



بلبلستان!

مقدمه

هدف از این فاز پروژه، آشنایی با ابزار جاوا سرولت و کانتینر Tomcat، طراحی MVC و render کردن سمت سرور با استفاده از JSP می باشد. همین طور مانند فاز قبل به پیاده سازی چند ویژگی دیگر از دامنه پرداخته می شود.

پیش نیازها

ابتدا لازم است که Tomcat را در سیستم خود نصب و راه اندازی کنید. برای نصب و راه اندازی این ابزار و اجرا کردن یک سرولت و JSP روی آن، از [این آموزش](#) استفاده کنید. علاوه بر محتوای مطرح شده در کلاس درس، در صورت نیاز می توانید از آموزش های موجود در [این مجموعه](#) برای آشنایی بیشتر با سرولت ها استفاده کنید.

در مرحله بعد لازم است که سرویس های جدیدی که معرفی می شوند پیاده سازی کنید. سپس در ادامه باید کدهای فازهای قبلی خود را به قالب سرولت تبدیل کنید و خروجی مجموعه سرویس هایتان را به وسیله JSP تولید کنید.

شرح فاز سوم

در فاز قبل سرویس های زیر را ایجاد کردید:

- مشاهده دروس ارائه شده
- مشاهده پروفایل
- اضافه کردن یک درس به برنامه هفتگی (پیش از ثبت نهایی)
- مشاهده دروس برداشته شده
- حذف یک درس برداشته شده (پیش از ثبت نهایی)
- نهایی کردن برنامه درسی ترم

تغییراتی که در این فاز می‌دهید به طور کلی شامل:

- جست‌وجوی دروس مورد نظر با اسم درس
- دریافت شماره دانشجویی کاربر (نوعی لاگین ابتدایی)
- حذف درس برداشته شده و اضافه کردن درس پس از ثبت نهایی

صفحات JSP

خروجی تمامی سرویس‌های شما در قالب چند صفحه JSP به کاربر نمایش داده می‌شوند. همانند فاز قبل در این فاز هم ظاهر صفحات اهمیتی ندارند و باید از قالب‌هایی که در اختیارتان قرار داده شده است استفاده کنید (به دلیل قابلیت‌هایی که باید برایشان پیاده‌سازی کنید). برای مثال دقت کنید تعدادی از این صفحه‌ها دارای لینک به صفحه خانه هستند و باید بتوانید از صفحات مختلف به خانه بروید. می‌توانید در صورت نیاز به این قالب‌ها اطلاعات بیشتری اضافه کنید. شما پنج صفحه لاگین، انتخاب واحد، خانه، پروفایل و برنامه هفتگی را برای بلبل‌ستان از سمت سرور render می‌کنید.

صفحه خانه

/

این صفحه از طریق ادرس بالا در دسترس قرار می‌گیرد. قالب آن home.html است. شامل تعدادی لینک به صفحات دیگر برنامه شماست که کاربر می‌تواند از استفاده کند. تنها پارامتر dynamic آن که لازم است render کنید شماره دانشجویی فرد در بالای صفحه است.

صفحه لاگین

/login

در فاز قبلی در تمامی pathها شماره دانشجویی هم قرار داده شده بود. در این فاز در صفحه لاگین شماره دانشجویی دریافت شده و سرویس‌ها در درخواست‌های بعد با توجه به آن کار خود را انجام می‌دهند. شما باید شماره دانشجویی وارد شده را در یک متغیر جاوا ذخیره کنید. در صفحاتی که render می‌کنید شماره دانشجویی را قرار دهید. اگر درخواستی برای برنامه شما آمد اما شماره دانشجویی‌ای نداشتید، کاربر را به صفحه لاگین ریدایرکت کنید. کاملاً ساده به این قابلیت نگاه کنید و خود را درگیر حالت‌های خاص نکنید. کاربر از صفحه خانه با زدن دکمه خروج (Log Out) به آدرس logout می‌رود.

/logout

در این فاز اگر کاربر وارد این آدرس شد باید متغیر شماره دانشجویی پاک شود و کاربر به صفحه لاگین ریدایرکت شود.

صفحه انتخاب واحد

/courses

در فاز قبل شما صفحات متعدد و جدای از هم برای کارهای مختلف انتخاب واحد داشتید. در این فاز این صفحات با هم یکی شده‌اند و بخش اصلی برنامه شما در این صفحه است. این صفحه در واقع ترکیب چهار صفحه زیر است:

- مشاهده دروس موجود
- اضافه کردن یک درس به برنامه هفتگی
- مشاهده دروس برداشته شده و امکان حذف یا اضافه کردن درس به آن‌ها
- نهایی کردن برنامه درسی ترم

بخش جدول بالایی

در ابتدای این صفحه اگر کاربر دروس ثبت‌شده‌ای داشت، برایش **render** می‌کنید (اگر نداشت جدول خالی می‌ماند) و در جلوی آن‌ها هم مانند دروس ثبت نهایی نشده، دکمه حذف وجود دارد. در واقع کاربر تغییراتی که می‌خواهد بدهد را در این جدول انجام می‌دهد و با زدن دکمه **submit plan** برنامه او تایید نهایی می‌شود (مانند گلستان). اگر اروری مانند:

- کمبود ظرفیت
- عدم رعایت حداقل و حداکثر
- عدم رعایت پیش‌نیازی
- اخذ مجدد درس در حالتی که فرد درس را پاس شده

وجود داشت باید ارور مناسب به کاربر نمایش داده شود (می‌توانید از **submit_failed** استفاده کنید) و اگر مشکلی نبود به ظرفیت ثبت‌نام شده دروس یکی اضافه شود و کاربر به صفحه برنامه هفتگی ریدایرکت شود.

دکمه ریستی برای این بخش در نظر گرفته شده که با زدن آن وضعیت برنامه فرد مطابق آخرین وضعیت برنامه تایید شده وی می‌شود. یعنی برای مثال اگر قبلاً انتخاب واحدی را انجام نداده بود با زدن این دکمه جدول کاملاً خالی می‌شود و اگر قبلاً انتخاب واحدی را ثبت نهایی کرده بود با زدن این دکمه وضعیت جدول مطابق با آخرین انتخاب واحد انجام شده وی می‌شود.

بخش جست‌وجو و جدول پایینی

جدول پایینی شامل اطلاعات همه دروس است (در این فاز یک فیلد **Signed Up** اضافه شده است که نشان دهنده تعداد افرادی است که درس را برداشته‌اند) اگر کاربر بر روی دکمه **Add** جلوی هر درس کلیک کرد در صورتی که خطایی مانند:

- تداخل زمانی درس با دروس جدول بالا
- تداخل امتحانی درس با دروس جدول بالا

نداشت درس برای فرد به جدول بالایی افزوده می‌شود.

کاربر همچنین می‌تواند با جست‌وجو کردن با استفاده از اسم درس، جدول را فیلتر کند.

دکمه clear search وضعیت جدول پایین را به حالت اولیه بر می گرداند.

برای این که تفاوت هر باتن دیگر در ریکویست پست مشخص شود هر باتن پارامتری به نام action می فرستد که جدول آنها در زیر آمده است.

Action	Description
remove	حذف درس از برنامه (جدول بالایی) همراه آن دو پارامتر course_code و class_code نیز ارسال می شود.
submit	برنامه فرد در صورت نبود ارور در سیستم ثبت نهایی شود.
reset	وضعیت برنامه فرد مطابق آخرین وضعیت برنامه تایید شده وی می شود.
search	بر اساس پارامتر search جدول پایینی را فیلتر کند.
clear	فیلتر جدول پایینی پاک شود.
add	اضافه کردن درس به برنامه (جدول بالایی) همراه آن دو پارامتر course_code و class_code نیز ارسال می شود.

صفحه پروفایل

/profile

این صفحه مانند فاز قبل پیاده سازی می شود.

صفحه برنامه هفتگی

/plan

در این فاز این صفحه شامل برنامه هفتگی دروسی است که ثبت نهایی شده اند.

پیاده سازی منطق برنامه

قابلیت جست و جو

برای جست و جو با اسم، تمامی دروسی که اسمشان با رشته ورودی مطابقت دارد و یا رشته ورودی، زیررشته اسم آنهاست برگردانید. هیچ شرطی که بر مبنای دروس پاس شده دانشجو و پیش نیازهای اوست اعمال نکنید.

قابلیت حذف درس برداشته شده

در این فاز کاربر شما می‌تواند پس از نهایی کردن یک برنامه باز هم درس اضافه کند یا کم کند؛ در نتیجه اگر درس ثبت شده‌ای را حذف کند، باید از تعداد ثبت‌نام شده‌های درس یکی کم شود. تا زمانی که ثبت نهایی نکرده است برنامه هفتگی‌اش تغییر نمی‌کند و تعداد ثبت‌نامی‌های درس کم و زیاد نمی‌شوند.

تعریف ظرفیت پر شده برای دروس

در این فاز هر درس یک تعداد افراد ثبت‌نام شده یا به تعبیر دیگر ظرفیت پر شده دارد که مقدار اولیه آن صفر است اگر ظرفیت درسی پر شده بود کسی حق ندارد آن را بردارد و باید خطای مناسب آن نمایش داده شود. (ارور کمبود ظرفیت)

نکات مهم

- سعی کنید بخش منطق پروژه خود را از کار با I/O جدا کنید، چرا که در فازهای بعد فرمت خروجی تغییر می‌کند.
- در صورت وجود ابهام فایل‌های قالب خروجی را مشاهده کنید و اگر ابهامتان رفع نشد سوال بپرسید.

نکات پایانی

- کافی است که یکی از اعضای گروه Hash مربوط به آخرین کامیت پروژه را در سایت درس آپلود کند. در هنگام تحویل، پروژه روی این کامیت مورد ارزیابی قرار می‌گیرد.
- متاسفانه سایت گیت‌لب برای ایرانیان در دسترس نیست و برای دسترسی به آن می‌توانید از [شکن](#) یا [FoD](#) استفاده کنید.
- ساختار صحیح و تمیزی کد برنامه، بخشی از نمره‌ی این فاز پروژه‌ی شما خواهد بود. بنابراین در طراحی ساختار برنامه دقت به خرج دهید.
- هدف این تمرین یادگیری شماسست. لطفاً تمرین را خودتان انجام دهید. در صورت مشاهده‌ی مشابهت بین کدهای دو گروه، از نمره هر دو گروه مطابق سیاستی که در کلاس گفته شده است کسر خواهد شد.
- سوالات خود را تا حد ممکن در فروم درس مطرح کنید تا سایر دانشجویان نیز از پاسخ آنها بهره‌مند شوند. در صورتی که قصد مطرح کردن سوال خاص‌تری داشتید، از طریق ایمیل با طراحان این فاز پروژه ارتباط برقرار کنید.
- ایمیل طراحان پروژه:

adabinejadmelika@gmail.com

mohammadalisadraci@gmail.com

موفق باشید