شده باید با نام کاربری شخصی که درخواست به او ارسال شده است باید نمایش داده شود.



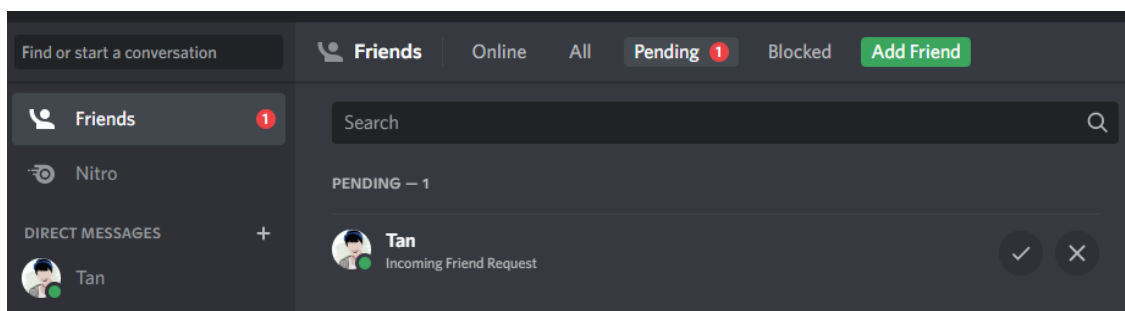
This screenshot shows the same 'Add Friend' interface as the previous one, but with a success message at the bottom: 'Success! Your friend request to Elaheh#1011 was sent.' The text input field now contains 'Elaheh#1011'.

پس از ثبت درخواست، در سمت کسی که درخواست را فرستاده است باید درخواست در قسمت Pending بخش لیست دوستان پیاده‌سازی گردید به شکل زیر نمایش داده شود و این شخص با کلیک بر ضربدر باید بتواند این درخواست را لغو کند.



The screenshot shows the 'Pending' section of the Discord interface. At the top, there are tabs for 'Friends', 'Online', 'All', 'Pending', 'Blocked', and 'Add Friend'. Below the tabs, there is a search bar. The text 'PENDING — 1' is displayed. Below this, there is a list item for 'Elaheh' with the status 'Outgoing Friend Request' and a red 'X' button to cancel the request.

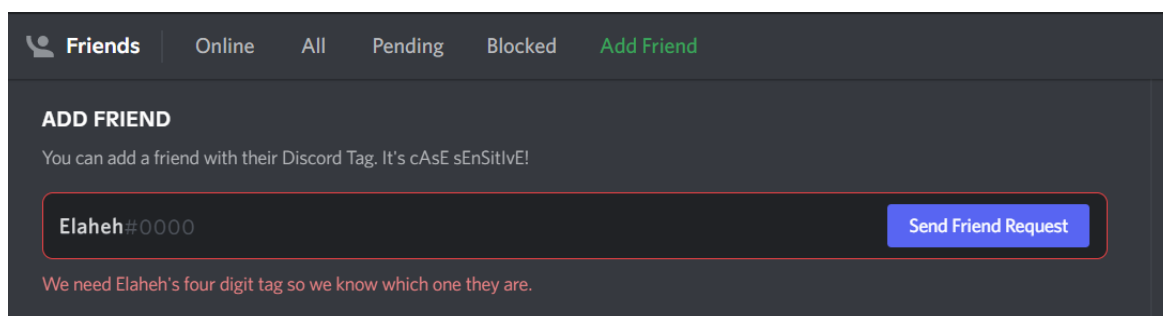
این درخواست به شخصی که درخواست را دریافت می‌کند به این صورت نمایش داده می‌شود:



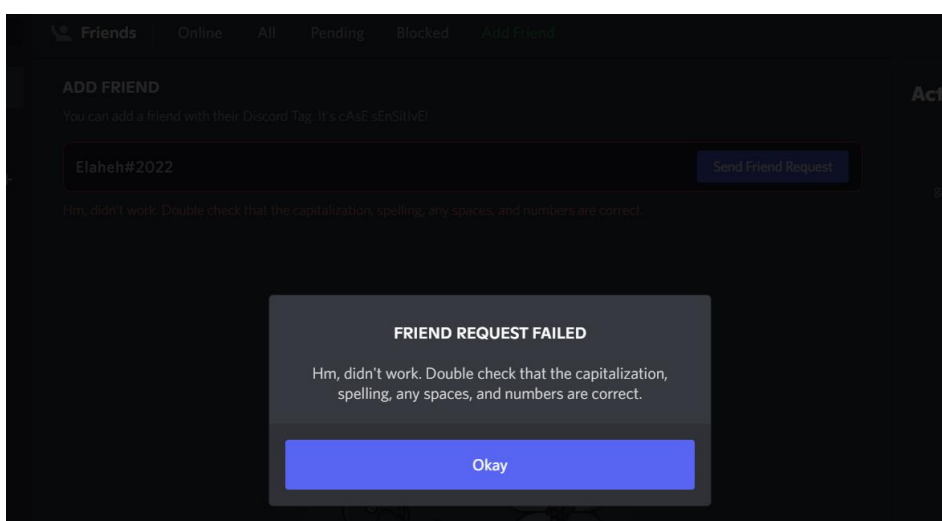
حال مخاطب می‌تواند درخواست را قبول یا رد کند. در صورتی که قبول کند، هر دو شخص به لیست دوستان یکدیگر اضافه خواهند شد.

در صورت اشتباه وارد کردن نام کاربری باید خطاهای مناسب به شکل زیر نشان داده شوند:

در صورتی که اعداد وارد نشوند خطای زیر باید نمایش داده شوند:

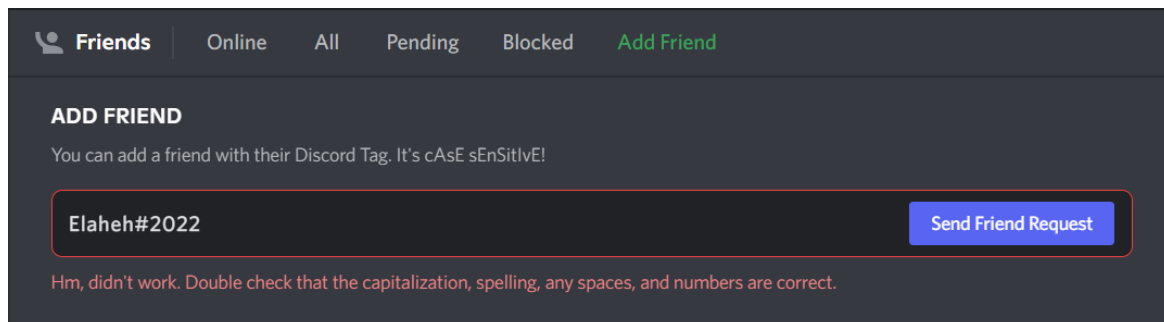


در صورتی که فرمت رعایت شده بود اما نام کاربری نامعتبر بود پیام زیر باید نمایش داده شود:



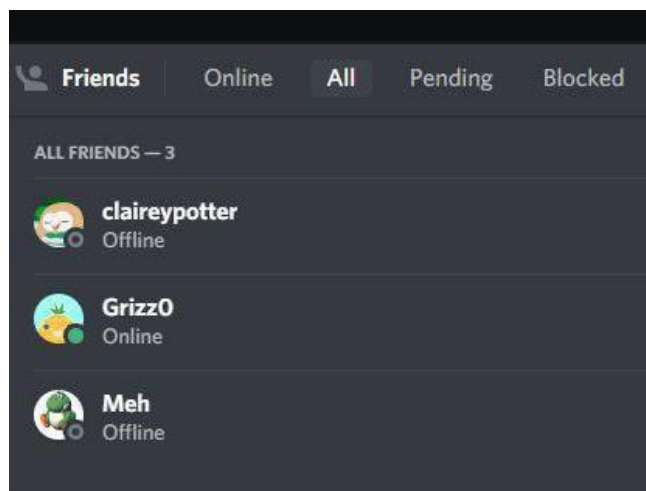


پس از کلیک بر روی Okay هم باید خطایی که در این عکس مشاهده می کنید نمایش داده شود:

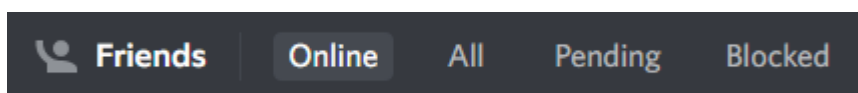


## لیست دوستان

در این صفحه باید لیستی از دوستان کاربر را در دسته بندی های مشخص شده نمایش دهید. لیست باید شامل نام کاربر، عکس پروفایل کاربری و وضعیت باشد.



لیست در ۴ حالت مجزا شامل Online، All، Pending و Blocked قابل نمایش است و با کلیک روی هر کدام، افراد مربوط به آن بخش و همچنین تعداد آن ها در بالای لیست نمایش داده می شوند.



توضیحات مربوط به افراد هر بخش به صورت زیر است:

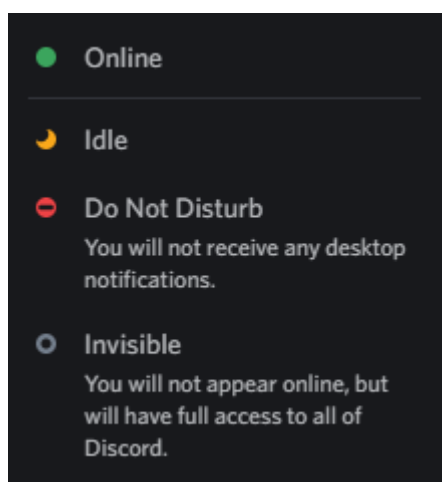
**Online:** دوستانی که وضعیت آن‌ها به صورت آنلاین یا idle است.

**All:** همه‌ی دوستان کاربر.

**Pending:** افرادی که به کاربر درخواست دوستی داده‌اند اما این درخواست هنوز قبول یا رد نشده است و در حالت pending قرار دارد.

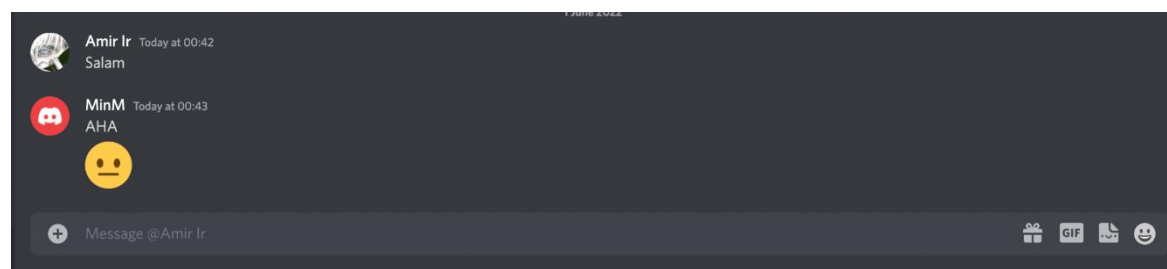
**Blocked:** افرادی که کاربر آن‌ها را مسدود کرده است.

لازم است که علاوه بر عکس کاربری، علامت مربوط به وضعیت کاربر را نیز نمایش دهید. برای مثال کاربر آنلاین یک دایره سبز کوچک برای نشان دادن وضعیتش دارد. نمونه این علائم برای هر وضعیت به شکل زیر است. (نیازی نیست دقیقاً از این علائم استفاده کنید و تصاویر مشابه نیز قابل قبول است)

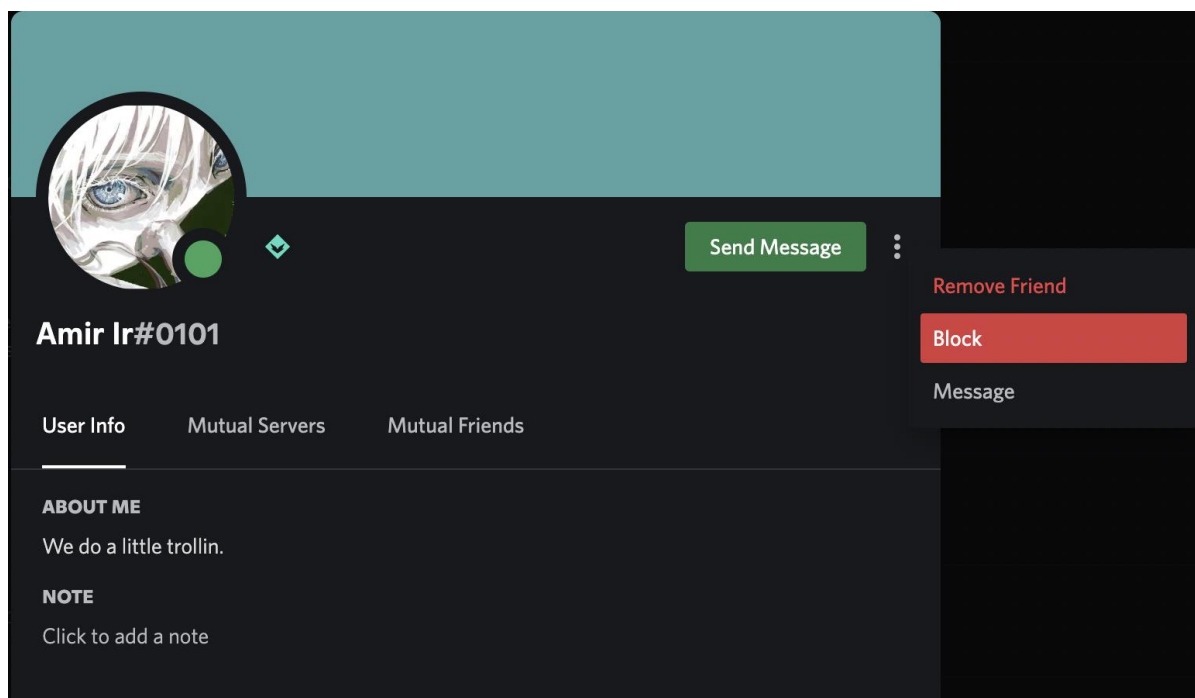


## چت خصوصی

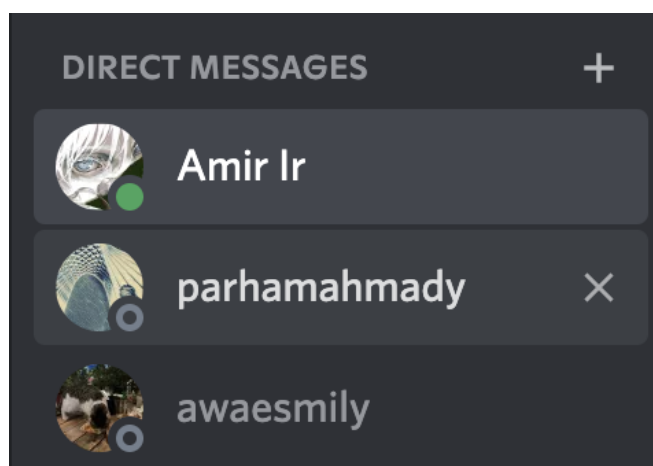
کاربران میتوانند به دوستان خود به صورت خصوصی پیام دهند. به صورتی که پیام مورد نظر خود را در Text Box مربوطه می‌نویسند و با فشردن دکمه ارسال این پیام برای هر دو طرف نمایش داده می‌شود.



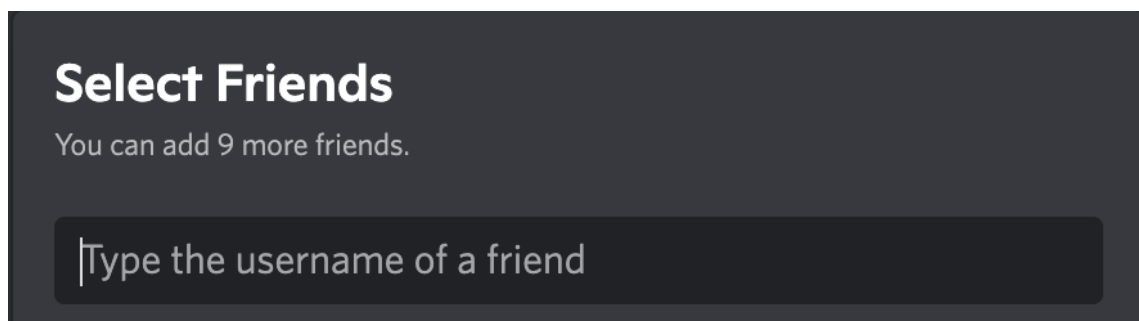
یک کاربر میتواند کاربر دیگر را مسدود<sup>8</sup> کند. در این صورت کاربر مسدود شده نمیتواند به او پیام داده و یا عکس پروفایل او را مشاهده کند.



همچنین بخشی برای ایجاد پیام خصوصی جدید و نمایش پیامهای خصوصی در صفحه اصلی باید وجود داشته باشد تا کاربران با استفاده از آن به پیامهای خود دسترسی داشته باشند.



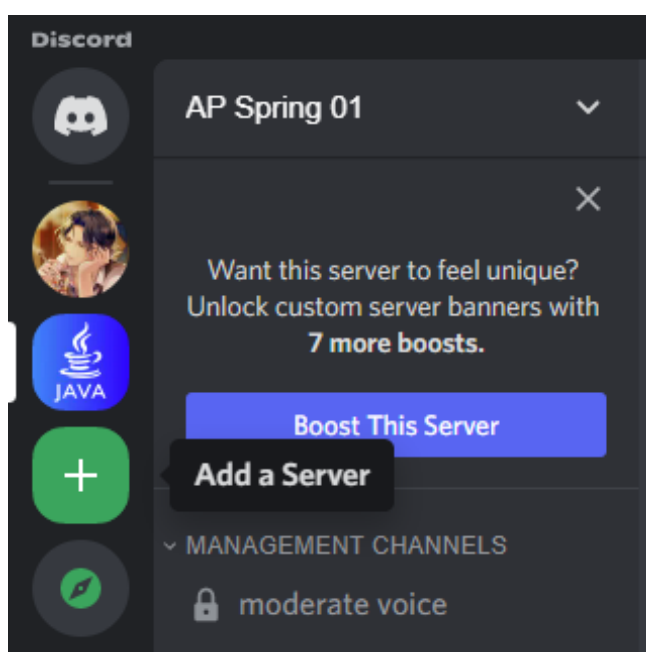
نحوه ایجاد پیام خصوصی جدید بدین صورت است که کاربر نام کاربری کاربر مورد نظر را در Text Box مربوطه نوشته و پس از کلیک بر دکمه ایجاد پیام به بخش پیام‌های خصوصی آن کاربر منتقل می‌شود.



سرور<sup>۹</sup>

## ساخت سرور (متفاوت با سرور backend)

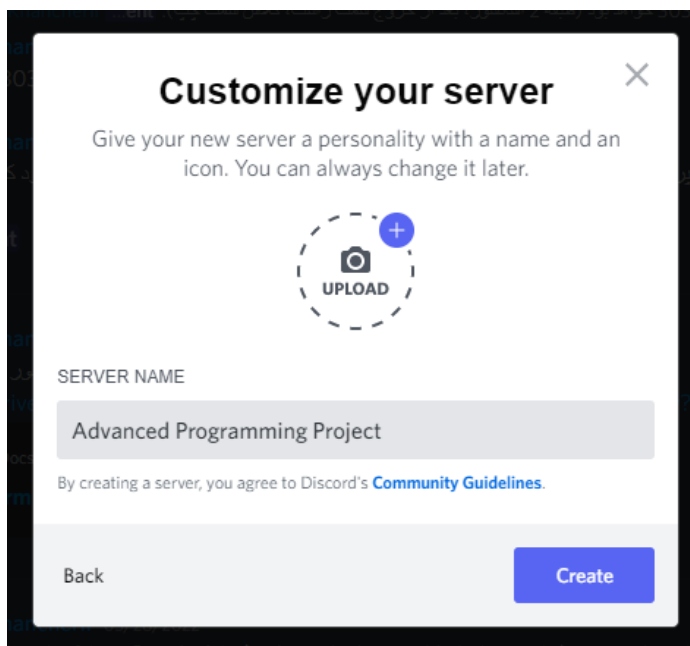
همان طور که در شکل زیر مشاهده می‌کنید، در نوار کناری سمت چپ برنامه باید لیستی از سرورهای موجود و یک دکمه برای افزودن سرور جدید وجود داشته باشد.



<sup>۹</sup> Server

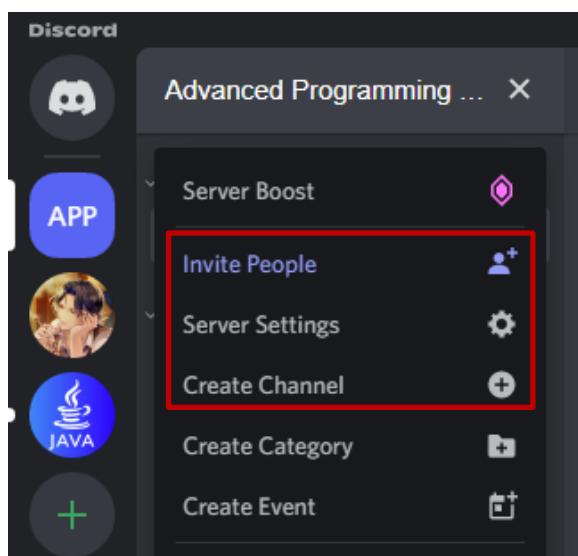
با فشردن این دکمه صفحه‌ای به کاربر نشان داده میشود که در آن نام سرور به عنوان ورودی دریافت شده، با فشردن دکمه‌ی create سرور ایجاد شده و به لیست سرورهای موجود کاربر اضافه میشود.

\* افزودن عکس برای سرور در این بخش الزامی نیست.



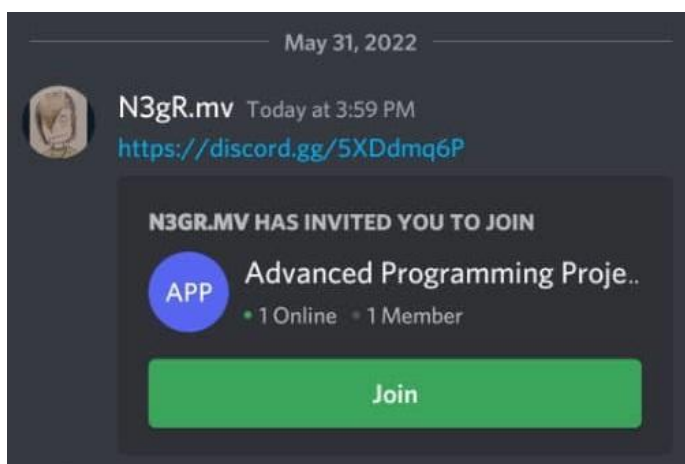
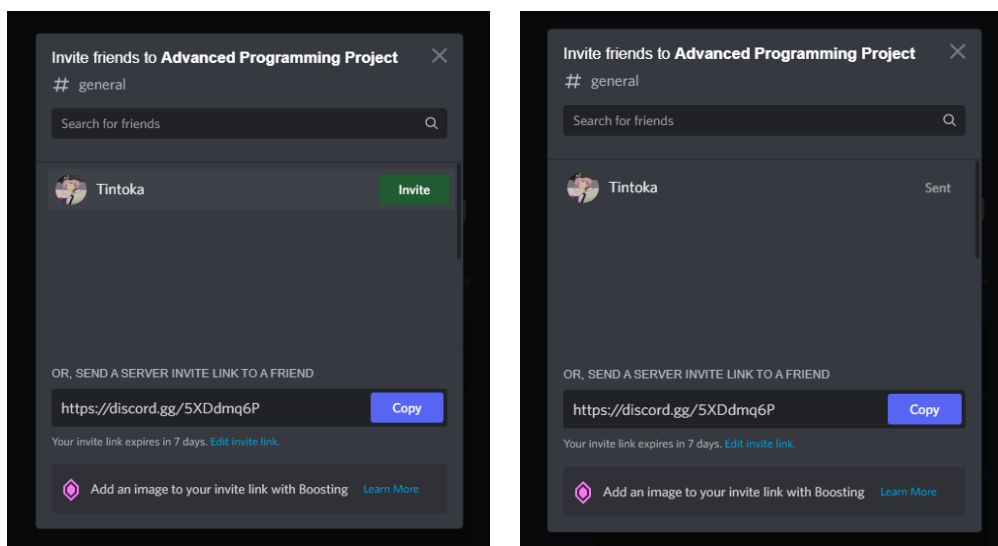
## تنظیمات سرور

پس از این که سرور ایجاد شد باید منویی مانند زیر برای آن وجود داشته باشد که از طریق بخش‌های مختلف این منو بتوان فردی را به سرور اضافه کرد (در صورتی که کاربر اجازه‌ی این کار را داشته باشد)، نام سرور را تغییر داد و یا کانال جدیدی به سرور اضافه کرد.

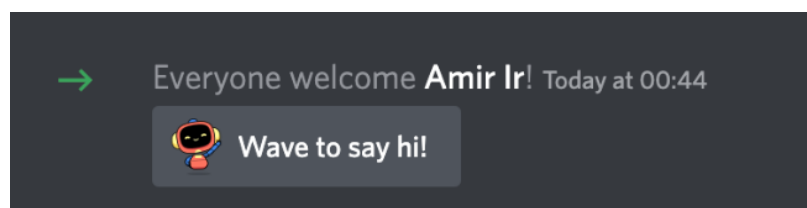


## افزودن کاربران به سرور

پس از این که قسمت Invite People در شکل بالا انتخاب شد صفحاتی مانند زیر به کاربر نشان داده میشود که در آن لیستی از دوستان نمایش داده شده و با استفاده از دکمه‌ای که در سمت راست نام آنها وجود دارد میتوان هر یک را به سرور اضافه کرد (شکل سمت چپ). پس از اضافه شدن کاربر به سرور این دکمه غیرفعال میشود (شکل سمت راست)

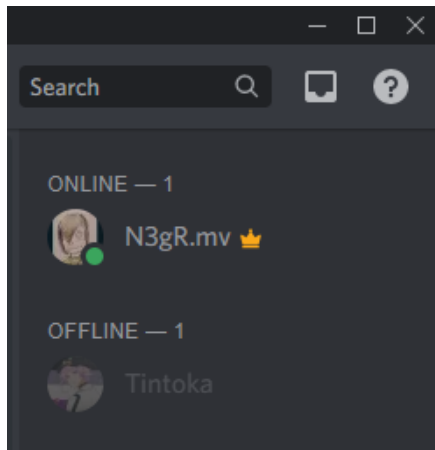


پس از قبول کردن درخواست و وارد شدن به سرور، متن خوش‌آمدگویی برای کاربر در کانال متنی ارسال می‌شود.



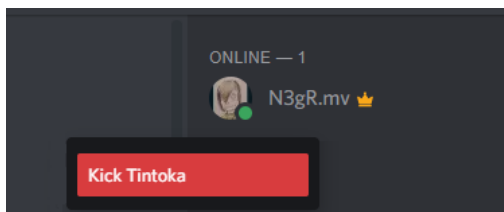
## نمایش کاربران موجود در سرور

مانند شکل زیر لیستی از تمام کاربران موجود در سرور به همراه وضعیت آنها باید در نوار سمت راست برنامه نمایش داده شود.



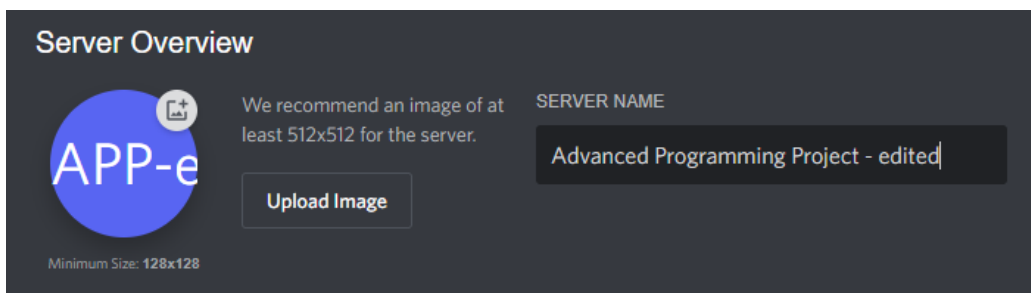
## حذف کردن کاربران از سرور

در قسمت قبل، که در آن لیستی از کاربران موجود در سرور نمایش داده میشود، باید این قابلیت وجود داشته باشد که با انتخاب هر یک از کاربران بتوان او را از سرور اخراج کرد (در صورتی که کاربر اجازه این کار را داشته باشد).



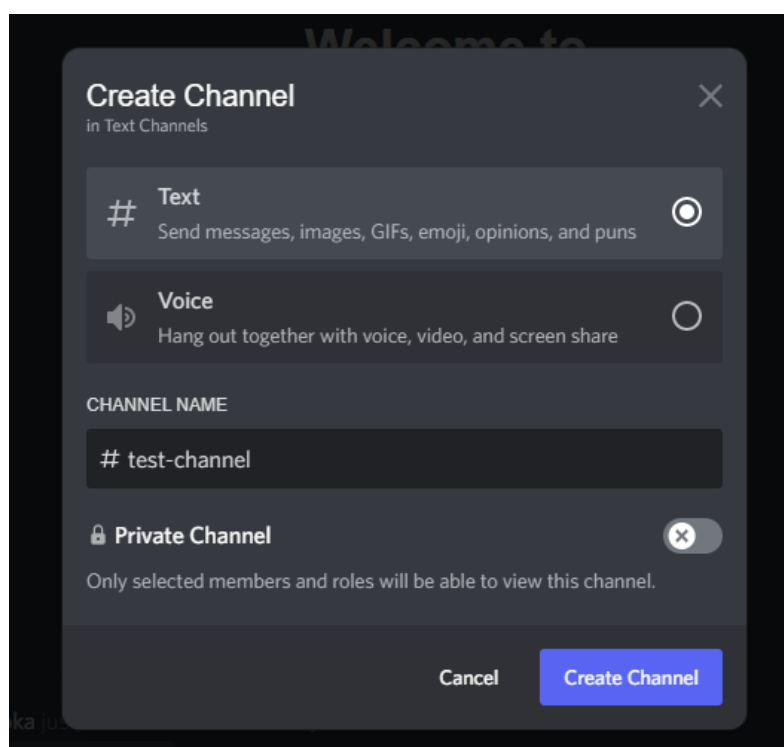
## تغییر نام سرور

همانطور که در فاز اول پروژه اشاره شد، پس از ایجاد سرور باید قسمتی برای تغییر نام آن وجود داشته باشد (در صورتی که کاربر اجازه این کار را داشته باشد). این کار با انتخاب قسمت Server Settings صورت میگیرد.

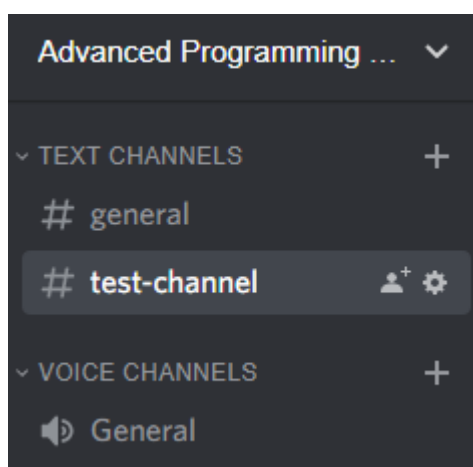


## افزودن کانال به سرور

با انتخاب گزینه **Create Channel** باید صفحه‌ای مانند زیر به کاربر (که اجازه این کار را دارد)، نشان داده شود که از طریق آن بتوان یک کانال جدید اضافه کرد. در این قسمت نوع کانال و نام آن مشخص شده و با فشردن دکمه **Create Channel** کانال جدید به لیست کانالهای موجود در سرور اضافه میشود.



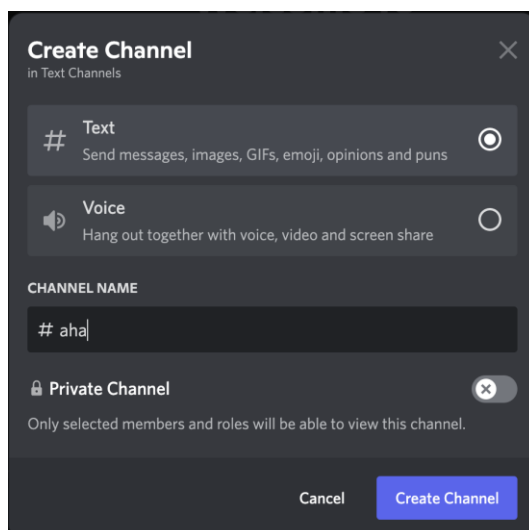
همانطور که در شکل زیر مشاهده می‌کنید، کانال جدیدی از نوع **Text Channel** به کانالهای موجود اضافه شده‌است.





## کانال<sup>۱۰</sup>

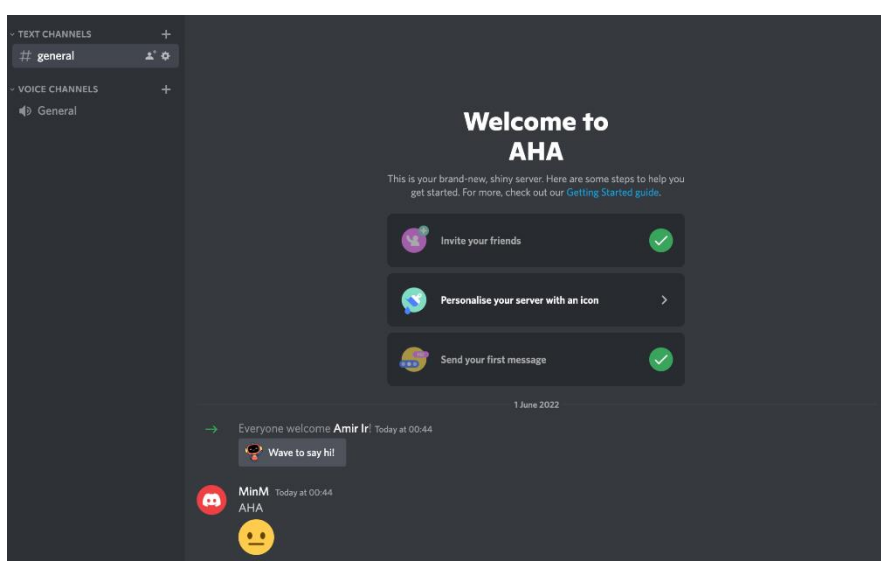
سازنده سرور و کسانی که قابلیت ساخت کانال در سرور را دارند، میتوانند یک یا چند کانال در سرور ایجاد کنند. هر کانال دارای دو مولفه نوع و اسم است.



## انواع کانال

- کانال متنی:

اعضای سرور می‌توانند در کانال متنی ایجاد شده به گفتگو بپردازند و پیام‌های خود را ارسال کنند.



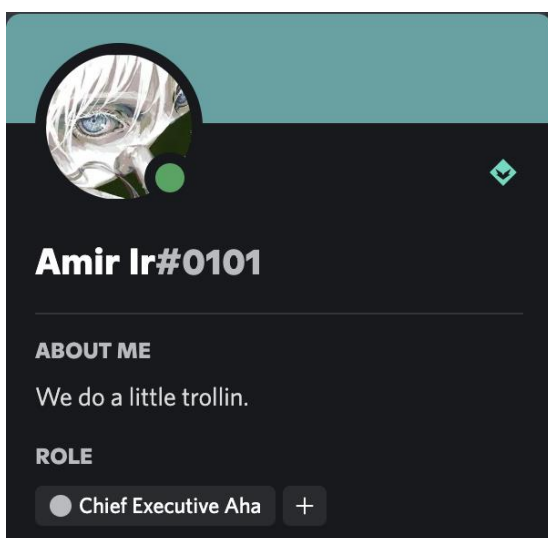
<sup>10</sup> Channel

## ● کانال صوتی:

توضیحات آن در بخش مربوط به کانال صوتی داده می‌شود.

## نقش

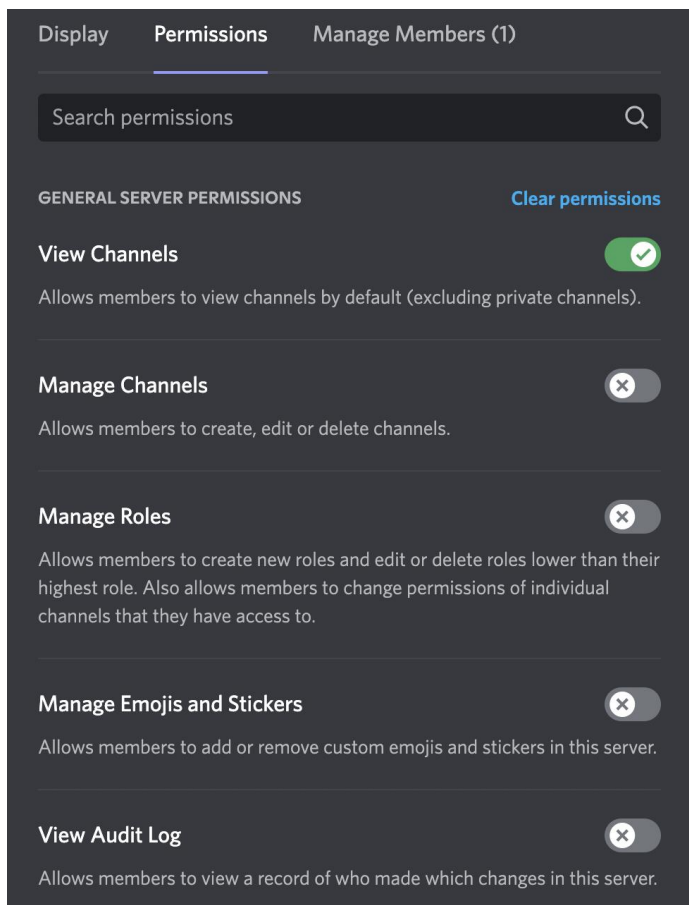
کاربر سازنده‌ی سرور میتواند یک نقش جدید بسازد و نام آن را تعیین کند. برای هر عضو سرور نیز باید بخشی برای اضافه کردن نقش به عضو وجود داشته باشد.



هر نقش میتواند یک یا چند مورد از دسترسی‌های زیر را داشته باشد:

۱. قابلیت ساخت کانال
۲. قابلیت حذف کانال
۳. قابلیت حذف یک عضو از سرور
۴. محدود کردن افرادی که به یک کانال دسترسی دارند
۵. منع کردن دسترسی کاربر به کل سرور (دسترسی تنها به تعدادی از کانالهای سرور)
۶. تغییر نام سرور
۷. قابلیت مشاهده تاریخچه ی چت
۸. قابلیت سنجاق کردن یک پیام در کانال متنی

برای هر یک از دسترسی های زیر در هنگام ایجاد نقش باید بخش مربوطه وجود داشته باشد و مشخص کند که نقش مورد نظر دسترسی فوق را دارد یا خیر. (مانند شکل زیر)



Display Permissions Manage Members (1)

Search permissions

GENERAL SERVER PERMISSIONS [Clear permissions](#)

**View Channels** ☒

Allows members to view channels by default (excluding private channels).

**Manage Channels** ☐

Allows members to create, edit or delete channels.

**Manage Roles** ☐

Allows members to create new roles and edit or delete roles lower than their highest role. Also allows members to change permissions of individual channels that they have access to.

**Manage Emojis and Stickers** ☐

Allows members to add or remove custom emojis and stickers in this server.

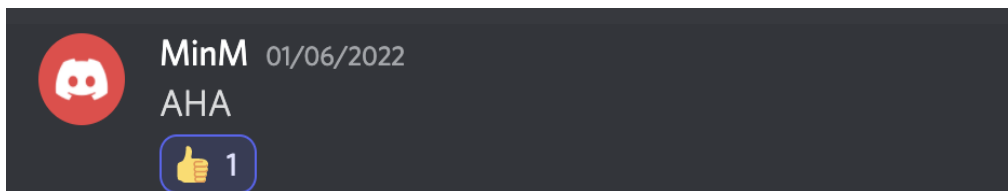
**View Audit Log** ☐

Allows members to view a record of who made which changes in this server.

همچنین تمامی اعضای سرور قابلیت اضافه کردن عضو به سرور را دارند و تنها کسی که قابلیت حذف سرور را دارد سازنده سرور است.

## واکنش

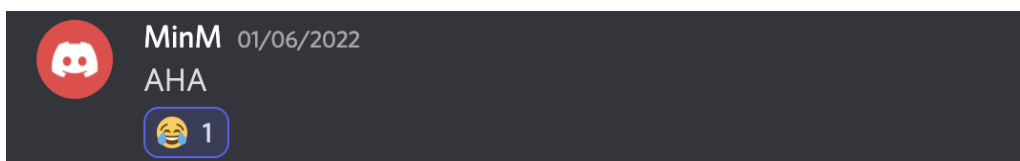
اعضای سرور می‌توانند به هر پیام در کانال متنی واکنش‌های  
۱. پسندیدم



۲. نپسندیدم



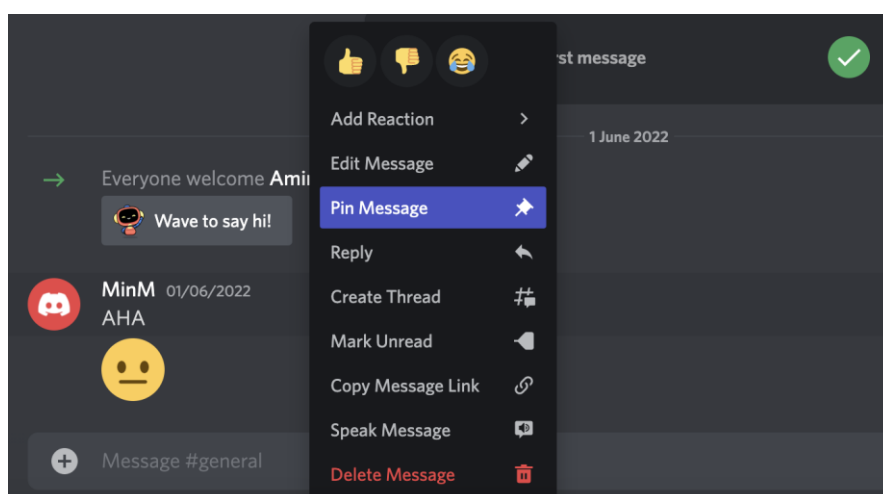
۳. خنده



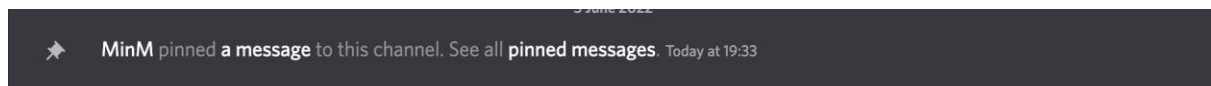
خود را نشان دهند. واکنش به پیام باید برای همه اعضای کانال قابل نمایش باشد.

## سنجاق کردن

ادمین و افرادی که قابلیت سنجاق کردن پیام را دارند می‌توانند یک پیام را در کانال متنی سنجاق کنند.

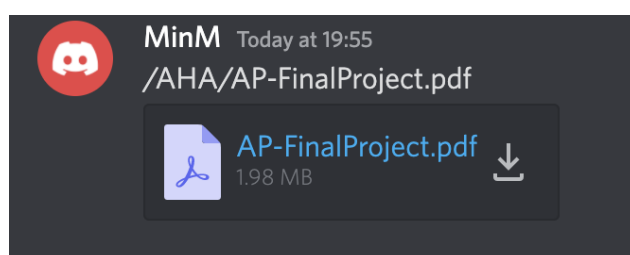


اعضای سرور باید بتوانند در هر لحظه که بخواهند پیام سنجاق شده را مشاهده کنند. مشاهده پیام سنجاق شده می‌تواند با فرستادن یک دستور انجام شود.



## ارسال فایل

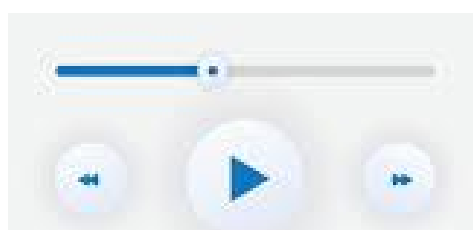
کاربران باید بتوانند از طریق کانال‌ها و یا چت خصوصی برای یکدیگر فایل ارسال کنند. در این صورت دیگر اعضا باید بتوانند در هر زمانی آن فایل را دانلود کنند (برای مثال از طریق لینکی که در اختیارشان قرار داده می‌شود). در زمان دریافت و ارسال فایل نباید مشکلی برای دیگر عملکردهای نرم‌افزار اتفاق بیفتد.



## موارد امتیازی

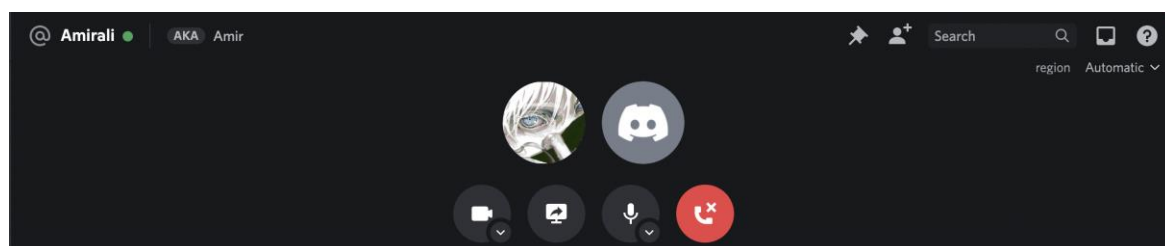
### پخش آهنگ (۱۰ امتیاز)

اعضای سرور می‌توانند به انتخاب خود یک آهنگ در سرور پخش کنند و آهنگ برای همه‌ی اعضای سرور همزمان پخش شود. قابلیت‌های کنترل آهنگ در حال پخش مانند تغییر وضعیت پخش (Pause، Stop) و (Resume) و تغییر ثانیه پخش (Seek) و همچنین رفتن به آهنگ بعدی و قبلی باید وجود داشته باشد.

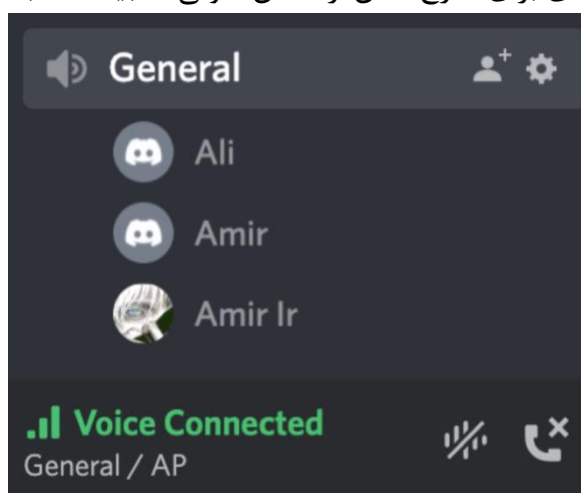


## برقراری تماس (۵ + ۵ امتیاز برای خصوصی و گروهی)

در این بخش باید در بخش چت خصوصی، گزینه‌ای برای شروع تماس دو نفره وجود داشته باشد. در هنگام تماس نیز عکس پروفایل و نام هر دو کاربر نیز نمایش داده شده و همچنین گزینه‌ای برای اتمام تماس وجود داشته باشد.



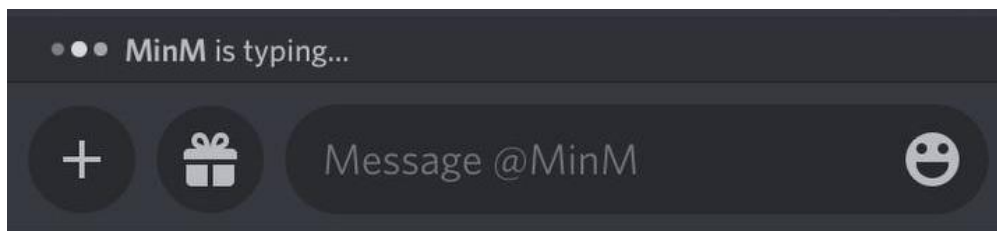
برای برقراری تماس چند نفره، باید در سرور کانال صوتی ایجاد شود و با وارد شدن هر کاربر به کانال صوتی، به تماس صوتی اضافه می‌شود. در این بخش نیز باید نام و عکس کاربران حاضر در تماس نمایش داده شده و همچنین گزینه‌ای برای خارج شدن از تماس صوتی، تعبیه شده باشد.



## نمایش وضعیت در حال نوشتن کاربران<sup>11</sup> (۵ امتیاز)

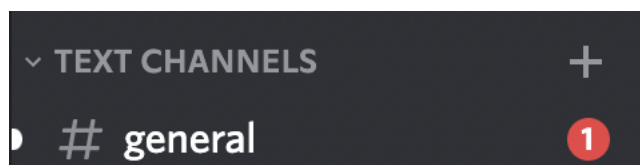
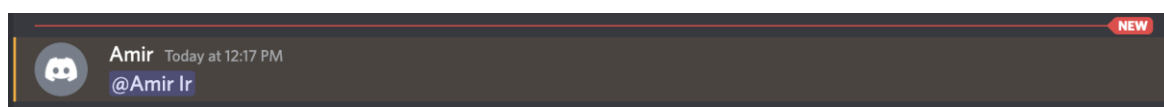
در صورتی که یک کاربر در حال نوشتن پیام باشد (در کانال‌ها و یا چت خصوصی)، پیام مناسبی برای کاربران دیگر نمایش داده شود.

<sup>11</sup> Is typing



ذکر کردن و نام بردن از یک کاربر<sup>12</sup> (۵ امتیاز)

کاربران بتوانند در پیام خود، از کاربران دیگری که در سرور عضو هستند نام ببرند. در این زمان کاربر نام برده شده باید با دریافت پیامی (notification) از آن اطلاع بیابند.



توجه داشته باشید که هرگونه خلاقیت اضافی، متناسب با پیچیدگی پیاده‌سازی آن، نمره امتیازی دریافت می‌کند.

<sup>12</sup> Mention