



توضیحات پروژه اول:

در این پروژه دانشجویان محترم باید یک سرویس نوبت دهی مطب ها را به صورت شی گرا برنامه نویسی کرده و با استفاده از ویژگی هایی که در ادامه خواهیم گفت اجرا کنند. در این پروژه شما باید برنامه را نه از دید کاربر و نه از دید ادمین بنویسید بلکه باید سیستم طوری طراحی شود که هر فرد در ابتدای ورود خود به سامانه باتوجه به سطح دسترسی که دارد ویژگی های مربوط به خود را ببیند. به عنوان مثال یک کاربر وقتی وارد سامانه خواهد شد باید بتواند نوبت گیری نماید و در سوی دیگر اگر منشی یک مطب خاص وارد سیستم شد باید بتواند نوبت هایی که به آنها تخصیص داده شده است را مشاهده کند. این موارد همگی در بخش سطح دسترسی ها تعریف شده اند. به یاد داشته باشید که یکی از مهم ترین کارهایی که در پروژه های گروهی برنامه نویسی انجام خواهد پذیرفت، مستند سازی دقیق و اصولی است.

شروع اولیه پروژه

در این قسمت به صورت پیشنهادی حداقل شی های موردنیاز شما برای برنامه نویسی این سیستم شرح داده شده است. در نظر داشته باشید که شما باید شی های بیشتر و یا متفاوتی تعریف کنید اما اجزا و ویژگی های هر شی باید توانایی این را داشته باشند تا سیستم را بدون مشکل ایجاد کنند.

شی های پیشنهادی برای این پروژه:

شی کاربر:

- ویژگی ها: شناسه کاربری، نام، ایمیل، رمز عبور، نوع کاربر (بیمار یا کارکنان کلینیک)
- متدها: ثبت نام، ورود، به روز رسانی پروفایل، مشاهده ملاقات ها

شی کلینیک:

- ویژگی ها: شناسه درمانگاه، نام، آدرس، اطلاعات تماس، خدمات ارائه شده، در دسترس بودن
- متدها: AddClinic, UpdateClinicInfo, SetAvailability, ViewAppointment

شی نوبت دهی:

- ویژگی ها: AppointmentID, ClinicID, UserID, DateTime, Status
- متدها: ثبت نوبت بیمار، لغو نوبت بیمار، زمان بندی مجدد نوبت بیمار

شی اطلاع رسانی:

- ویژگی ها: شناسه اعلان، شناسه کاربری، پیام، تاریخ زمان
- متدها: ارسال اعلان (نوتیفیکیشن)

در نظر داشته باشید که این چهار شی فقط به عنوان مثال آورده شده اند و شما برای به دست آوردن نمره کامل این بخش نیاز دارید تا حداقل ۶ شی با ویژگی ها و متدهای از پیش تعیین شده در سیستم خود استفاده کنید.

در قسمت اولیه پروژه شما نیازمند این خواهید بود که شی های خود را تعریف کرده و UML diagram مربوطه را رسم نمایید برای دریافت نمره کامل باید کد و مستندات این بخش به صورت کامل با یکدیگر همخوانی داشته باشند و تمام اعضای گروه تسلط کافی به ارائه جزئیات آن داشته باشند. برای ارائه دیاگرام این پروژه از Class Diagram استفاده کنید.

گام دوم مهندسی جداول

در این قسمت از پروژه شما باید جداول مربوط به سیستم خود را به صورت Relational تعریف کرده و قادر باشید که به صورت صحیح کوثری ها را روی آن اجرا کنید. شما برای این قسمت نیاز دارید حداقل ۵ جدول ایجاد کنید و این جدول ها را به یکدیگر مرتبط کنید. در جلسه تمام اعضا باید بتوانند کوثری های تست را روی دیتابیس خود اجرا کنند. برای به دست آوردن نمره کامل این بخش علاوه بر کدهای ایجاد جداول باید به صورت کامل مستند باشند و تمامی اعضا به آن مسلط باشند!

گام سوم اجرا سیستم

در این بخش محدودیت ها و سناریوهایی برای شما تعریف خواهد شد که باید مطابق با آن سیستم خود را بسازید.

سناریو اول: کاربر به عنوان صفحه اولیه خود باید به اکانت خود وارد شود. او نام کاربری خود را وارد خواهد کرد و سپس با انتخاب یکی از گزینه های موجود (رمز یک بار مصرف/ رمز عبور ثابت) به مسیر خود ادامه خواهد داد. پس از ورود کاربر صفحه اصلی وبسایت را مشاهده خواهد کرد و باید بتواند از لیست روبروی خود میان وقت های رزرو فعلی، سابقه وقت های گذشته و رزرو نوبت جدید یک گزینه را انتخاب کند. در این سناریو کاربر رزرو وقت جدید را انتخاب می کند و در یک قسمت سرچ کلید واژه ای از اسم دکتر/کلینیک مورد نظر خود را یادداشت می کند). برای پیاده سازی این ماژول می توانید از RegEX و پکیج های آماده پایتون استفاده کنید (.پس از سرچ کردن کاربر از میان گزینه های احتمالی خود یک دکتر/کلینیک را گزینش خواهد کرد. پس از ورود به صفحه دکتر وقت های موجود به او نمایش داده خواهد شد و پس از انتخاب، کاربر یک پیغام تاییدیه از سمت سیستم دریافت میکند و قادر است وقت خود را در قسمت وقت های رزرو فعلی مشاهده کند.

محدودیت های سناریو اول و راهنمایی: وقت های موجود باید از یک API که مستندات آن در انتهای پروژه آمده است استفاده کند و همچنین پس از ثبت توسط کاربر باید با یک دستور وقت موجود را از وقت های موجود در API کم کند. همچنین کاربر باید بتواند در هر لحظه به صفحه منو اولیه باز گردد و از حساب کاربری خود خارج شود .

سناریو دوم: در این سناریو منشی مطب با همان صفحه سناریو اول وارد می شود و با نام کاربری و متد ورود انتخابی خود وارد سیستم می شود. در اینجا سیستم باید تشخیص دهد که نقش این کاربر در پایگاه داده فرق دارد و منو متفاوتی را به او نمایش دهد. در این سناریو منشی باید سه منو نوبت های فعلی، لغو نوبت و اضافه کردن ظرفیت نوبت دهی را مشاهده کند. در این سناریو منشی پس از انتخاب افزایش ظرفیت نوبت دهی، باید بتواند به تعدادی که می خواهد نوبت های کلینیک خود را افزایش دهد. پس از افزایش یا لغو نوبت منشی باید یک پیغام تاییدیه از سیستم دریافت کند .

محدودیت های سناریو دوم و راهنمایی: در این بخش شما باید با متد POST و توسط API و مستندات آن قادر باشید ظرفیت مطب و کلینیک خود را تنظیم کنید. همچنین نوبت دهی ها نیز باید در یک جدول در دیتابیس خود تعریف کنید و به صورت مادام آن را به روز نگه دارید.

نکات استفاده از API :

ابتدا شما باید فایل همراه پروژه (api.py) را اجرا کنید. (کتابخانه های موردنیاز را قبل از اجرا نصب کنید)

با اجرای این فایل api در سیستم شما به صورت مجازی ایجاد خواهد شد.

استفاده از متد GET:

شما می توانید با استفاده از آدرس (<https://localhost/slots>) استفاده کنید. با ریکوئست دادن به این آدرس شما یک json در پاسخ دریافت خواهید کرد که ظرفیت پزشکان را نمایش می دهد.

استفاده از متد POST:

برای استفاده از این متد شما باید دو آرگومان `<the clinic code>` و `<number of reserved>` را به ریکوئست خود اضافه کنید و در غیر ارور مربوط به آرگومان دریافت خواهید کرد.

مدیریت پروژه:

در این پروژه به عنوان یک مهندس صنایع یکی از مهم ترین وظایف شما تعریف درست تسک ها و شکستن آنها به زیرکارهای کوچک تر خواهد بود. شما باید کارهای خود را با الگوی [1,3,5,8] فیبوناچی زمان بندی کنید و اگر کاری بیشتر از ۸ واحد زمانی به طول می انجامد را به کارهای با واحد های زمانی کوچک تر تقسیم کنید. برای این کار باید از سیستم مدیریت تسک های گیت هاب استفاده کنید. در زمان تصحیح ۳۰ درصد از نمره شما برای مدیریت درست پروژه و استفاده از مدیریت تسک گیت هاب خواهد بود.

نکات مهم

- استوری پوینت نشان دهنده اندازه، پیچیدگی و تلاش موردنیاز برای تکمیل پروژه است. در این روش شما یک عدد از مقیاس فیبوناچی به هر استوری پوینت اختصاص می دهید. هرچه این عدد بزرگتر باشد، استوری پوینت پیچیده تر بوده و تلاش بیشتری برای تکمیل آن نیاز است. برای مثال وقتی شما در فیبوناچی ۱ واحد زمانی را برای استوری پوینت در نظر میگیرید آن کار باید در ۱ ساعت تمام شود.
- برای دریافت نمره مدیریت پروژه گیت هاب نکات زیر را اجرا کنید
 - کدهای شما باید به تسک هایی که در مدیریت پروژه می سازید لینک شده باشند
 - ساخت تسک و زمان بندی انجام شدن آنها باید در گذر زمان باشد
 - همه افراد باید در آن مشارکت داشته باشند
 - تسک ها باید بر حسب الگوی فیبوناچی ریزدانی داشته باشند
- مثال غلط: تسک ۱ : کدنویسی منطق کلاس های موجود در پایتون - امتیاز: ۱۶
- مثال درست: تسک ۱ : کدنویسی منطق کلاس های سمت کاربر امتیاز : ۸ - تسک ۲ کد نویسی منطق کلاس های سمت کلینیک امتیاز : ۸
- API این پروژه در ریپو زیر موجود است. لطفا فایل readme را به دقت مطالعه بفرمایید.
- <https://github.com/AlibakhshiAlireza/ProjectAPI>

نکات تحویل

۱. این پروژه باید به صورت گروهی انجام شود.
۲. فایل ارسالی باید شامل گزارش کاری از مراحل انجام شده در هر قسمت به صورت خلاصه در قالب Pdf و فایل کد در یک فایل فشرده با نامی مطابق با الگوی زیر در صفحه‌ی کوئرا آپلود شود.

P1_21776_Student-ID.zip
۳. مهلت ارسال پروژه ساعت ۲۳:۵۹ روز ۲۰ دی ۱۴۰۲ است.
۴. به ازای هر روز تأخیر 15% از نمره تمرین کسر می‌شود و نهایتاً ۳۰% نمره باقی می‌ماند. دقت کنید که کسر نمره بلافاصله بعد از پایان مهلت ارسال در نظر گرفته می‌شود.
۵. حداقل ۱۰٪ از نمره پروژه مربوط به ویراستاری مناسب، انشای درست، نداشتن غلط املایی، مرتب بودن قالب گزارش، رعایت نیم‌فاصله‌ها، استفاده از معادل‌های صحیح و به صورت کلی رعایت اصول ارائه گزارش است.
۶. در صورت کشف تقلب از دو فرد، معادل نمره‌ی کل این پروژه از نمره نهایی هر دو نفر کسر خواهد شد.
۷. در فایل توضیحات حتماً باید repository گیت هاب ذکر شود و زمان ارائه دسترسی به مصحح‌ها داده شود. در صورتی که در گیت هاب موارد تعیین شده برای اندازه‌گیری مشارکت مشاهده نشود ۵۰ درصد از نمره گروه کسر خواهد شد.
۸. مشکلات و سوالات خود را در نوشته مربوط به پروژه موردنظر در صفحه درس در سایت کوئرا مطرح کنید.

موفق باشید