

سامانه آموزش مجازی

فهرست

2	<u>مقدمه</u>
2	<u>معرفی پروژه</u>
3	<u>بخش های پروژه</u>
3	<u>پنل ورود کاربران</u>
5	<u>صفحه اصلی (Homepage)</u>
6	<u>صفحه هر درس</u>
11	<u>پیام رسان سامانه</u>
12	<u>قوانین کلی</u>
13	<u>فاز اول</u>
13	<u>فاز دوم</u>
16	<u>نمره اضافه</u>
16	<u>نکات تکمیلی</u>
17	<u>مراجع کمکی</u>

● مقدمه

همونطور که می‌دونید پاندمی ویروس کرونا با گذر زمان (خداروشکر!) کم‌رنگ‌تر شده و روال آموزش حضوری دوباره از سر گرفته شده است. اما! هر لحظه ممکنه یک کشور محترم مثل چین (🇨🇳) دوباره ویروس جدیدی رو توزیع کنه: / هدف این پروژه هم اینه که اگر دانشجویها باز گیر همچین فلاکتی افتادن (زبونم لال!) از یک سامانه مطمئن و خوب مجازی برای ادامه درس ها و امتحانات و ارتباط با اساتید استفاده کنند تا خیالشون راحت باشه:)

فلذا از شما عزیزان خواستاریم تا پروژه زیر را پیاده سازی کنید!

● معرفی پروژه

هدف این پروژه پیاده سازی یک سامانه آموزش مجازی مبتنی بر زبان جاوا همراه با رابط گرافیکی، اتصال شبکه به صورت سرور کلاینت و اصول شی‌گرایی می‌باشد.

در این سامانه اساتید به سادگی می‌توانند درس مخصوص خود را ایجاد کرده و دانشجویان کلاس خود را به آن اضافه کنند و سپس انواع مطالب درسی، کوییزها، تمرین و امتحانات را با تنظیمات دلخواه خود قرار دهند تا دانشجویان از آنها استفاده کنند.

دانشجویان نیز میتوانند تمام دروسی که برایشان ایجاد شده است را مشاهده کرده، تمرینات را بارگذاری و مطالب مورد نیاز خود را بارگیری کنند و در صورت نیاز میتوانند از قابلیت پشتیبانی سامانه استفاده کرده و به مدیریت تیکت ارسال کنند و یا پیشنهادات و انتقادات خود را مطرح کنند.


از طرفی برای رهایی از دو راهی تلگرام یا واتس‌اپ! و ارتباط ساده بین اساتید و دانشجویان، سامانه به یک پیام‌رسان مخصوص خودش مجهز بوده که در ادامه به جزئیات آن پرداخته می‌شود.

در ادامه به توضیح مفصل هریک از بخش‌های سامانه همراه با جزئیات و قوانین مطلوب آن خواهیم پرداخت.

• بخش های پروژه

❖ معرفی پنل ورود کاربران:

در ابتدای اجرای برنامه توسط کاربر، پنل مخصوص کاربران جهت ورود، ثبت نام و یا پیگیری فراموشی رمز عبور نمایش داده می شود.

- در بخش ورود (login) کاربران تنها با شماره شناسه آموزشی خود (شامل شماره دانشجویی برای دانشجویان و کد آموزشی برای اساتید) به همراه رمز عبور خود میتوانند وارد سامانه شوند. دقت کنید که در این مرحله یک باکس جهت بررسی ربات نبودن کاربر به صورت  I'm NOT a robot نیز لازم بوده که در صورت چک نشدن آن از ورود کاربر به سامانه جلوگیری می شود.
- در بخش ثبت نام (sign up) یک کاربر جدید میتواند در سامانه ثبت نام کند. حداقل اطلاعات مورد نیاز برای ثبت نام در سامانه شامل نام، نام خانوادگی، نام کاربری، رشته تحصیلی، شناسه آموزشی، پست الکترونیک، تلفن همراه، نقش کاربر (شامل استاد یا دانشجو) و رمز عبور و تکرار رمز عبور می باشد. عکس پروفایل نیز به عنوان یکی از اطلاعات درخواستی به صورت آپشنال باید وجود داشته باشد اما انتخاب آن الزامی نیست. علاوه بر فیلدهای مذکور، شما می توانید به دلخواه اطلاعات اضافه تری را نیز برای ثبت نام در سامانه درخواست کنید.
- در بخش فراموشی رمز عبور، ابتدا لازم است تا کاربر پست الکترونیک خود را وارد و تایید کند، پس از آن شما باید از صحت وجود ایمیل وارد شده در سیستم اطمینان حاصل کنید، و سپس یک ایمیل پشتیبانی به پست الکترونیک کاربر شامل پیام مناسب همراه با یک کد عرقمی جهت اعتبارسنجی (verification) ارسال کنید. در مرحله بعد تا حداکثر زمان ۲ دقیقه باید برای وارد کردن کد ارسالی به کاربر زمان داده شود و پس از آن کاربر می تواند دوباره درخواست ارسال ایمیل مجدد کند (در زمان ۲ دقیقه امکان ارسال مجدد غیرفعال است). دقت کنید که در صورت درخواست ایمیل مجدد کد ارسالی قبلی باطل شده و کد جدید مدنظر است. همچنین توجه داشته باشید که این روال می تواند بی شمار تکرار شود و برنامه شما باید قابلیت ارسال ایمیل های پیاپی را داشته باشد. در صورتی که کد وارد شده توسط کاربر منطبق بر کد ارسالی باشد، کاربر می تواند دوباره رمز عبور خود را تنظیم کرده و پس از آن به صفحه ورود مرجوع می شود.

❖ قوانین پیاده سازی:

- شما باید برای هریک از فیلدهای مورد نظر وابسته به نوع آن صحت سنجی معتبری را انجام دهید.
- نام، نام خانوادگی و رشته تحصیلی باید فقط شامل حروف زبان انگلیسی و حداکثر ۱۵ کاراکتر باشند.
- نام کاربری و رمز عبور میتوانند شامل حروف انگلیسی کوچک، بزرگ و یا اعداد ۰ تا ۹ باشند.
- دقت کنید که نام کاربری حداقل شامل ۵ و رمز عبور حداقل شامل ۸ کاراکتر است و هر دو حداکثر می توانند ۱۲ کاراکتر باشند.
- توجه داشته باشید که اطلاعاتی از جمله شناسه آموزشی، پست الکترونیک، تلفن همراه و نام کاربری یکتا بوده و در صورت تکراری وارد شدن آن توسط کاربر فرم ثبت نام نامعتبر بوده و یک پیام مناسب به کاربر نمایش داده می شود.

- پست الکترونیک باید معتبر و به فرم xxx...xxx@yy...yy.zzz باشد که بخش اول میتواند شامل حروف انگلیسی کوچک، بزرگ، علامت نقطه و یا اعداد ۰ تا ۹ باشد (حداکثر ۱۵ کارکتر).
- بخش دوم تنها شامل حروف انگلیسی کوچک و علائم نقطه (.) و dash (-) است (حداکثر ۸ کارکتر).
- بخش سوم نیز فقط شامل حروف کوچک انگلیسی و حداکثر ۴ کارکتر است.
- شناسه آموزشی برای دانشجویان شامل یک کد ۱۰ رقمی و برای اساتید یک کد ۶ رقمی شامل اعداد ۰ تا ۹ می باشد.
- تلفن همراه نیز ۱۱ رقمی بوده و الزاماً با ۰۹ شروع می شوند.
- توجه کنید تمامی فیلدهای یاد شده به جز بارگذاری عکس الزامی هستند و هیچکدام با کادر خالی قابلیت سابمیت ندارند.
- در صورت بارگذاری عکس تنها پسوند های (jpg, .jpeg, .png) قابل قبول هستند و باقی تنظیمات تصویر را به دلخواه خود قرار دهید.

نمونه فرم مطلوب:

Login

Education ID:

Password:

☐ I'm not a robot

Submit

[forgot password?](#)

❖ معرفی صفحه اصلی:

پس از ورود کاربر، این صفحه (homepage) به اون نشان داده می‌شود. در این صفحه درس‌هایی که استاد و یا دانشجو در آن عضویت دارند به ترتیب چیده شده‌اند. علاوه بر آن ساعت و تاریخ نیز در بخشی از صفحه به کاربر نمایش داده می‌شوند.

همچنین پروفایل کاربر در محل مناسبی همراه با نام کاربری او نمایش داده می‌شود.

از طرفی کاربر باید بتواند اتفاقات (event) مختلفی که تا مدت زمان ۲ هفته رخ خواهند داد مانند ددلاین تمارین، کوپیزها و ... را مشاهده کند و با کلیک روی هر کدام به صفحه مخصوص آن ارجاع داده شود.

دقت کنید که در این صفحه شما باید به سلیقه خودتان یک منو شامل آپشن‌هایی از جمله: پروفایل، پیام‌رسان یا messenger و خروج از سامانه داشته باشید.

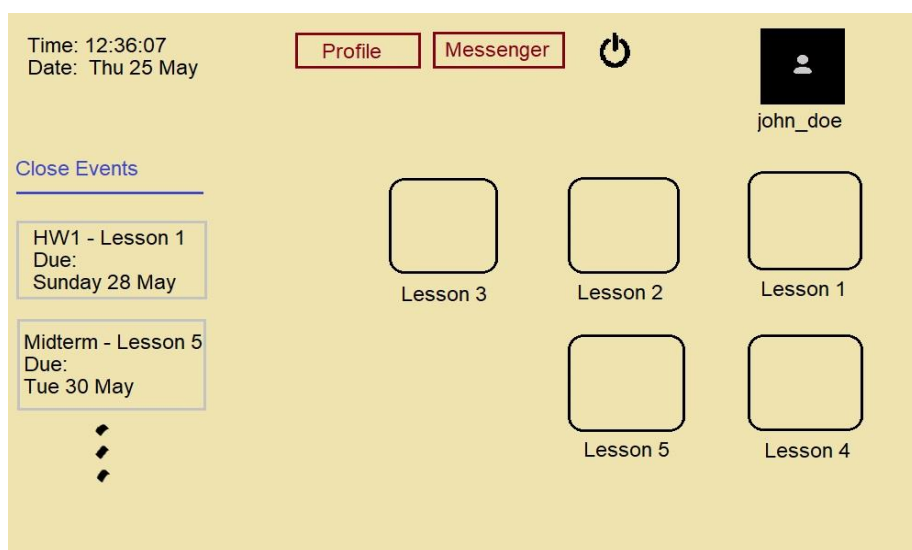
- گزینه پروفایل پنل جدیدی جهت ویرایش اطلاعات شخصی از جمله عکس پروفایل و ... باز میکند.
- گزینه پیام‌رسان یک پنجره کاملاً جدید که مربوط به شبکه اجتماعی سامانه است (مسنجر سامانه) را باز می‌کند (در ادامه توضیح داده شده است).
- گزینه خروج از سامانه باعث log out کاربر شده و به صفحه ورود باز می‌گردد.

با کلیک بر روی هر یک از درس‌ها صفحه مربوط به آن درس نمایش داده می‌شود که در بخش بعدی توضیح داده می‌شود.

❖ قوانین پیاده سازی:

نمایش ساعت باید به گونه‌ای باشد که علاوه بر ساعت و دقیقه، ثانیه شمار را نمایش دهد و مانند تایمر به صورت دائم در هر ۱ ثانیه آپدیت شود.

باقی قسمت‌های مربوطه از جمله نمایش‌های گرافیکی و ... را با سلیقه خود پیاده سازی کنید.



- مثالی از یک Homepage:

❖ معرفی هر صفحه درس:

در این صفحه اساتید و دانشجویان هر دو می توانند مطالب درسی بارگذاری شده، تمارین و یا امتحانات تعبیه شده یا منقضی شده را مشاهده کنند. دقت کنید که به طور کلی هر صفحه درسی شامل ۴ بخش کلی می شود که عبارتند از اعلانات، مطالب درسی، تمرین ها و امتحانات (کوییزها نیز امتحان محسوب می شوند). در هر بخش، مطالب مربوطه (مثلا تمرینات در بخش تمارین) قرار میگیرند، مستقل از اینکه ددلاین آنها گذشته و یا هنوز نرسیده است.

نمونه ای از صفحه یک درس:

اعلانات:

اطلاعیه برگزاری کلاس ها

مطالب درسی:

season1.pdf

تمرینات:

تمرین ۱

تمرین ۲

امتحانات:

Exam 1

Quiz 2

❖ قوانین پیاده سازی:

شما می‌توانید نحوه نمایش اعلانات و مطالب درسی را به سلیقه خودتان پیاده کنید، به عنوان مثال می‌توانید در یک پنجره جداگانه آن را نشان دهید و یا به صورت دائلود فایل پیاده کنید و یا اینکه به صفحه دیگری با قابلیت بازگشت بروید.

با کلیک روی هر تمرین باید به صفحه ای دیگر رفت به طوری که در آن اطلاعات تمرین از جمله عنوان، زمان باقیمانده و وضعیت تحویل نشان داده شود و در صورتی که از مهلت مقرر آن گذشته باشد، تعداد روز های گذشته از تمرین و در صورتی که هنوز برای انجام آن زمان باقی است، زمان باقی مانده را بر اساس تعداد روز، ساعت و دقیقه نشان دهید و امکان آپلود تمرین را برای انواع فایل های متنی، تصویری و زیپ فراهم کنید.

با کلیک بر روی هر نوع امتحان یا کوئیز باید به صفحه دیگری رفت به طوری که عنوان و وضعیت آزمون در آن نمایش داده می‌شود. در صورتی که آزمون تمام شده باشد وضعیت تصحیح آن شامل تصحیح نشده و یا نمره تعیین شده از سوی استاد نشان داده شود. همچنین در صورتی که آزمون قابلیت مرور داشته باشد باید این قابلیت را نیز پیاده کنید. در صورتی که موعد آزمون فرا نرسیده باشد می‌توانید تاریخ و ساعت شروع آزمون را نمایش دهید و در صورتی که در بازه زمانی امتحان مراجعه شود باید امکان شرکت در آزمون فراهم باشد.

(مخصوص اساتید)

دقت کنید که اساتید یک منوی مخصوص دارند که از طریق آن می‌توانند اعلانات، تمرین، آزمون ها و یا مطالب جدید قرار دهند، مطالب درسی و اعلانات می‌توانند هر نوع از فایل های متنی، صوتی، تصویری و یا عکس باشند و جز اینگونه فایل ها (... , .mp3 , .mkv , .mp4 , .png , .jpeg , .txt , .pdf)، بقیه قابل قبول نیستند.

- همچنین دقت کنید اساتید باید قابلیت اضافه کردن دانشجویان به درس را داشته باشند و این کار به دو روش زیر قابل انجام است:

الف) اضافه کردن دانشجویان از فایل اکسل (.xlsx) که ستون اول این فایل اکسل، شماره دانشجویی و ستون دوم نام و ستون سوم آن نام خانوادگی دانشجویان است.

ب) به صورت دستی و تک نفری (هر دانشجو دارای نام، نام خانوادگی و شماره دانشجویی است).

در هر دو حالت بالا دانشجویانی که به درس اضافه شده‌اند در صورت عدم عضویت به صورت خود کار در سامانه ثبت نام می‌شوند (برای فیلدهای دیگر فرم ثبت نام مقادیر پیش فرض یا بیهوده (default/dummy) در نظر بگیرید).

- بارگذاری تمرین: هر تمرین شامل یک عنوان، زمان تحویل (ددلاین) و صورت تمرین (به صورت یک فایل متنی، تصویر و یا زیپ) می‌باشد که توسط استاد تعیین می‌شود. همچنین استاد در این قسمت (بخش تمرین از منوی مخصوص اساتید) علاوه بر ایجاد تمرین جدید باید بتواند تاریخچه تمرین را مشاهده کند (شامل افرادی که تمرین را انجام داده‌اند و تاریخ پایان آزمون) و پاسخ های آپلود شده توسط دانشجویان را مشاهده و تصحیح کند. سایر تنظیمات را می‌توانید به سلیقه خود پیاده کنید.

- بارگذاری امتحان/کوئیز:

- استاد باید بتواند برای آزمون طراحی شده، زمان در نظر بگیرد. به این صورت که در یک تایم آزمون در اختیار کاربران قرارگیرد، مدت زمان آزمون مشخص باشد و پس از پایان زمان، آزمون از دسترس کاربران

خارج گردد. زمان و تاریخ شروع آزمون - مدت زمان آزمون - زمان و تاریخ اتمام آزمون باید توسط استاد تعیین گردد (منظور از زمان، زمان سیستم کامپیوتری شماست که با استفاده از کلاس های جاوا میتوانید به آسانی آن را به دست آورید).

- استاد درس میتواند آزمون را به نحوی تعیین کند که سوالات آزمون یکی یکی در صفحات جداگانه باشند و یا در یک صفحه کلی همه سوالات قرار داشته باشند.

- اساتید میتوانند ۳ نوع سوال تستی، تشریحی و یا صحیح/غلط طرح کنند. صورت سوال ها همگی متنی هستند اما سوالات تشریحی باید قابلیت آپلود انواع فایل های متنی، تصویری و زیپ را داشته باشند.

- استاد می تواند ترتیب سوالات را برای هر دانشجو به شیوه دلخواه تغییر دهد و یا ممکن است ترتیب سوالات برای همه دانشجویان یکسان باشد.

- استاد می تواند برای آزمون طراحی شده قابلیت مرور سوالات برای کاربران را قرار دهد. به این صورت که دانشجویان بتوانند سوالات و جواب های آزمون های گذشته ی خود را ببینند.

- استاد درس پس از پایان هر آزمون باید بتواند پاسخ های دانشجویانی که در آن شرکت کردند را مشاهده کرده و به آنها نمره دهد.

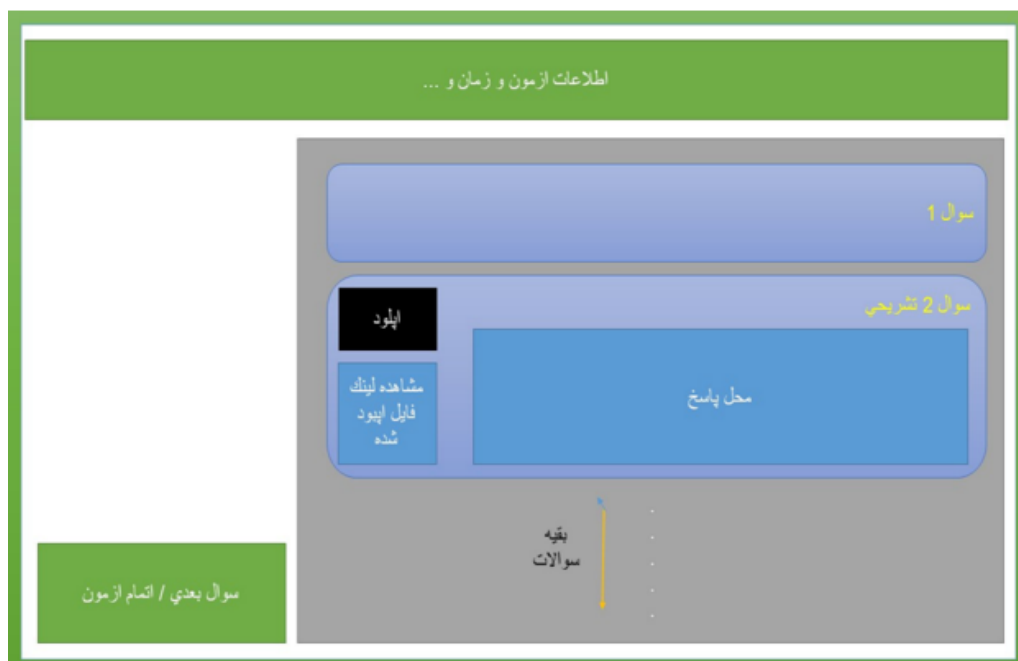
- پس از اتمام هر آزمون و تصحیح آن، استاد میتواند یک خروجی اکسل بگیرد که شامل همه نمرات دانشجویان به صورت فردی است (ستون اول فایل شماره دانشجویی، ستون دوم نمرات).

- همچنین در این بخش استاد میتواند تاریخچه سایر آزمون ها را نیز مشاهده کند که شامل شرکت کنندگان، تاریخ شروع آزمون و مدت زمان آن، پاسخ های دریافتی و خروجی فایل اکسل در صورتی که استاد اقدام به خروجی گرفتن کرده باشد، می باشد.

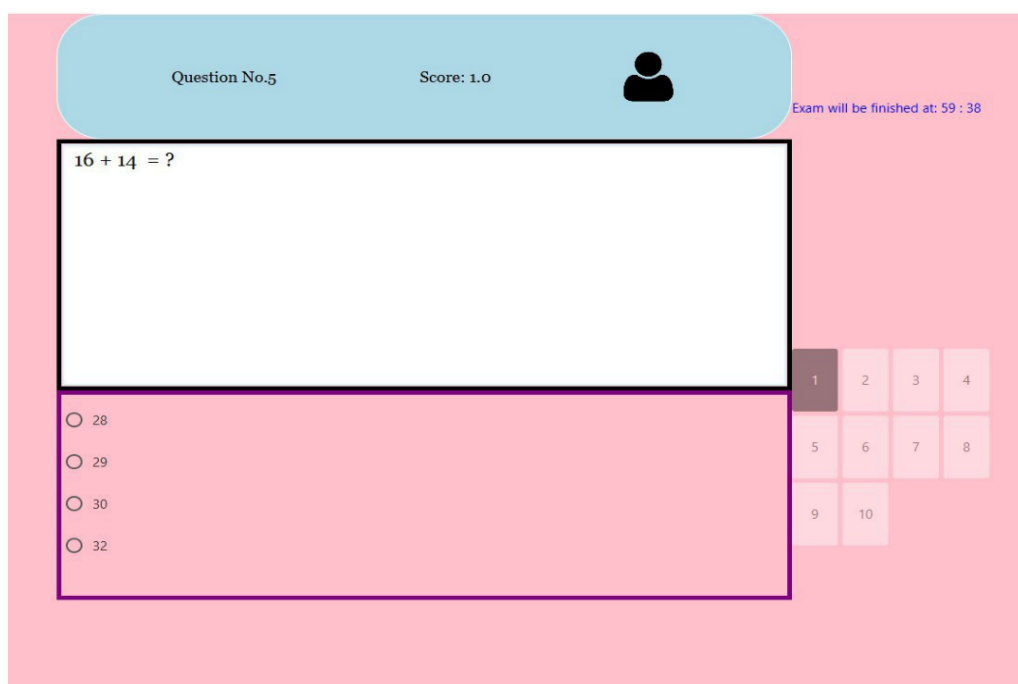
- نهایتا با ایجاد هریک از اعلانات، مطلب درسی، تمرین و یا آزمون جدید، موضوع ایجاد شده باید در اختیار دانشجویان و در صفحه عمومی درس قرار بگیرد و در صورتی که ددلاین آن در کمتر از ۲ هفته باشد باید در بخش رویداد های نزدیک در صفحه اصلی کاربران نیز نمایش داده شود.

* نمونه ای از لیست امتحانات:

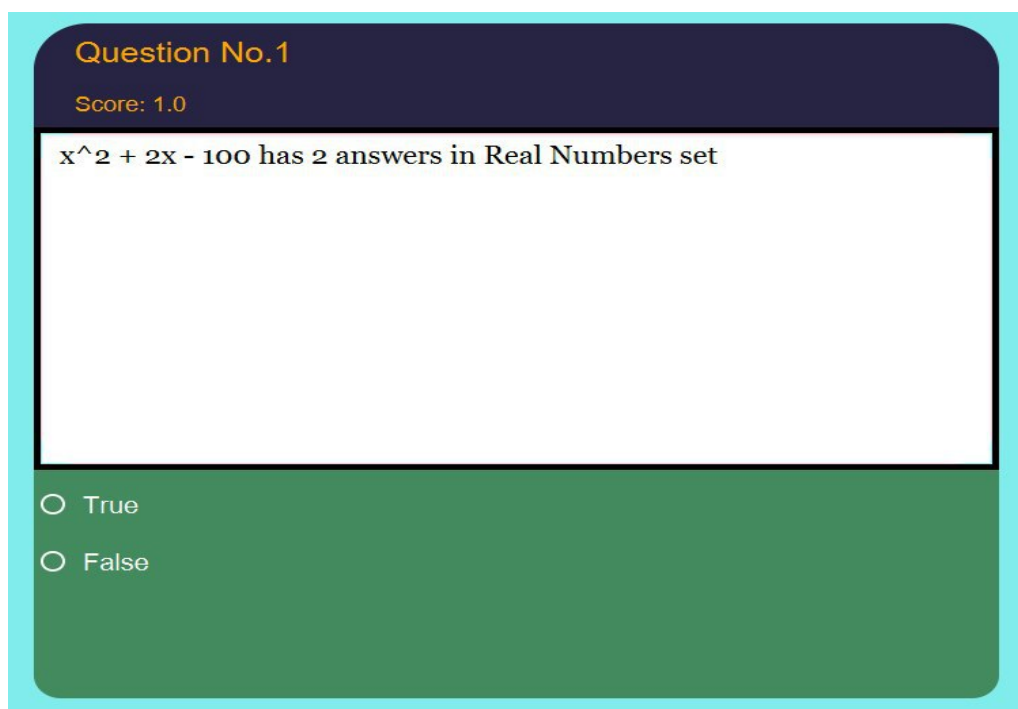
Exam Name	Date	Exam's Duration
second exam, RK051	2020-07-15	60.0 Minutes
first Exam, Mew	2020-07-21	60.0 Minutes
TestExam 1, Mew	2020-07-22	120.0 Minutes
TestExam 1, Mew	2020-07-22	120.0 Minutes
Exam No.8, Mew	2020-07-22	120.0 Minutes



* نمونه ای از یک سوال تشریحی:



* نمونه ای از یک سوال تستی در آزمون:



* نمونه ای از یک سوال صحیح/غلط:

(مخصوص دانشجویان)

* با مراجعه دانشجو به صفحه یک آزمون، باید نام آزمون و زمان آزمون و مشخصات آن در صفحه اطلاعات آزمون نمایش داده شود. در آن صفحه دکمه ای با عنوان شروع آزمون وجود دارد که پس از کلیک روی آن کاربر وارد آزمون شده و در صورت خروج از آزمون مجدداً فرصت شرکت در آن را ندارد.

* تذکر: قبل از شروع آزمون کاربر به سوالات دسترسی ندارد و با فرا رسیدن زمان شروع آزمون کاربر با کلیک روی شروع، وارد آزمون خواهد شد.

* پس از باز شدن صفحه آزمون، اطلاعات زیر برای سوالات امتحان وجود دارد:

الف- شماره و صورت سوال

ب- بارم سوال

د- مکان مناسب برای پاسخ به سوال (همانطور که تاکید شد در صورت تشریحی بودن باید مکانی برای آپلود عکس، فایل متنی و یا زیپ تا حداکثر ۵ فایل در نظر بگیرید).

* هنگام انتخاب فایل از سیستم باید فقط فایل هایی با فرمت های (pdf, .zip, .jpg, .png) را برای انتخاب نمایش دهید.

* دقت کنید پس از آپلود، لازم نیست فایل را در محل تعبیه شده برای پاسخ نمایش دهید فقط کافیست یک لینک محلی در مکان پاسخ ایجاد کنید که پس از کلیک روی آن، فایل مربوطه باز شود.

* زمان کل باقی مانده از امتحان را نیز باید به صورت یک تایمر که دقایق و ثانیه های باقیمانده را نشان می دهد در محل مناسبی از صفحه قرار دهید.

* دانشجویان باید بتوانند سوالات و جواب های آزمون های گذشته خود را ببینند. (توجه کنید که اگر از طرف استاد قابلیت مرور فعال نباشد، دانشجو نمی تواند سوالات و جواب های خود را ببیند).

* نهایتاً بعد از اتمام آزمون و پس از تصحیح استاد، دانشجویان میتوانند با مراجعه به صفحه امتحان نمره کل امتحان و رتبه خود را میان شرکت کنندگان مشاهده کنند. دقت کنید که ممکن است آزمون هنوز تصحیح نشده باشد در این صورت عبارت "در حال تصحیح" برای دانشجو نمایان است.

❖ معرفی پیام‌رسان سامانه:

سیستم آموزشی مجازی به یک پیام‌رسان مخصوص خودش مجهز است که در یک پنجره جداگانه و مشابه یک برنامه مستقل عمل می‌کند. در این پیام‌رسان برای هر کاربر به ازای هر صفحه درسی یک گروه وجود دارد که تمام اعضای آن درس عضو گروه هستند و می‌توانند با یکدیگر چت کنند.

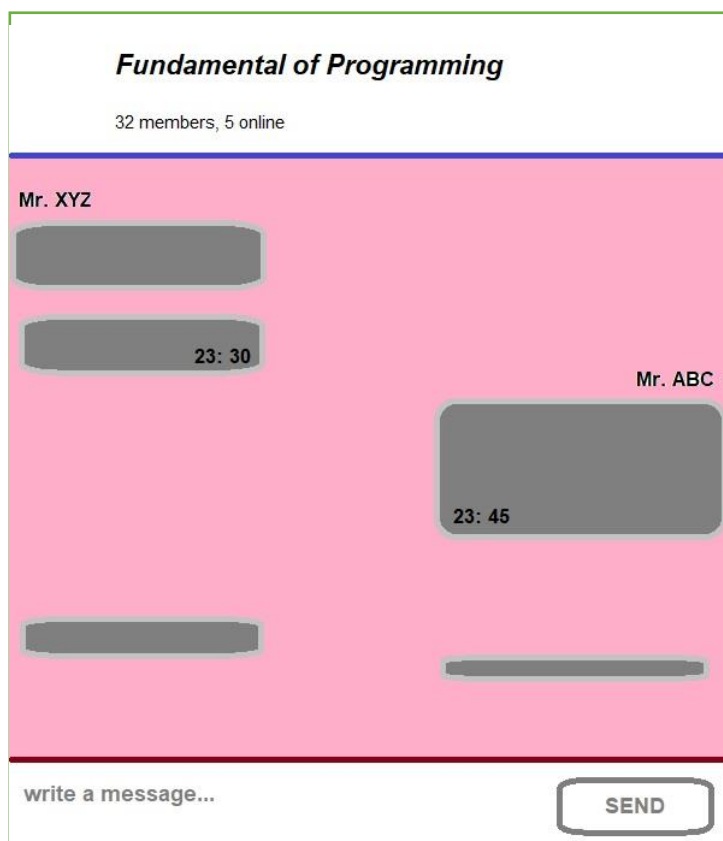
به عنوان مثال اگر یک نفر دارای ۳ درس مبانی برنامه نویسی، ریاضی ۱ و ادبیات باشد، آنگاه با باز کردن پیام‌رسان شاهد ۳ گروه با نام‌های همین دروس خواهد بود که در هر گروه استاد درس و باقی دانشجویان عضو هستند.

کاربران می‌توانند بین گروه‌های موجود در پیام‌رسان جابجا شده و در هر کدام مستقلاً پیام ارسال کرده و چت کنند. از جمله حداقل قابلیت مورد نیاز این پیام‌رسان قابلیت ارسال پیام‌های متنی، صوتی و تصویری علاوه بر فایل است. همچنین ادمین هر گروه استاد درس بوده که باید قابلیت‌هایی از جمله تنظیم نام گروه، اضافه و یا حذف کردن یک عضو گروه را داشته باشد. (موارد اضافه‌تر در بخش نمره اضافه قرار دارند)

❖ قوانین پیاده‌سازی:

دقت کنید که گروه به محض تشکیل صفحه درس توسط باید تشکیل و در پایگاه داده یا فایل شما ایجاد و ذخیره شود. همچنین با هر بار اضافه کردن یک دانشجو به کلاس به صورت خودکار او نیز عضو کلاس خواهد شد.

برای نمایش کاربران در پیام‌رسان نمایش نام کاربری آنها کافیست. لازم است همراه هر پیام ارسالی زمان ارسال را نیز قرار دهید. برای پیاده‌سازی این پیام‌رسان نیاز به استفاده از سرور-کلاینت و برنامه نویسی شبکه (socket programming) دارید که با جزئیات کامل در بخش قوانین کلی به آن پرداخته شده است.



* نمونه مثالی از یک گروه
در پیام‌رسان مورد نظر:

• قوانین کلی

- در این پروژه محدودیت کتابخانه گرافیکی وجود ندارد و می‌توانید از **Swing** یا **java fx** استفاده کنید.
- می‌توانید از ابزارهای کمکی (**drag and drop**) برای بهبود و سرعت بیشتر در ساخت بخش‌های مختلف گرافیکی استفاده کنید. (**SceneBuilder** و یا امکانات داخل **Netbeans**)
- دقت کنید که برنامه شما باید از قابلیت نخ‌ها و اجرا چندروندی (**multi-thread**) پشتیبانی کند و تمامی تسک‌های همزمان و موازی باید به این صورت پیاده شوند.
- در اجرا همروندی و چندنخی نیازمند استفاده از قابلیت **Synchronization** هستید تا بتوانید منابع و ورودی‌های مشترک بین چند نخ را به درستی مدیریت کنید.
- توجه کنید که برنامه شما باید به صورت سرور-کلاینت اجرا شود و باید این قابلیت را داشته باشد که به عنوان مثال چندین برنامه (کلاینت) به صورت زنده با هم تعامل داشته باشند. به عنوان مثال چندین نفر بتوانند همزمان در یک امتحان شرکت کنند و همزمان استاد در پیام‌رسان مربوطه پیام ارسال کند.
- اگر ۱۰ نمونه از برنامه شما اجرا شود، در هنگام چت، تغییرات باید برای همه این ۱۰ نفر به شرط عضویت در صفحه درس و گروه اعمال شود.
- برای پیاده‌سازی پروژه نیاز به استفاده از **web-socket** و **socket programming** دارید که در بخش مراجع کمکی آموزش‌های کاملی از این مباحث قرار دارد.
- دقت کنید در بخش پیام‌رسان شما باید برای سرور خود قابلیت **broadcast** کردن را پیاده کنید تا یک پیام مشابه همزمان برای چندین کلاینت ارسال شود.
- نیاز به ذخیره حجم قابل توجهی از اطلاعات دارید که بعد از پایان برنامه اعم از کلاینت یا سرور، اطلاعات قبلی از بین نرود.
- برای ذخیره‌سازی اطلاعات می‌توانید از فایل‌ها و انواع پایگاه داده‌ها (**MongoDB**، **MySql**) و ... استفاده کنید.
- در بخش لینک‌های کمکی می‌توانید آموزش‌های مربوط به مباحث مرتبط به هر دو فاز پروژه را در صورت نیاز پیدا کنید.

• فاز اول

در این فاز شما به پیاده سازی کامل برنامه خود روی سرور محلی (localhost یا همان کامپیوتر خودتان) می پردازید. در پایان این فاز برنامه شما باید بتواند به خوبی تمامی پیش نیاز های گفته شده را اجرایی کرده و به صورت کامل و جامه روی کامپیوتر خودتان اجرا شود.

• فاز دوم

NGROK

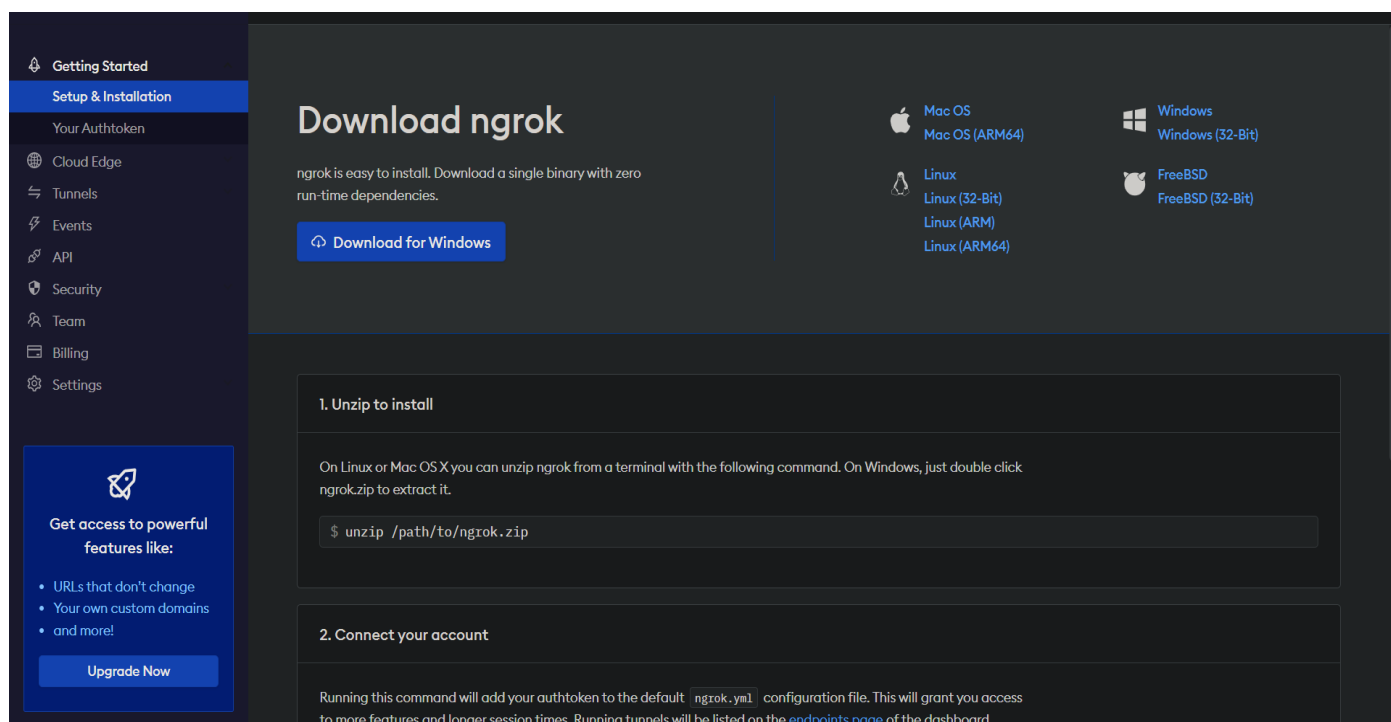
- چه نیازی به ngrok داریم واقعا؟!

احتمالا تا به حال کدهایی نوشته اید که با شبکه کار کنند و احتمالا تمام این کدها را روی شبکه محلی کامپیوتر خود اجرا کرده اید. شاید برایتان سوال باشد که چگونه می توان پس از اجرای کد، از طریق یک کامپیوتر دیگر و بر بستر اینترنت به آن برنامه درخواست ارسال کرد؟

اگر سرور خریداری کرده باشید و یا اینترنت ثابتی که در اختیار دارید، آی پی ثابت (Static IP) داشته باشد، می توانید به راحتی با بدست آوردن IP آن از طریق گوگل از طریق سرچ کردن عبارت **What is my IP** به راحتی از بیرون از طریق آن IP آدرس و پورت مشخصی که برنامه شما روی آن گوش می دهد، به آن دسترسی پیدا کنید. با این وجود، عموما ISP هایی که اینترنت ADSL ارائه میدهند، کاربران را پشت سیستمی به نام NAT قرار می دهند. به طور خلاصه NAT یا **Network Address Translation** باعث می شود که تعداد سیستم مجزا، همگی یک آی پی یکسان از بیرون پیدا کنند. یعنی عملا شما درون شبکه NAT یک آی پی خصوصی دارید که از بیرون قابل دسترس نیست و هر پیامی که می فرستید با عبور از NAT، یک آی پی عمومی خاص که مربوط به آن NAT می شود را پیدا کرده و از طریق آن پیام ارسال می شود. در نتیجه کاربران مختلفی که پشت NAT هستند، همگی از طریق یک آی پی ارتباط برقرار می کنند. در نتیجه این موضوع، سیستم های دیگر نمی توانند مستقیما شروع کننده ارتباط با سیستم شما باشند. در اصل همیشه سیستم شما شروع کننده ارتباط با بیرون است و پس از عبور پیام از NAT، براساس آی پی و پورت مبدا و مقصد، NAT متوجه می شود که جواب های بعدی که دریافت می شود را به کدام یک از کامپیوترهای شبکه درونی ارسال کند. با توجه به این موضوع برای این که بتوانید از بیرون به کامپیوتر خود وصل شوید، نیازمند ابزارهای جانبی هستید. ngrok یک ابزار رایگان است که این امکان را برای شما فراهم می کند. کارکرد ngrok به این صورت است که شما ارتباط اولیه را با آن برقرار می کنید و با برقرار شدن این ارتباط، ngrok می تواند به راحتی پیام ها را برای شما ارسال کند. حال ngrok نقش یک واسط را ایفا کرده و افراد دیگری که قصد اتصال به شما را داشته باشند، می توانند درخواست خود را به ngrok فرستاده و ngrok با توجه به برقرار بودن ارتباط، درخواست آنان را برای شما ارسال می کند.

- نحوه کار با ngrok

برای کار با ngrok لازم است به سایت آن یعنی **ngrok.com** بروید و با کلیک روی گزینه **Sign up** در سایت آن ثبت نام کنید. البته بدون ثبت نام هم می توان از آن استفاده کرد اما تنها پروتکل **HTTP** در دسترس خواهد بود و از آن جایی که به احتمال زیاد پروژه شما از طریق پروتکل **TCP** کار می کند، نیازمند ثبت نام در این سایت هستید. پس از ثبت نام با صفحه زیر مواجه می شوید:



در این صفحه متناسب با سیستم عامل خود، نسخه مربوط را دانلود کنید. پس از دانلود، با یک فایل فشرده رو به رو خواهید شد و پس از اکسترکت کردن آن، فایلی به نام **ngrok** در اختیار دارید (در سیستم عامل ویندوز این فایل پسوند **.exe** هم دارد) این فایل را در جایی اکسترکت کرده و سپس از طریق **CMD** (در ویندوز) یا ترمینال (در **MacOS** یا لینوکس) به آن پوشه رفته و دستور زیر را اجرا کنید:

```
./ngrok authtoken 1mtwmlwQQkDdKyE6lJaRAP0qEUw_4mDz91krJVKCNTGeNusht
```

توجه کنید که در سیستم عامل ویندوز این دستور به صورت زیر است:

```
.\ngrok.exe authtoken 1mtwmlwQQkDdKyE6lJaRAP0qEUw_4mDz91krJVKCNTGeNuht
```

توجه کنید که کد جلوی دستور بالا مثال است و این کد را باید براساس عبارتی که در سایت نمایش داده می شود وارد کنید. همچنین ممکن است در لینوکس نیاز به استفاده از **sudo** داشته باشید. با این کار اطلاعات کاربری شما در کامپیوترتان ذخیره می شود.

حال باید عملیات اصلی را انجام بدهید. فرض کنید پورتی که سرور برنامه شما روی آن اجرا شده و گوش می دهد، پورت ۱۲۳۴۵ باشد. برای برقراری ارتباط درست، دستور زیر را وارد کنید:

```
./ngrok tcp 12345
```

or

```
.\ngrok.exe tcp 12345
```

با اجرای این دستور ارتباط شما با سرورهای ngrok برقرار می شود و با صفحه ای مانند صفحه زیر رو به رو می شوید:

```
ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      online
Account             [REDACTED]
Version             2.3.40
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           tcp://2.tcp.ngrok.io:16344 -> localhost:12345

Connections         ttl    opn    rt1    rt5    p50    p90
                   0      0      0.00   0.00   0.00   0.00
```

مهم ترین داده موجود در این صفحه، قسمتی است که در تصویر بالا دور آن کادر قرمز کشیده شده است. برای این که بتوانید از بیرون (یک کامپیوتر دیگر) به سرور خود متصل شوید، لازم است که به جای آی پی محلی 127.0.0.1 که تا به حال به آن درخواست می زدید به این آدرس درخواست بزنید. یعنی در کلاینت شما، باید این آدرس را جایگزین کنید. به عنوان مثال، در مورد بالا باید سوکت سمت کلاینت شما به این شکل ساخته بشود:

```
socket = new Socket("2.tcp.ngrok.io", 16344);
```

با این کار به راحتی امکان ارتباط برقرار کردن با سرور خود را از بیرون خواهید داشت.

در کل تمام کاری که شما باید انجام بدهید، ساخت اکانت در ngrok ، اجرای درست ngrok و تنظیم آدرس سوکت سمت کلاینت متناسب با آدرس داده شده در ngrok است و نیازی به هیچ تغییر دیگری در کدهای خود نخواهید داشت.

● نمره اضافه

- استفاده از عبارات منظم (Regular Expressions) جهت صحت‌سنجی فرم های ورود و ثبت‌نام
- استفاده از کد کیچا برای اعتبارسنجی ربات نبودن در فرم ورود کاربر
- قابلیت ویرایش و حذف دانشجویان از صفحه درس توسط استاد
- قابلیت محروم کردن یک یا چند کاربر خاص از شرکت در یک آزمون و یا تمرین توسط استاد
- پیاده سازی چت خصوصی، ویرایش و حذف پیام، اضافه کردن اعضا، اموجی، استیکر و به طور کلی هرگونه بهبود گرافیکی و عملیاتی در قسمت پیام‌رسان و بخش های دیگر پروژه نمره اضافه قابل توجهی خواهد داشت.
- پیاده سازی پروژه مبتنی بر فریم‌ورک spring-boot در جاوا که مخصوص کار با شبکه می‌باشد.

● نکات تکمیلی

- ✓ استفاده مناسب از مفاهیم و امکانات برنامه نویسی شی گرا در پیاده سازی پروژه ضروری است.
- ✓ می‌توانید پروژه را در گروه های دو نفره انجام دهید. (پیشنهاد می‌شود حتی الامکان از انجام پروژه به صورت تک نفره خودداری کنید).
- ✓ در صورت انجام پروژه به شکل گروهی، باید در بستر گیت‌هاب پروژه خود را به صورت private ایجاد کرده و انجام دهید. دقت کنید که نهایتاً ممکن است لازم باشد تا سطح دسترسی پروژه را عمومی کنید.
- ✓ هر یک از افراد تیم باید به بخش هایی که خودش پیاده کرده است تسلط کافی داشته باشد.
- ✓ هرگونه عدم صداقت در انجام و تحویل پروژه باعث کسر کل نمره پروژه خواهد شد.
- ✓ فاز اول و دوم پروژه را در دو پوشه جدا به فرمت زیر در یک فایل زیپ قرار داده و در مهلت مقرر ارسال کنید. (ارسال توسط یکی از اعضای گروه کفایت)

$\begin{cases} \text{StudentName_StudentID_Phase1} \\ \text{StudentName_StudentID_Phase2} \end{cases} \Rightarrow \text{StudentName_StudentID_APFinal.zip}$

مهلت تحویل:

۱۰ تیر ۱۴۰۱

ساعت ۲۳:۵۹

• مراجع کمکی

- گرافیک پروژه

* آموزش کار با کتابخانه JavaFX و SceneBuilder :

منابع انگلیسی:

[JavaFX Java GUI Tutorial - 1 - Scene Builder](#)

[JavaFX Java GUI Tutorial - 2 - Working with Scene Builder](#)

[JavaFX Scene Builder Tutorial for Beginners](#)

منابع فارسی:

[\(قسمت ۱\) / \(قسمت ۲\) / \(قسمت ۳\)](#)

[آموزش اضافه کردن کتابخانه گرافیکی JFOENIX](#)

- برنامه نویسی شبکه

منابع انگلیسی:

[Java Socket Programming - Multiple Clients Chat](#)

Java Socket Programming: [Part 1](#) – [Part 2](#) – [Part 3](#) – [Part 4](#)

منابع فارسی:

[موضوع ترد و نخها](#)

[موضوع شبکه و سوکت پروگرامینگ](#)

[موضوع مباحث تکمیلی شبکه و هندل کردن چند کلاینت با ترد](#)

[آموزش اسپرینگ بوت + دانلود کدها](#) (نمره اضافه)

- راه اندازی NGROK (فاز دوم)