

به نام خدا

پروژه میان‌ترم 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-MidtermProject-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 تا دو روز پس از ددلاین، که با هر ساعت تاخیر یک درصد از سقف نمره قابل کسب، کسر می‌شود. برای مثال در صورت ارسال با ۱۰ ساعت تاخیر، سقف نمره قابل دریافت ۹۰ درصد است.

مهلت تحویل: تا یکشنبه ۱۵ خرداد 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 باید برای تمامی قسمت‌ها عملیات Exception Handling را برای استثنای 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)

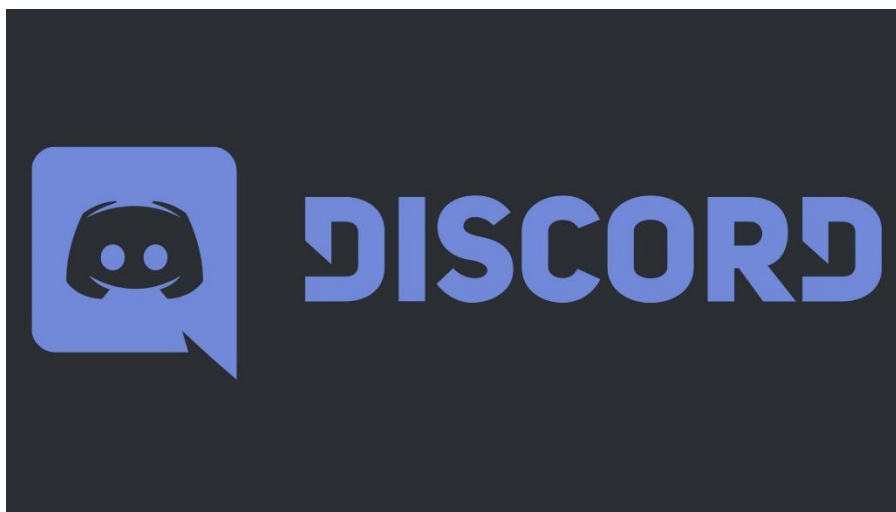
حداکثر نمره قابل کسب از بخش‌های امتیازی ۲۵ نمره است. در نتیجه در صورت کسب کامل نمره بخش‌های اصلی و بخش‌های امتیازی نمره ۱۲۵ را دریافت خواهید کرد. دقت کنید این بدان معنی است که اگر بخشی از نمره اصلی را از دست دهید، امکان رسیدن به حداکثر نمره ۱۲۵ وجود ندارد.

صفحه

فهرست

۴	دیسکورد.....
۵	حساب کاربری.....
۵	درخواست دوستی (Friend Request).....
۶	لیست دوستان.....
۶	چت خصوصی.....
۶	سرور.....
۷	کانال.....
۷	نقش.....
۹	موارد امتیازی.....
۱۰	نکات.....

دیسکورد^۱



دیسکورد یک نرم‌افزار برقراری تماس تلفنی در بستر شبکه برای سیستم عامل‌های ویندوز^۲، آی‌اواس^۳، اندروید^۴، لینوکس^۵ و مک‌اواس^۶ 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 (جزئیات آن در آینده به اطلاع شما می‌رسد).

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).

¹ Discord

² Windows

³ iOS

⁴ Android

⁵ Linux

⁶ macOS

حساب کاربری

- اطلاعات اجباری هر کاربر شامل نام کاربری و رمز عبور و ایمیل بوده و اطلاعات اختیاری شماره تلفن است.
 - برای ساخت یک حساب جدید، کاربر باید تمام اطلاعات لازم را وارد کند (در صورتی که سمت سرور مشکلی باشد، برای مثال درست نبودن فرمت، باید پیام مناسبی برگردانده شود. از پرتاب کردن استثناء برای این قسمت استفاده کنید).
 - توجه:** نام کاربری باید یکتا و حداقل شامل ۶ حرف و تنها شامل حروف و اعداد انگلیسی باشد.
 - توجه:** پسورد کاربران باید شامل حروف بزرگ و کوچک و اعداد بوده و حداقل ۸ کاراکتر داشته باشد.
 - توجه:** هنگام وارد کردن ایمیل، لازم است تا معتبر بودن الگوی آن بررسی شود (در صورت ورود شماره تلفن نیز لازم است الگوی آن بررسی شود).
 - هنگام ورود به حساب، کاربر باید ابتدا نام کاربری و رمز عبور خود را وارد کرده و در صورت صحت اطلاعات، وارد حساب خود می‌شود. در غیر این صورت سرور پیام مناسبی برمی‌گرداند.
 - کاربر می‌تواند یک عکس برای حساب خود بارگذاری کند (نیازی به نمایش این عکس در نرم‌افزار نیست و تنها باید به صورت مناسبی ذخیره شده تا در نسخه گرافیکی به درستی نمایش داده شود).
 - کاربر می‌تواند برای خود یک وضعیت^۷ تعریف کند:
 - Online ○
 - Idle ○
 - Do Not Disturb ○
 - Invisible ○
- به منظور بررسی صحیح بودن الگوی مقادیر وارد شده، بهتر است از «regex» استفاده کنید.

درخواست دوستی (Friendship Request)

- هر کاربر باید بتواند درخواست دوستی به دیگر کاربران بفرستد و یا از دیگران دریافت کند.
- این درخواست‌ها سمت سرور ذخیره می‌شوند و کاربران با دریافت لیست آن‌ها، می‌توانند آن را قبول و یا رد کنند.

⁷ Status

لیست دوستان

- در هر لحظه از برنامه، باید بتوان لیستی از نام کاربری تمام دوستان کاربر نمایش داد.
- در این لیست لازم است تا آنلاین/آفلاین بودن افراد مشخص شده باشد اما در صورتی که کاربر دوست یک وضعیت برای خود تعریف کرده باشد، وضعیتش نمایش داده می شود (در صورت آفلاین شدن کاربر به صورت خودکار وضعیت به آفلاین تغییر می کند).

چت خصوصی

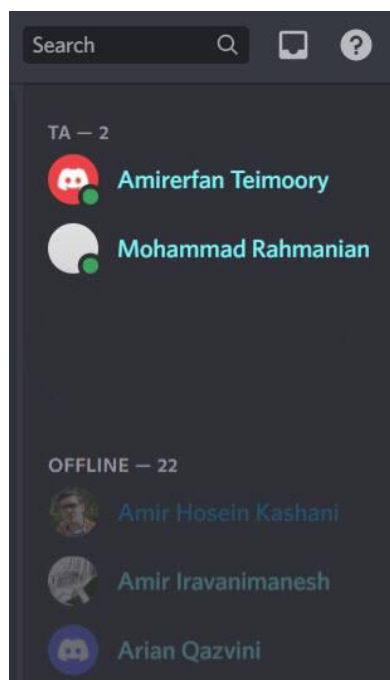
کاربران می توانند به دوستان خود به صورت خصوصی پیام دهند. یک کاربر می تواند کاربر دیگر را مسدود^۸ کند. در این صورت کاربر مسدود شده نمی تواند به او پیام دهد و یا عکس پروفایل او را مشاهده کند.

سرور^۹

دقت داشته باشید این سرور نوعی گروه بوده و با نرم افزاری که به عنوان سرور پیاده سازی می کنید متفاوت است. کاربر قابلیت ساخت یک سرور دارد که نام سرور هنگام ساخت آن باید مشخص شود و در آینده نیز قابل تغییر خواهد بود. سازنده ی سرور و کسانی که او مشخص می کند (در بخش نقش خواهید خواند) می توانند دوستان خود را به سرور اضافه کنند یا افراد موجود در سرور را حذف کنند. هنگامی که فرد جدیدی به سرور اضافه می شود، پیام خوش آمدگویی به آن کاربر فرستاده شود. سرور قابلیت ساخت چندین کانال با قابلیت های مختلف را در خود دارد (با کانال نیز در بخش مربوط به آن آشنا خواهید شد). هر عضو سرور باید بتواند وضعیت سایر آنلاین/آفلاین بودن کاربران سرور را مشاهده کند.

^۸ Block

^۹ Server



کانال^{۱۰}

سازنده‌ی سرور و کسانی که قابلیت ساخت کانال در سرور را دارند، می‌توانند یک یا چند کانال در سرور ایجاد کنند. کانال دارای مولفه‌های زیر است:

- اسم
- نوع

- کانال متنی: اعضای سرور می‌توانند در کانال متنی ایجاد شده به گفتگو بپردازند و پیام‌های خود را ارسال کنند.
- 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. هر نقش می‌تواند یک یا چند مورد از دسترسی‌های زیر را داشته باشد.

¹⁰ Channel

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^{۱۱} کردن دسترسی کاربر به کل سرور (دسترسی تنها به تعدادی از کانال‌های سرور)
 ۶. تغییر نام سرور
 ۷. قابلیت مشاهده تاریخچه‌ی چت
 ۸. قابلیت سنجاق^{۱۲} کردن یک پیام در کانال متنی
- همچنین تمامی اعضای سرور قابلیت اضافه کردن عضو به سرور را دارند و تنها کسی که قابلیت حذف سرور را دارد سازنده‌ی 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 تمامی 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.

¹¹ Ban

¹² Pin

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

اعضای سرور می‌توانند به انتخاب خود یک آهنگ در سرور پخش کنند و آهنگ برای همه‌ی اعضای سرور همزمان پخش شود (۵ امتیاز). قابلیت‌های کنترل آهنگ در حال پخش، در بخش گرافیکی پروژه صورت می‌گیرد.

برقراری تماس

کاربران امکان برقراری تماس دو نفره را داشته باشند (۱۰ امتیاز). برقراری تماس گروهی نیز از امتیاز بالاتری برخوردار است (۵ امتیاز اضافه‌تر). تماس دو نفره از طریق چت خصوصی و تماس گروهی از طریق ایجاد نوع جدیدی از کانال، به نام کانال صوتی صورت می‌گیرد.

نمایش وضعیت در حال نوشتن کاربران^{۱۳}

در صورتی که یک کاربر در حال نوشتن پیام باشد (در کانال‌ها و یا چت خصوصی)، پیام مناسبی برای کاربران دیگر نمایش داده شود (۵ امتیاز).

ذخیره اطلاعات در پایگاه‌داده

بجای استفاده از فایل، اطلاعات مورد نیاز نرم‌افزار در پایگاه‌داده ذخیره شود (۵ امتیاز).

¹³ Is typing

ذکر کردن و نام بردن از یک کاربر^{۱۴}

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 (۵ امتیاز).

پیاده سازی سرور با کمک RESTful API

پیاده سازی روشی جهت ارتباط بین سرور و کلاینت اجباری است. برای این بخش از ساکت و مباحثی که در درس یادگرفتید استفاده کنید. در عمل نوشتن یک روش غیراستاندارد جهت ارتباط بین سرور و کلاینت مرسوم نیست. روش REST یکی از روش های استاندارد جهت ارتباط بین سرور و کلاینت است که پیاده سازی api ها با این استاندارد نمره اضافه دارد. برای این بخش باید از پکیج های دیگری استفاده کنید که یادگیری آن به عهده خود دانشجو است (۱۰ نمره).

توجه داشته باشید که هرگونه خلاقیت اضافی، متناسب با پیچیدگی پیاده سازی آن، نمره امتیازی دریافت می کند.

نکات

- برای این پروژه شما باید دو نرم افزار جدا تحت عنوان سرور^{۱۵} و کلاینت^{۱۶} توسعه دهید.
- در این پروژه نیازی به پیاده سازی گرافیک نیست ولی باید رابط کاربری مناسبی جهت کار و اجرای نرم افزار ایجاد کنید. دقت کنید که با شیوه کدنویسی مناسب، باید جابه جایی بین اینترفیس کنسولی با گرافیکی نیاز به حداقل کدنویسی داشته باشد.
- باید اطلاعات مورد نیاز برای اجرای برنامه را در فایل و به صورت «binary» ذخیره کنید.
- کاربر باید بتواند رمز عبور خود را عوض کند.
- کاربر باید بتواند از حساب کاربری خود خارج شده، به حساب کاربری دیگری وارد شود.
- توصیه می شود به منظور نگهداری از هویت کاربر و کنترل سطح دسترسی او، از توکن ها مانند JWT^{۱۷} استفاده کنید.
- چند کلاینت باید بتوانند بدون مشکل و به طور همزمان، به سرور پیاده سازی شده شما متصل شوند و از خدمات آن استفاده کنند.

¹⁴ Mention

¹⁵ Server

¹⁶ Client

¹⁷ JSON web token