PusulaTalent Case Dökümantasyon

Projede authentication ve identity mekanizması kurulmuştur . Giriş için default olarak belirlediğim admin giriş bilgileri ; email = admin@school.com password = Admin123! Bilgileriyle admin rolüyle giriş yapabilirsiniz .
Projede authhorization olduğu için postman gibi bir apı test uygulamasıyla endpointleri test edebilrisiniz , çünkü header da jwt access tokenı göndermeniz gerekicek istek atmak için . Admin bilgileriyle login olduktan sonra geriye access token dönüyor .
O access tokenı postman da headers da "key : Authorization value = Bearer "access token değeri" girerek rolün yetkisine göre apı ye istek atabilrisniz .

ENDPOİNTLER

Teacher Controller

POST create-teacher =>
Sadece admin istek atabilr . Sisteme yeni bir öğretmen kaydetmek için tetiklenir . Öğretmenin mail adresine giriş yapacağı kullanıcı bilgileri mail olarak gidiyor . Sistemde random şifre oluşturuyorum . Yazdığım mail service ile mail gönderiyoruz handler da .

DELETE delete-teacher=>
Sadece admin istek atabiilir db den öğretmenin verilerini siler . Öğretmenin id sine göre işlem yapar .

GET GetStudentByOwnCourse =>
Bu endpoint öğretmenin id sini bekler . İlgili öğretmenin ders verdiği öğrencilerin listesini geriye döner .

Student Controller

POST create-student=>

Hem öğretmen hem de admin öğrenci oluşturabilir . Öğretmende olduğu gibi öğrencide de öğrencinin mail adresine giriş bilgileri gider . Öğrenciye burda class atanmaz , sadece öğrenci oluşturulur .

POST assign-student-class=>

Öğrenci bu endpointle istenen sınıfa atanıyor.Önceden create edilen öğrenci , istenen sınıfın id sine göre atanır .

GET =>
öğrenci listesini verir

SchoolClass Controller

POST CreateSchoolClass=>
Okul için sınıf oluşturulur . Bir sınıfın birden fazla şubesi de olabilir . 9A 9B gibi . O yüzden sınıf ismini sitring bekliyor .

DELETE SchoolClass/{id} =>
id ye göre sınıf siler . Sınıfın id sini query parameter olarak bekler

GET AllSchoolClasses =>
Bütün sınıfları döner .

Note Controller

POST AddNote=>
İstenen ders için ilgili öğrenciye seçilen hoca tarafından not eklenir .

GET GetAvarageNoteByStudentId/{StudentId} =>
Öğrencinin id sine göre ders kredisi ve toplam kredisine göre not ortalamasını dönen endpoint

GET GetAllNote/{StudentId} =>
Öğrencinin id sine göre bütün nolarını aldığı ders detaylarıyla beraber döner

COURSE CONTROLLER

POST assign=>
Seçilen sınıfa ders atama işlemini öğretmen ile beraber gerçekleştiren endpoint . Çünkü bir dersi birden fazla öğretmen aynı anda veriyor olabilir . Bu sebeple sınıfa ders atama yaparken o derse öğretmeni de atıyoruz . Bu endpoint dersin id sini , öğretmenin id sini , sınıfın id sini bekler

POST =>
Ders create eder

DELETE assign/{id}=>
Atanmış dersi siler

DELETE {id}=>
dersi siler

ABSENTEEİSM CONTROLLER

POST =>
Seçilen öğrenci için ilgili güne devamsızlık create eder .

GET GetByStudentId/{id}=>
Seçilen öğrenci için idevamsızlık bilgisini döner .

AUTHCONTROLLER

POST login =>
Default admin bilgileriyle giriş yapabilrsiniz.Öğrenci ve öğretmen ise create edilip mail adreslerine gelen şifre ile giriş yapabilirler bu endpointle .

POST register=>
Yeni bir admin register olabilir

Proje genel açıklama ;
Projede katmanlı mimari kullandım . Business logic'i temiz ve sürdürebilir tutmak için CQRS pattern kullandıım .Validasyon pipeline , controller da temiz temiz bir yapı gibi sunduğu özelliklerden dolayı MediatR kütüphanesini kullandım CQRS ile beraber.
Projenin gereksinimlerinden dolayı tek bir handler da birden fazla transaction kullanmak için UnitOfWork kullandım . Generic bir Repository sınıfı oluşturup handler daki busines larımı bu metotlarla yaptım . Business ve DataAccess katmanlarındaki serviceler için service registration sınıfları oluşturdum .
Öğrenci ve öğretmen giriş bilgilerini güvenli bir şekilde iletmek için bir mail altyapısı kurdum . Mail gönderim ihtiyaçlarına göre tekli ve çoklu opsiyonları mevcut .
Veritabınımı codefirst yaklaşımı ile Entityframework orm si ile kurguladım .
Authentication için JWT authentication kullandım .

Projeyi elimden geldikçe CLEAN CODE prensiplerine göre oluşturmaya çalıştım .
Aynı zamanda çalıştığım için zaman çok sınırlıydı benim için .
Projenin frontend kısmını tamamlayamadım bu yüzden . Tam olmadığı için de paylaşamadım .
Teşekkür ederim ...