**به نام خدا**

**مقدمه :**

در این فایل به توضیح کل بحث داکیومنت ها خواهیم پرداخت که ما برای کوچک کردن یک فرآیند بزرگ با تعداد تیبل های زیاد مجبور به ریز کردن این فرآیند و به ازای هر یک از فرآیند (بیزینس) ها یک فایل جداگونه در نظر گرفتیم و همچنین این مستندات به صورت تعدادی فایل ورد در گیت قرار خواهد گرفت که با تغییر هر فایل و یا به ازای هر پروژه امکان ایجاد برنچ جدید و اصلاح این مستندات موجود باشد.

در اینجا ما میخواهیم به تعدادی تیبل ها که این پروژه برای راحتی کنترل به پایگاه داده اضافه شده اند و همچنین معماری نرم افزار بپردازیم. این نرم افزار برای یک نرم افزار برای برگذاری دوره ها ، بر گذاری آزمون های انلاین و افلاین ، یا با اعمال تغییرات به عنوان سایت فروشگاهی به کار رفته است.

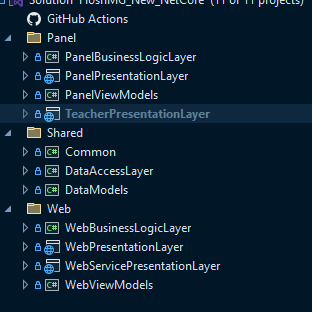
* **معماری کلی پروژه**

پروژه تشکیل یافته اند 3 بخش اساسی می باشد که در این پروژه ها داری چندین پروژه Asp.net Core (3.1) و چندین Class Library که در این کلاس یک کلاس بخصوص به عنوان ViewModel وجود دارد که به عنوان کلاس های نمایش نهایی یا همان DTO(Data Transfer Object) می باشد که این قسمت نیازی به توضیح ندارد.

اما در این پروژه 4 پروژه اصلی وجود دارد که شامل :

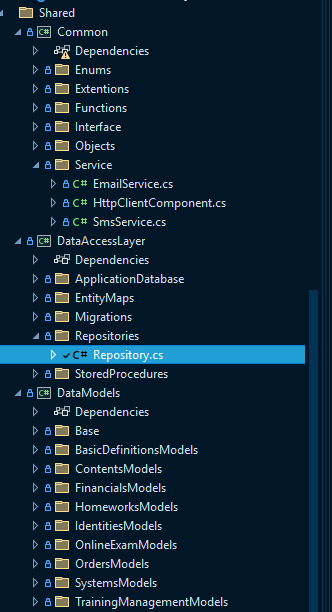
1. **WebServicePresentationLayer**
2. **WebPresentationLayer**
3. **PanelPresentationLayer**
4. **TeacherPresentationLayer**

که شامل پنل برای معلمان و اساتید و همچنین لایه ای برای نمایش پنل مدیریت نرم افزار می باشد همچنین لایه ای برای ارائه وب سرویس (API) وجود دارد و در آخر لایه دیگر برای نمایش صفحه اصلی وب سایت (Landing Page) می باشد.



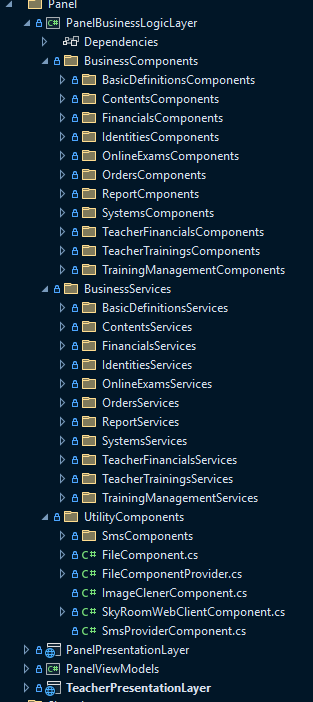
همان طور در عکس می بینید پروژه شامل سه فولدر می باشد که هر توضیحات هر کدام شامل :

1. **Shared**



در این فولدر ما تمام فعالیت های مشترک را داریم که ارتباط با سرویس های ایمیل و ارسال اس ام اس یا سرویس های دیگر بیرونی رو داریم که از نام های هر فولدر ثابل تشخیص هستند ، همچنین بحث دیتابیس که به روش (Code first approach) انجام شده است و همچنین تمام انواع Entity ،Entity Configuration و Repositories در این قسمت وجود دارد که پوشه Shard به تمام پوشه های دیگر refrence داده شده است تا اینکه بتوانیم در از این کد ها در دیگر بخش ها نیز استفاده نماییم.

1. **Panel /Web**



در این قسمت که با پوشه Web دارای شرایط و فرمت تقریبا یکسانی توسعه داده شده است ، در پروژه مذکور از خصوصیت و امکانی مانند DI(dependency injection) استفاده نشده است و برای استفاده از هر سرویس در کلاس Contractor این هر نوع از سرویس ها new شده و از آنها استفاده شده است که به نظر شخص من مشکل اساسی می باشد.

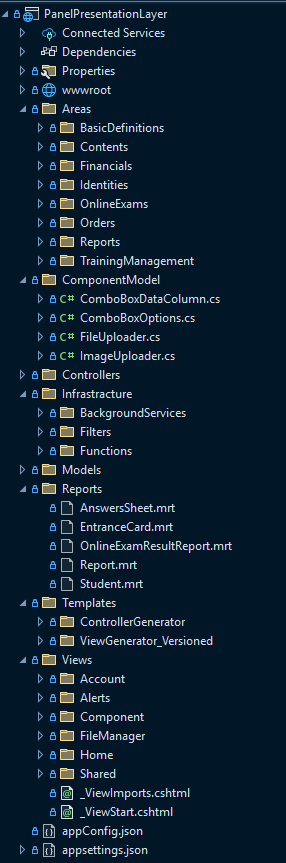
با این حال در این پروژه ها برای تحویل داده نهایی به view از چهار لایه استفاده شده است لایه ها شامل :

* **Repository Layer**
* **Component Layer**
* **Service Layer**
* **Controller**

در لایه های Presentation کد هایی برای بررسی نقش کاربر ها وجود دارد که براساس مقادیر اطلاعات کاربران درخواست دهنده به سمت کنترلر ها بررسی و در صورت وجود مشکل با برگشت پیغام عدم دسترسی به کاربران اطلاع داده می شود که همچنین تعدادی از BackGroundService وجود دارد که در حین اجرای پروژه فرآیندی را به طور مداوم در حال انجام هستند نیز در این بخش قرار داده شده اند.

بعد از تمام این موارد یک سری ساختاری برای گزارش گیری و چاپ گزارشات ، Exception Middleware و View های پروژه که هم در ردیف اصلی و هم در فولدر Area وجود دارد نیز در این قسمت قرار دارد.

در آخر قابل ذکر است که تمام اطلاعات مانند رمز ها کانکشن استرینگ و دیگر داده های حساس در فایلی به نام appConfig.json قرار دارند.



* **جداول و دیتابیس**

در این قسمت به توضیح مختصری از از دیتابیس ها خواهیم داشت در کل در پروژه و پایگاه داده ما 183 تیبل و 12 اس پی وجود دارد که در این قسمت ما به تعدادی از تیبل های کلی اشاره خواهیم و برای هر بیزینس به صورت جداگانه روابط و تیبل های هر بیزینس به صورت مجزا نمایش و توضیح داده خواهد شد.

در این پروژه برای راحتی در استفاده از انواع و نوع های به جای اضافه یا حذف کردن یک نوع ما از تیبل برای آن ها استفاده نموده ایم که هر وقت نیاز بر اضافه یا حذف شد از طریق دیتابیس اقدام به این کار خواهیم کرد. برای دیدن روابط به فایل Types\_Diagram.html تمامی فیلد های مورد نیاز را میتوانید تماشا کنید.

**همچنین در صورت تمایل فایل HooshMg\_Diagram.html شامل تمام تیبل ها می باشد و میتوانید از آن استفاده نمایید.**