راهنما
Assignment 2

گام اول : به بیان آماده سازی بستر پروژه و کد نویسی data-access می پردازد و در نهایت یک وب سرویس راه می اندازد اما در این نسخه وابستگی های لوکال زیادی در پروژه موجود است و خیلی از موارد مانند نام جدول و operator هنوز از کاربر گرفته نمی شوند و این سرویس مبتنی بر یک جدول خاص با ملزومات خاص عمل می کند ، اما در هر حال وب سرویس در نهایت به درستی کار می کند و **REST api** های به دست آمده قابل تست اند.

گام دوم : به توسعه فیچر های افزون بر فیچر های درخواستی سوال می پردازد و توانستیم بیان کنیم چگونه این موارد از قبیل ، درج چند رکورد همزمان ، سرچ حرفه ای و یا برخی کوئری های آماری و تحلیلی چگونه ایجاد شده اند.
با توجه به ملزومات برخی از این کوئری ها ، ساختار به وجود آوردن این عملیات ها با بقیه پروژه متفاوت است ،برخلاف گام قبلی و گام های بعدی در برخی عملیات های این گام ، رویه یا procedure استفاده شده است و استانداردهای خاص خودشان را دنبال می کنند.

گام سوم : در این گام در واقع تبدیل و منتقل شدن پیاده سازی های گام اول را به استاندارد های Assignment آغاز می کنیم و همچنین در این گام برای اولین بار یک فیچر برای انتخاب و کوئری نویسی به شیوه Multiple Condition ایجاد می کنیم که اینجا custom نام دارد .

نکته : همه قطعه کد های مربوط به همه مراحل و گام ها موجود هستند. از npm و دستوراتش برای نصب پکیج های پروژه و ران کردن پروژه استفاده کنید و **از POSTMAN برای تست وب سرویس و api ها استفاده کنید و یا از وب سایت های آنلاین تست api استفاده نمایید.** توجه کنید که تمام REST api ها تعاملی هستند و باید برای سرویس ایجاد شده حداقل نام جدول و operator مد نظر را از طریق متد post ارسال کنید تا پاسخ دریافت کنید پس اجرای عادی URL ها در مرورگر کار درست و مرسومی نیست و امکان تست سرویس را به شما نمی دهد. من استفاده از https://reqbin.com/ را برای تست api ها پیشنهاد می کنم که به سادگی پس از نصب یک اکستنشن بر روی کروم اجازه تست هرگونه سرویس api در هر پروژه ای را می دهد.

گام چهارم : سرویس ما به طور کامل در این گام عملیاتی می شود ، هر چند که از همان گام نخست در سطح network در دسترس بوده است ، در این گام وابستگی های لوکال به طور کامل حذف می شود ، نام گذاری فایل ها تغییر می کند ، نام ها و آدرس api ها تغییر می کنند ، کد ها و فانکشن ها بهینه می شوند و قسمت هایی از کد که به نوعی با توجه به یک جدول خاص ، به طور نیمه هارد کد شده و وابسته به تعداد ستون ها و مقادیر یک جدول خاص در حال کار کردن بودند ، در لایه ی service-handler به طور کامل برطرف می شوند و در لایه data-access نیز ، تمامی احتمال های مربوط به جداول احتمالی مختلف به صورت داینامیک پوشش داده می شود و در این نقطه می توان ادعا کرد به هیچ جدول ، عملیات و operator خاصی وابستگی نداریم و همه شرایط را از کاربر به واسطه سطح network و به کمک REST api دریافت می کنیم و باز هم از طریق همان REST api به کاربر response می دهیم و در واقع سرویس ما اطلاعات مورد نیاز کاربر را به او بر می گرداند و یا دستوراتش را در دیتابیس اعمال میکند، حال کاربر می تواند از این REST api ها برای توسعه application و یا وبسایتش یا هرگونه سیستمی که امکان تعامل با REST api دارد استفاده کند. در واقع ما مطابق با بسیاری از سرویس دهنده ها و وب سرویس های استاندارد بر بستر شبکه ،از REST api برای ارتباط با کلاینتمان بهره بردیم. در نهایت ما یک دیتا اکسس به عنوان سرویس داریم که از طریق شبکه دریافت اطلاعات و تغییر و انجام عملیات بر آن ها مطابق دستورالعمل تهیه شده میسر است.

- **یک تعریف ساده به کمک گوگل : سرویس API چیست؟** سرویس‌های API رابط‌هایی هستند که برنامه‌ای را با شرح نحوه تعامل با یک سیستم به منظور بازیابی و/یا تغییر داده‌های درون آن ارائه می‌کنند.
- توجه کنید که ما در این پروژه از **REST api** استفاده کردیم که به طور گسترده برای ایجاد و توسعه **وب سرویس** استفاده می شود و در معماری **سرویس گرا** کاربرد دارد.
- **آیا REST API یک سرویس است؟**

پاسخ سوال به کمک گوگل :

بله، API های REST نوعی از پیاده سازی به شیوه **سرویس گرا** هستند. API REST یک سبک معماری استاندارد برای ایجاد یک API **وب سرویس** است. یکی از الزامات REST API، استفاده از روش های HTTP برای درخواست و پاسخ از طریق **شبکه** است.