پروژه درس برنامه‌نویسی پیشرفته

بخش دوم

تابستان ۱۴۰۱ – دانشکده علوم ریاضی

دانشگاه صنعتی شریف

استاد درس:

دکتر مجتبی تفاق

تیم طراحی:

ایمان قدیمی - علی توسلی - سینا قاسمی‌نژاد - نیما کلیدری - مریم مقدس

معرفی پروژه:

در این پروژه شما قرار است سامانه‌ی EDU و بخشی از CW را بر بستر شبکه، با امکانات بیشتر تکمیل کنید.

اهداف پروژه:

- اتصال اجزای برنامه بر بستر شبکه
- پیروی از اصول Clean Code
- استفاده و آشنایی با مفهوم Thread
- کار با Resource و فایل
- آشنایی و کار با دیتابیس

رفع اشکال:

برای این بخش از پروژه یک جلسه توجیهی برگزار می‌شود که زمان آن متعاقبا اعلام خواهد شد. قبل از جلسه توجیهی، حتما داک را مطالعه کنید. همچنین شما می‌توانید برای رفع پرسش‌ها، اشکال‌ها و ابهام‌های خود از طریق گروه تلگرامی و کوئرا با تیم طراحان پروژه در ارتباط باشید.

نحوه تحویل:

- این بخش به صورت مجزا از بخش‌های دیگر ارزیابی و به صورت آنلاین تحویل گرفته می‌شود.
- هر بخش پروژه باید حداکثر تا تاریخی که به شما اعلام شده است در کوئرا آپلود شود. کدی که در زمان تحویل مورد ارزیابی قرار می‌گیرد، کد آپلود شده در کوئرا است. دقت کنید که ددلاین‌های اعلام شده قابل تغییر نیست، پس برنامه‌ریزی لازم را برای رساندن پروژه خود به ددلاین‌ها داشته باشید.
- دقت کنید که فقط اجرا شدن صحیح کد مدنظر نیست و از شما انتظار می‌رود که به صورت اصولی و با استفاده از مفاهیم تدریس‌شده در کلاس کد بزنید.
- بخش دیتابیس پروژه امتیازی بوده، **ولی برنامه شما باید قابلیت سیو و لود اجزای برنامه را داشته باشد.**

موارد غیرمجاز:

- عدم تسلط کافی بر کد پروژه
- شباهت بیش از حد دو یا چند پروژه
- واگذاری کامل یا بخشی از پروژه به شخصی دیگر

رخ دادن این اتفاق‌ها برای هیچ فردی قابل پذیرش نیست و در صورت بروز هر کدام از این اتفاق‌ها ممکن است هر تصمیمی در رابطه با ارزیابی فرد گرفته شود.

در صورتی که یکی از این اتفاق‌ها رخ داده باشد لازم است که افراد حتما دلیل این مساله را پیش از تشخیص توسط تیم درس اعلام کنند و دلیل این اتفاق را توضیح دهند. در این صورت فقط ارزیابی مربوط به بخش اعلام شده تحت تاثیر قرار خواهد گرفت.

در صورتی که مشاهده یکی از این اتفاق‌ها توسط تیم درس، پیش از اعلام فرد رخ دهد این تیم از فرد درخواست خواهد کرد که در این رابطه توضیح دهند و در صورت قابل قبول نبودن توضیح، فرد موردنظر موفق به گذراندن درس نخواهد شد.

نکات مهم:

- ممکن است در قسمت‌هایی از داک ابهاماتی وجود داشته باشد، این موارد به وسیله کامنت در همین داک جواب داده می‌شوند و قابل مشاهده هستند.
- در صورتی که درباره جزئیات قسمتی از پروژه توضیح داده نشده، می‌توانید از **خلاقیت** خود استفاده کنید و به روشی معقول و متناسب با موارد آموزش داده شده در کلاس آن را پیاده‌سازی کنید.

تعریف پروژه:

در این فاز، شما موظف هستید که پروژه‌تان را بر اساس قواعد برنامه‌نویسی شبکه، به شبکه متصل کنید و تمامی قابلیت‌های فاز قبل (نکات امتیازی مد نظر نیست!) را نیز در این فاز به صورت آنلاین پیاده‌سازی کنید. **به غیر از حالت آفلاین، تمامی قابلیت‌هایی که در ادامه می‌بینید، باید در بستر شبکه و طبق معماری کلاینت-سرور پیاده سازی شوند.** توضیحات امکانات اضافه و همچنین امکانات حالت آفلاین در ادامه آمده است.

ساختار شبکه پروژه:

- در این بخش از پروژه شما باید برنامه‌ای که در بخش‌های قبلی نوشته‌اید را بر روی شبکه پیاده‌سازی کنید و ساختار کلاینت-سرور مشابه تمرین 6 طراحی کنید و تمام نکاتی که در تمرین 6 و کلاس‌های تی‌ای در مورد قواعد برنامه نویسی شبکه گفته شده (هم برای پروژه و هم برای تمرین)، از جمله وجود مفهوم Authentication Token، Ping و عدم دسترسی مستقیم کلاینت‌ها به یک‌دیگر و ارتباط مجزای هر یک با سرور به نحوی که چند کلاینت به‌طور همزمان قابلیت اجرا و ورود به برنامه را داشته باشند را نیز باید رعایت کنید.
- توجه کنید که وجود دکمه‌ای برای رفرش یا همگام سازی برنامه و یا هرگونه عملکرد مشابه آن درون پروژه **مجاز نیست**. برنامه شما باید در هر لحظه به صورت خودکار اطلاعات را از سرور بخواند و درون کلاینت اجرا کند. بنابراین در صورت وجود یک تغییر در اطلاعات موجود از سوی کاربری دیگر، مثلا ثبت نمره یک دانشجو توسط استاد، این تغییر باید توسط کاربر دوم و در هنگام رفرش خودکار دیده شود. برای رفرش خودکار می‌توانید از Ping زدن استفاده کنید و سرور نباید تا زمانی که درخواستی از سوی کلاینت برای آن ارسال نشده است، اطلاعاتی برای هیچ کلاینتی بفرستد.
- انتظار می‌رود که با توجه به ساختار کلاینت-سرور، تمامی اطلاعات از فقط طریق سرور قابل دسترسی باشند و کلاینت به هیچ اطلاعاتی از دیگران دسترسی نداشته باشد. تنها دسترسی کلاینت، تعداد محدودی از اطلاعاتی است که برای حالت آفلاین نیاز دارد.
- توجه کنید که تنها باید اطلاعات لازم به کلاینت فرستاده شود. اگر می‌خواهید اطلاعات پروفایل یک کاربر دیگر را به کلاینت بفرستید، فرستادن اطلاعات محرمانه او، مانند رمز عبور، مجاز نمی‌باشد! پس دقت کنید که تنها اطلاعات مورد نیاز را به کلاینت ارسال کنید.

- هنگامی که اتصال به شبکه قطع می‌شود، بصورت خودکار باید به کلاینت اعلام شود و برنامه بدون بسته شدن، بصورت آفلاین قابلیت اجرا کردن برخی از قابلیت‌ها را داشته باشد. در ادامه توضیحات دقیق‌تری در رابطه با حالت آفلاین داده شده است.

صفحه انتخاب واحد:

قابلیت اضافه کردن، حذف و ویرایش دروس در فاز قبل برای معاون آموزشی تعریف شده بود. در این فاز برخی اطلاعات جدید باید هنگام اضافه شدن درس توسط معاون آموزشی به درس اضافه شود. در هنگام تعریف درس باید نام درس، استاد (اساتید) درس، دستیاران استاد درس، کد درس، پیشنیازی‌ها و همنیازی‌ها و مقطع و ظرفیت درس و همچنین زمان کلاس و امتحان، وارد و ثبت شوند. سپس در صفحه اصلی در سمت دانشجویان، منوی انتخاب واحد توسط دانشجو تعریف شود. این منو همیشه پنهان است بجز زمانی که برای انتخاب واحد توسط معاون آموزشی تعریف شده است.
برای پیاده‌سازی منوی انتخاب واحد باید سال ورود هر دانشجو را نیز ذخیره کنید. معاون آموزشی می‌تواند بر اساس چند فیلتر (سال ورودی دانشجو، مقطع دانشجو، ...) زمان انتخاب واحد دانشجویان مختلف را تعیین کند و هر دانشجو تنها در زمان مشخص‌شده برای خود و در صورتی که مجاز به ثبت نام بود، می‌تواند انتخاب واحد کند. کاربر باید بتواند در زمان‌هایی که توسط معاون دانشکده مشخص می‌شود، تعدادی واحد را اخذ کرده به طوری که شرط‌های زیر رعایت شوند.
در این صفحه 2 بخش خواهیم داشت. در یک بخش دانشجو می‌تواند یک دانشکده را انتخاب کند و تنها دروس همان دانشکده نمایش داده شود. دروسی که باید نشان داده شوند باید بتوانند حداقل بر اساس 3 مشخصه (مانند زمان امتحان، ترتیب الفبایی نام دروس و مقطع درس) مرتب شوند. در بخش دیگر، لیست دروس پیشنهادی‌ای که

دانشجو بهتر است بردارد، بر اساس این‌که پیشنیازی‌هایش را پاس کرده باشد، ظرفیت خالی داشته باشند و مقطع دانشجو با رشته همخوانی داشته باشد نشان داده شوند. در ادامه‌ی همین بخش، دروس نشاندارشده‌ی دانشجو نیز دیده شود.

در روبه‌روی هر درسی که دانشجو آن را اخذ نکرده است، 3 گزینه نشاندار کردن/بدون نشانه کردن، اخذ درس، و درخواست از معاون برای اخذ این درس وجود داشته باشد که با کلیک روی درخواست از معاون، درخواست اخذ درس با پیغامی مناسب برای معاون دانشکده دانشجو ارسال شود و در بخش پیغام‌های معاون ظاهر شود. معاون نیز باید در انتهای همان پیام، بتواند این درخواست را رد کند یا قبول کند و درس برای دانشجو اخذ گردد. همچنین برای دروسی که هنوز دانشجو اخذ نکرده است، باید 3 گزینه *نشاندار کردن/بدون نشان کردن*، *حذف* و *تغییر گروه* موجود باشد که با کلیک روی *تغییر گروه*، اگر درس تنها شامل یک گروه بود، پیغامی مبنی بر اینکه نمی‌توان عملیات را انجام داد به دانشجو نمایش داده شود و در صورتی که گروه دیگری موجود بود، در یک جعبه پیام کوچک، لیست بقیه گروه های درس را نشان دهد که در صورتی که دانشجو روی آن کلیک کند، گروه برای او تغییر پیدا کند و در صورتی که ظرفیت گروه انتخابی دانشجو (برای تغییر گروه) تکمیل بود، پیغام مشابه برای معاون ارسال شود.

در هر نوع اخذ و تغییر گروه، یک سری خطا وجود دارد که باید در صورت بروز هر یک، از اخذ و یا تغییر گروه درس جلوگیری شود و پیغام خطای مناسب به دانشجو نشان داده شود.

انواع خطاها:

- ظرفیت درس تکمیل شده باشد.
- پیشنیازی رعایت نشده باشد.
- زمان کلاس با دروس دیگر تداخل داشته باشد.
- زمان امتحان با دروس دیگر تداخل داشته باشد.
- از دروس معارف تنها یک درس را می‌توان اخذ کرد.

همچنین پس از اتمام زمان انتخاب واحد، باید یک بار دیگر دروسی که پیشنیازی دارند که گذرانده نشده است یا همنیازی دارند که اخذ نشده است، بررسی شود و در صورت وجود مشکل در یک درس به نحو درستی حذف شود. (برای مثال اگر دانشجویی دروس 1، 2 و 3 را در انتخاب واحد اخذ کرده باشد، و درس 2 همنیازی درس 3 را لازم دارد و درس 3 پیشنیازش درس 1 باشد، درس 3 باید حذف شود زیرا درس 1 باید در ترم‌های قبل گذرانده می‌شده و پس از حذف درس 3، درس 2 نیز حذف می‌شود زیرا همنیاز درس 3 است).

همچنین پس از این زمان، صفحه‌ی درس در درس‌افزار به صورت خودکار ایجاد شده و تمام دانشجویان درس به آن افزوده می‌شود.

صفحه پیغام‌ها:

درخواست‌های دریافت‌شده از سایر کاربران سامانه و اطلاع‌رسانی‌های سیستم، در این بخش به کاربر نشان داده می‌شود. سپس کاربر می‌تواند درخواست را قبول یا رد کند. توضیحات نحوه‌ی ارسال درخواست و اطلاع‌رسانی‌های سیستم در سایر بخش‌ها آمده است. لازم به ذکر است تمام درخواست‌های آموزشی و دانشجویی به استادها و معاون و رئیس، باید در بخش پیغام‌های آنها و پاسخ آنها نیز در بخش پیغام‌های صفحه ی دانشجو نمایش داده شوند.

صفحه پیام‌رسان:

صفحه‌ی پیام‌رسان که باید از طریق صفحه اصلی به آن دسترسی داشته باشیم، شامل دو بخش است:

- **چت‌ها**

لیست چت‌های کاربر در این بخش نمایش داده می‌شود. در هر آیتم، باید نام مخاطب چت و آخرین پیام چت مشخص باشد. همچنین این لیست بر اساس آخرین پیامی که داده شده است مرتب شده است. کاربر پس از انتخاب هر چت، وارد چت روم آن مکالمه می‌شود و او باید بتواند تمامی پیام‌های رد و بدل شده تا به الان را با ترتیب زمانی صحیح مشاهده کند. همچنین در بالای چت روم نام و عکس طرف مقابل نمایش داده شود. کاربر باید بتواند پیامی به فرد مقابل ارسال نماید. این پیام می‌تواند بصورت فایل صوتی، عکس، PDF و یا فیلم باشد و زمان ارسال هر پیام باید مشخص باشد. توجه کنید که پیام‌ها برای هر دو طرف باید همزمان و هماهنگ ارسال شود.

- **ایجاد چت**

هر کاربر در سامانه، باید بتواند چت جدید ایجاد کند. دانشجو پس از ورود به این صفحه، لیستی از دانشجویان هم‌رشته و هم‌ورودی‌اش و استاد راهنمایش را مشاهده می‌کند و می‌تواند به یکی از آنها یا بصورت دسته جمعی به بخشی از آن ها و یا همه آنها پیام ارسال کند. همچنین اگر دانشجو خواست به استاد یا دانشجویی که در این لیست نیست پیام بدهد، با دادن کد دانشجویی یا استادی مخاطب، به او درخواست ارسال پیام میدهد. این درخواست در لیست پیغام‌های مخاطب او بلافاصله باید ارسال شود. استاد نیز در هنگام ورود به این صفحه، لیستی از دانشجویانی که استاد راهنمایشان هست را مشاهده میکند و به همان صورت، میتواند در سه حالت فوق به آنها پیام دهد. معاون آموزشی و رئیس دانشکده نیز به همین صورت، به تمام دانشجویان دانشکده پیام بدهد. آقای محسنی نیز لیست کل دانشجویان دانشگاه را دارد و همینطور نیز میتواند به سه روش به آنها پیام بدهد.

کاربران جدید برنامه:

- **ادمین EDU:**

یک کاربر در سامانه با کد 1 داریم که قرار است پاسخگوی مشکلات دانشجویان باشد. هنگامی که یک کاربر به شبکه متصل است، می‌تواند به این کاربر پیام دهد و مشکلات خود را توضیح دهد. تا زمانی که اتصال برقرار نشده است، این پیغام ذخیره می‌شود و بلافاصله پس از اتصال به شبکه، پیام باید برای ادمین ارسال شود. توجه کنید در صورت خارج شدن از برنامه، این پیام همچنان باید ذخیره شود و پس از اتصال ارسال شود.

- **آقای محسنی:**

در صفحه اصلی‌اش، تنها یک بخش برای جستجوی دانشجویان و یک بخش برای پیام دادن با یک سری فیلتر خاص (با حداقل 3 مشخصه) به دانشجویان وجود دارد. در بخش جستجوی دانشجویان، لیستی باز می‌شود که شامل همه دانشجویان به ترتیب سال ورودشان است. در هر آیتم، نام و شماره دانشجویی و مقطع هر دانشجو مشخص می‌شود (نیازی به دیدن یکجای تمام دانشجویان نیست و میتوان در آن اسکرول کرد). در بالای این لیست، یک textbox وجود دارد که هرگاه آقای محسنی ارقام شماره دانشجویی را وارد می‌کند، باید بلافاصله یا با یک وقفه، لیست زیر آن به‌روز شود و کل دانشجوهایی که شماره دانشجویی هایشان در آن ارقام صدق میکند ظاهر شوند. (توجه شود نباید هیچ کلیدی برای سرچ کردن وجود داشته باشد و با نوشتن یک شماره دانشجویی، در لیست زیر آن تنها یک دانشجو با شماره دانشجویی مشخص موجود باشد) برای مثال اگر 98 را تایپ کند، لیست کل دانشجوهای 98 و اگر 98101 را تایپ کند، لیست دانشجوهایی که شماره دانشجویی‌شان با 98101 شروع می‌شود در لیست زیرش ظاهر شود. آقای محسنی باید بتواند با کلیک روی هر آیتم دانشجو، مشخصات کامل دانشجو که در صفحه مشخصات خود دانشجو نیز مشخص بود را ببیند.

- **دستیار آموزشی:**
دستیار آموزشی همان دانشجویان هستند که به صورت دیگری به کلاس درس‌افزار افزوده می‌شوند که توضیح داده می‌شود.

درس‌افزار (Courseware):

این قسمت مانند قسمت *درس‌های من* در CW می‌باشد. این قسمت مختص به اساتید و دانشجویان می‌باشد. توضیحات این قابلیت در ادامه آمده است:

- **اضافه کردن دانشجو به صفحه درس:**

اگر دانشجویی بعد از پایان مهلت انتخاب واحد به درس اضافه شد، استاد درس می‌تواند با وارد کردن شماره دانشجویی دانشجو در بخش از صفحه درس، او را به صفحه درس اضافه کند. همچنین استاد، یک دانشجو را می‌تواند به دو صورت دانشجو یا دستیار آموزشی اضافه کند. پس از اضافه شدن به کلاس (چه بصورت خودکار و چه به صورت دستی توسط استاد درس) یک اطلاع‌رسانی از طرف سیستم باید در قسمت "پیغام‌ها" به کاربر ارسال شود. این اطلاع‌رسانی توسط سیستم انجام می‌شود و مشخص می‌شود که کاربر به عنوان دانشجوی درس و یا به عنوان دستیار آموزشی به کلاس اضافه شده است (باید با اضافه کردن دستیار آموزشی، دانشجو باید به لیست دستیاران آموزشی درس مورد نظر اضافه شود).

- **صفحه درس:**

دانشجو پس از ورود به صفحه اصلی، می‌تواند منوی *درس‌افزار* را مشاهده کند که در دسته‌ی درس‌های من، درس هایی را که اخذ کرده است و برای آنها کلاس ایجاد شده است را می‌بیند. با کلیک روی هر درس، به صفحه ی آن درس هدایت

می‌شود که می‌تواند تقویم آموزشی درس را که شامل پایان مهلت تمرین‌ها است را در آن ببیند. سپس مطالب آموزشی، تمرین‌ها و امتحان‌ها را می‌تواند مشاهده کند و وارد هر کدام بشود. به سلیقه خودتان باید هر سه بخش را نمایش دهید اما رعایت ترتیب زمانی بر 2 اساس ریلیز شدن و فاصله تا تمام شدن ضروری است. برای استاد هم دقیقا صفحات به همین صورت طراحی شده اند اما در بالای هر بخش (مطالب آموزشی و تمرین و امتحان) استاد باید یک گزینه ایجاد نیز مشاهده کند اما دستیار آموزشی نباید کلید ایجاد را داشته باشد.

- **ایجاد صفحه‌ی درس:**

هر درس بصورت خودکار و پس از تعریف شدن، یک صفحه برای آن در درس‌افزار ایجاد می‌شود که استاد مربوطه این درس را در صفحه خود می‌تواند مشاهده کند. بلافاصله پس از اخذ یک درس در انتخاب واحد دانشجو (پس از نهایی شدن انتخاب واحد و حذف دروس غیرمجاز)، صفحه‌ی درس باید به درس‌افزارش اضافه شود. همچنین در صورت حذف کردن درس، این صفحه نیز باید از لیست صفحاتش پاک شود.

- **مطالب آموزشی:**

هنگامی که استاد میخواد مطلب آموزشی جدیدی را ایجاد کند، با کلیک روی کلید ایجاد در ابتدا نام مطلب آموزشی را از استاد می‌گیریم. سپس صفحه‌ای باز می‌شود که 2 گزینه اضافه کردن فایل مدیا و اضافه کردن متن در آن وجود دارد. با هر بار اضافه کردن یک آیتم جدید، در پایین آن یک گزینه اضافه کردن آیتم جدید و ثبت نهایی مطلب وجود دارد که در صورت اضافه کردن مطلب جدید، باز می‌تواند متن یا مدیای جدید اضافه کند. استاد باید بتواند حداکثر 5 آیتم به صفحه آموزشی اضافه کند.

مدیا باید شامل عکس، فیلم، فایل صوتی یا PDF باشد.

صفحه مطالب آموزشی نیز پس از تکمیل، باید به یک صورت برای دانشجو و استاد نشان داده شود. اما در صفحه مطلب آموزشی برای استاد، باید سه گزینه برای *تغییر* و *حذف* و *اضافه کردن* هر آیتم و یک گزینه در بالای کل بخش برای *پاک کردن* مطلب آموزشی وجود داشته باشد.

دستیار آموزشی بخش حذف آیتم را ندارد ولی می‌تواند آن را تغییر دهد.

- **تمرین:**

پس از ایجاد تمرین توسط استاد، باید زمان باز شدن تمرین، زمان بسته‌شدن، زمان آپلود کردن تمرین بدون کسر نمره، نام تمرین، توضیحات و فایل PDF تمرین و همچنین نوع فایل مجاز که به دو دسته متنی یا مدیا تقسیم شده است، از استاد گرفته شود. پس از ایجاد، استاد در صفحه تمرین می‌تواند فایل‌ها و یا متن‌های ارسالی دانشجویان و زمان ارسال آنها را ببیند و در صورتی که مدل آپلود مدیا بود، بتواند آن‌ها را دانلود کند و به هر کدام از آنها در بخش همان دانشجو نمره بدهد.
دانشجو نیز با ورود به این صفحه، فایل و توضیح سوالات، وضعیت تحویل، نمره، زمان باز و بسته شدن تمرین و مهلت ارسال بدون کسر نمره را مشاهده می‌کند. در محلی از این صفحه، می‌تواند بسته به مدل تمرین، یک متن به عنوان پاسخ نوشته یا مدیای پاسخ را آپلود کند.
دستیار آموزشی در این بخش نیز می‌تواند لیست دانشجویان را مشاهده کند اما اسم و شماره دانشجویی‌شان را نمی‌تواند ببیند و کاراکتر های آن ها را بصورت * می‌بیند. اما می‌تواند پاسخ‌ها را دیده و نمره بدهد.

- **تقویم آموزشی جامع:**

در این بخش درس‌افزار، دانشجو و استاد، هر دو پایان مهلت تمرین‌های (بدون کسر نمره) هر درسی که در صفحه‌ی درسشان قرار دارد بعلاوه‌ی پایان‌ترم‌های درس‌هایی که دارند را می‌توانند ببینند.

حالت آفلاین:

ممکن است به علت بروز برخی خطاها، مانند خاموش بودن سرور، یا اشتباه بودن آدرس سرور و...، اتصال به سرور امکان پذیر نباشد. همچنین ممکن است به علت مشکلات ناگهانی سمت سرور مانند قطع شدن ارتباط سرور و دیتابیس، سرور به صورت ناگهانی قطع شود و در نتیجه اتصال کلاینت و سرور هم دچار مشکل شود. در این دو حالت برنامه‌ی شما باید بتواند **برخی** از کارها را به صورت آفلاین نیز انجام دهد.

در صورتی که اتصال برقرار نباشد یا اتصال کاربر قطع شود، باید این موضوع به کاربر اطلاع داده شود. همچنین باید یک دکمه وجود داشته باشد که کاربر با کلیک بر روی آن، بتواند به صورت مجدد سعی کند تا به سرور متصل شود و به کاربر اطلاع داده شود.

همچنین در حالت آفلاین، باید همواره پیامی مبنی بر آفلاین بودن به کاربر نمایش داده شود.

برخی از امکانات برنامه باید به صورت آفلاین و در هنگامی که کاربر به سرور متصل نمی‌باشد، قابل انجام باشند. امکاناتی که انتظار می‌رود تا برنامه در حالت آفلاین بتواند آن‌ها را انجام دهد به صورت زیر است:

1. اطلاعات صفحه‌ی اصلی (از دید تمامی کاربران)
2. خدمات آموزشی: (از دید اساتید و تمامی دانشجویان تمامی مقاطع)
    - برنامه هفتگی
    - لیست امتحانات
3. امور کارنامه: (از دید دانشجویان تمامی مقاطع)
    - وضعیت تحصیلی
4. پروفایل کاربر (از دید تمامی کاربران)
    - تمامی موارد فاز قبل به غیر از ایجاد تغییر در اطلاعات
5. مشاهده پیام‌های قبلی تا **حداقل ده پیام آخر** (در صورت وجود) رد و بدل شده در هر چت‌روم (از دید تمامی کاربران)
    - کاربر نمی‌تواند در این حالت پیامی ارسال کند و تنها می‌تواند مشاهده کند.

دیتابیس (Database):

این بخش از پروژه امتیازی بوده و اجباری به پیاده‌سازی آن نیست ولی در صورتی که از دیتابیس‌های معرفی‌شده استفاده نکردید، لازم است اطلاعات کاربران و دیگر اجزای

برنامه را همچنان در یک فایل متنی یا JSON نگه دارید تا برنامه شما **قابلیت سیو و لود اتفاقات جدید** را داشته باشد.

در این بخش هدف این است که شما داده‌های پروژه خود را در یک پایگاه‌داده (دیتابیس) ذخیره کنید و کار با یک ORM یا زبان SQL را فرا بگیرید.

به عنوان پیشنهاد، شما برای دیتابیس خود می‌توانید از سیستم‌های مدیریت دیتابیس MySQL و یا PostgreSQL که از مدل رابطه‌ای استفاده می‌کنند بهره ببرید.

برای تحویل این بخش حتما از یک رابط کاربری گرافیکی جداگانه برای دیتابیس خود استفاده کنید. به عنوان مثال می‌توانید از امکانات نسخه Ultimate اینتلیجی استفاده کنید یا برای دیتابیس Postgresql از pgadmin استفاده کنید.

همچنین شما می‌توانید از ابزارهای ORM (مانند Hibernate) و ... برای پیاده‌سازی دیتابیس‌های مورد نیاز استفاده کنید یا با استفاده از کتابخانه‌های ارتباط مستقیم (مانند JDBC) برای ارتباط با دیتابیس استفاده کنید.

نمره شما در این بخش به کیفیت کار شما در تعریف درست رابطه‌ها، تعریف درست اتفاقاتی که بعد از حذف یا بروزرسانی می‌افتد (cascade, no action, restrict, ...)، تعریف درست entityها و عملکرد درست بستگی دارد.

- ## معماری

با توجه به معماری سرور-کلاینت محور بخش آخر پروژه، سرور سامانه باید یک دیتابیس منحصر به خود داشته باشد (از طرفی هر کلاینت نیز می‌تواند هنگام اجرا داده‌های خود را در دیتابیس نگه دارد، اما در این بخش الزامی به پیاده‌سازی آن نیست و پیاده‌سازی دیتابیس سمت کلاینت مورد ارزیابی قرار نمی‌گیرد).

در زمان شروع به کار سرور، سرور تلاش می‌کند که به دیتابیس خود وصل شود. در صورتی که نتوانست (به هر علتی) اتصال خود را برقرار کند، سرور پیام خطای مناسبی را به کاربر می‌دهد.

- دقت کنید که در این حالت سرور می‌تواند منتظر برقرار شدن مجدد اتصال بماند یا اینکه به طور کامل متوقف شود که خودتان می‌توانید پیاده سازی دلخواه را انتخاب کنید اما در هر حالت، سرور نمی‌تواند بدون دیتابیس خود به کار ادامه دهد و کلاینت‌ها را مدیریت کند.
- دقت کنید که اگر به هر دلیلی، اتصال سرور با دیتابیس خود در حین اجرای برنامه از بین رفت، در این صورت سرور باید این را به تمام کلاینت‌های خود اطلاع دهد و هشدار مناسبی را برای کلاینت‌ها بفرستد (طبیعتا در این حالت کلاینت‌ها آفلاین خواهند شد).

- ## ساختار جدول‌ها (Entity)

هر دیتابیس شما، شامل تعدادی جدول است که در هر جدول نیز تعدادی record وجود دارد. انتظار می‌رود که داده‌های درون هر جدول با نام و کاربرد آن هم‌خوانی داشته باشند، از طرفی نباید داده‌هایی (رکوردها) که بهم ارتباطی ندارند در یک جدول قرار بگیرند. با توجه به ساختار رابطه‌ها در جدول‌های SQL، برای هر رکورد یا ردیف تعدادی ستون وجود دارد. انتخاب نوع و نام ستون‌ها باید توجیه منطقی داشته باشد. همچنین در صورت وجود ارتباطی بین رکوردهای جدول‌ها باید از Foreign-Keyهای مناسبی استفاده کنید و برای رکوردهای هر جدول نیز

باید یک (یا چند) ستون را به عنوان Primary-Key در نظر بگیرید تا رکوردها را به طور یکتا، مشخص کند.

برای درک بهتر می‌توانید مثال زیر را در نظر بگیرید:

در برنامه درس‌های مختلفی وجود دارد. از طرفی هر درس را استادی ارائه می‌دهد و تعدادی از کاربران این درس را در برنامه درسی خود دارند. از طرف دیگر، هر درس توسط یک دانشکده ارائه می‌شود. در اینجا می‌توانید با یک جدول کمکی ارتباط درس‌ها و دانشجو را برقرار کنید (جنس ارتباط این دو از نوع many to many است، زیرا هر درس متعلق به تعدادی دانشجو است و هر دانشجو می‌تواند چند درس را داشته باشد) از طرفی می‌توانید با یک جدول دیگر، ارتباط بین دروس و دانشکده‌ها را برقرار کنید (جنس این رابطه از نوع many to one است زیرا هر دانشکده تعدادی درس دارد ولی هر درس متعلق به یک دانشکده است).

- در صورتی که از ابزارهای ORM استفاده می‌کنید، معمولا این ابزارها طراحی خودکار مناسب جدول‌ها را انجام می‌دهند. مطالعه این ابزارها می‌تواند در این بخش به شما کمک کنند.

داده‌هایی که باید ذخیره شوند

سرور شما باید بتواند داده‌های زیر را از دیتابیس استخراج کند:

- مشخصات تمام کاربران
- تمام کلاس‌ها و مشخصات آنها
- تمام چت‌ها

به عنوان یک قاعده کلی، بجز configهای برنامه، **تمام اطلاعات مورد نیاز در بخش‌های قبلی** باید از دیتابیس دریافت شوند.

- توجه کنید که نیازی به بهینه‌سازی بیش از نیاز اولیه در پیاده سازی ساختار و معماری‌های مربوط به دیتابیس و جدول‌های آن نیست. هدف این بخش

پیاده‌سازی یک پایگاه ذخیره‌سازی و کار کردن با آن است بنابراین کافی است حد متوسطی از استانداردها را رعایت کنید.

کلین کد (Clean Code):

شما در این فاز از پروژه باید تا حدی که در کلاس درس تدریس شده، اصول کلین کد را رعایت کنید.

1. پروژه شما باید از پکیج بندی مناسب برخوردار باشد.
2. از قراردادهای نامگذاری در جاوا برای نام کلاس‌ها، متدها، فیلدها و ... پیروی کنید. همچنین برای نامگذاری، از نام‌های مناسبی استفاده کنید. برای مثال سعی کنید نام‌ها به صورت فینگلیش نباشد، همچنین از یک استاندارد در نامگذاری خود پیروی کنید. برای مثال در اکثر استانداردهای نامگذاری در جاوا، نام پکیج‌ها شامل حروف کوچک هستند، نام کلاس‌ها با حروف بزرگ شروع می‌شوند و ...
3. وظایف کلاس‌های مختلف مختلف مشخص باشد و تا حد امکان این وظایف مینیمال باشند.
4. تا جای امکان، تلاش کنید که ارتباط منطق برنامه و گرافیک آن جدا باشد، تا در صورتی که نیاز به تغییر در منطق داشته باشید، نیاز به تغییر گرافیک نباشد و برعکس. برای اینکار می‌توانید از یک سری واسطه بین منطق و گرافیک بهره ببرید و یک پل ارتباطی بین قسمت‌های مختلف پروژه ایجاد کنید. به طوری که ارتباط قسمت منطق برنامه و گرافیک از طریق این پل ارتباطی باشد. این پل ارتباطی می‌تواند به صورت مجموعه‌ای از کلاس‌ها و اینترفیس‌ها باشد.
5. در این فاز از پروژه دیتا مدل‌های شما باید تا حد امکان مستقل از منطق برنامه و روش سیو و لود باشد. به طور کلی کلاس‌های دیتا مدل‌های شما باید ساختاری شبیه POJO را داشته باشد.

6. تابع‌هایی که در هر کلاس می‌زنید تا حد امکان کوتاه و کلی (general) باشند و قابلیت باز استفاده (reusability) داشته باشند. به طور خاص تا حد امکان کد تکراری نداشته باشید. برای مثال اگر می‌بینید تابعی وجود دارد که بیش از 7 یا 8 خط است، به این فکر کنید که چطور می‌توانید این تابع را به چند تابع مجزا با تعداد خط‌های کمتر تقسیم کنید (البته 7 یا 8 اعداد دقیقی نیستند و یک قانون سرانگشتی می‌باشد).

کانفیگ (Config):

از فایل‌های کانفیگ استفاده کنید و از hard-code کردن پارامترهای ثابت برنامه بپرهیزید. توضیحات این فایل‌ها در ادامه آمده است.

در این بخش از پروژه شما باید از ابزارهای مربوط به فایل‌های کانفیگ استفاده کنید. فایل‌های کانفیگ فایل‌هایی هستند که ما در آن‌ها پارامترهای مورد نیاز برای اجرای برنامه را نگه می‌داریم و از hard-code کردن (نوشتن آنها در کد اصلی) این پارامترها خودداری می‌کنیم. به عنوان یک مثال کوچک می‌توان به سایز فریم اشاره کرد. می‌توان به جای نوشتن سایز فریم در کلاس فریم آن را در فایل کانفیگ نوشت و در هنگام لود شدن برنامه این مقدار را از فایل‌های کانفیگ خواند.

اما این کار چه مزیتی دارد؟ فرض کنید برنامه‌ای ساخته اید و آن را در اختیار دوستانتان قرار داده اید. اما به دلایل مختلف (مثلا یکسان نبودن رزولوشن نمایشگرها) دوستانتان نمی‌توانند به درستی برنامه را اجرا کنند. بنابراین شما مجبورید که برای هر کدام از دوستانتان، یک نسخه با ویژگی‌های مطلوب (مانند رزولوشن مانیتور شخص) کامپایل کنید و برای او بفرستید (فرستادن سورس کد اصلی نیز، در بسیاری اوقات به دلایل امنیتی گزینه‌ی مناسبی نخواهد بود). در این شرایط در صورتی که از کانفیگ استفاده کنید، کافی است که اعداد موجود در آن فایل را تغییر دهید و نیازی به کامپایل کردن مجدد کد نیست!

برای استفاده و خواندن از فایل‌های کانفیگ راه‌های زیادی وجود دارد. برای مثال می‌توانید خودتان با استفاده از مفاهیم کار با فایل، اقدام به خواندن و نوشتن فایل‌های کانفیگ

کنید، یا از کلاس ها و لایبرری‌های آماده استفاده کنید.برای نوشتن و استفاده از فایل‌های کانفیگ، می‌توانید از کلاس‌های جاوا مثل Properties استفاده کنید و یا حتی از کتابخانه‌های خارجی مثل cfg4j و یا Commons Configuration استفاده کنید یا اینکه به روش دلخواه خود عمل کنید. نکته مهمی که خوب است رعایت کنید، برای کارهای مختلف فایل‌های کانفیگ مختلف داشته باشید و ساختاری شبیه به پکیج بندی کدتان را در فایل‌های کانفیگ رعایت کنید.

مواردی که میتوان آنها را در فایل کانفیگ قرار داد:

1. اعداد ثابت داخل برنامه، حداقل تعداد واحد برای ماینور، حداقل معدل برای ماینور، حداکثر تعداد واحد در یک ترم و ...
2. آدرس‌ها، مانند آدرس عکس‌ها، موسیقی‌ها، فونت‌ها، URL دیتابیس و یا آدرس فایل‌های جیسون (آدرس فولدرها)
3. گرافیک: مختصات و سایز کامپوننت‌های هر پنل (در صورتی که از روش مختصاتی استفاده کرده باشید.)
4. آدرس خود فایل‌های کانفیگ! یک فایل کانفیگ که خود شامل آدرس سایر فایل‌های کانفیگ باشد و فقط آدرس این فایل را در کد خود قرار دهید. برای اینکه این آدرس هم قابلیت عوض شدن داشته باشد.
5. تمام متن‌های ثابت برنامه (مانند متن لیبل‌ها، دکمه‌ها، متن توصیه نامه و ...) را درون فایل‌های کانفیگ قرار دهید.

با آرزوی موفقیت

تیم درس برنامه‌نویسی پیشرفته