GAZİ ÜNİVERSİTESİ

TEKNOLOJİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

# İçindekiler Tablosu

GAZİ ÜNİVERSİTESİ

TEKNOLOJİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BMT-311 VERİ TABANI YÖNETİM SİSTEMLERİ

2024-2025 GÜZ DÖNEMİ

PROJE RAPORU

GRUP ÜYELERİ:

22181616010 – Bera Daştan

22161617009 – Mert Özmen

İÇİNDEKİLER

# GİRİŞ

Projemiz bir eğitim merkezinde bulunan öğrenci , öğretmen ve derslerin kayıtlarını veritabanında tutarak görüntüleme imkanı sunarak ders kayit ve yönetimini dijital ortama taşımayı amaçlamaktadır.Hedef kitlesi öğrenci, öğretim görevlileri ve yönetim birimleridir.

https://github.com/beradastan/derskayitproje

# PROJE İÇERİĞİ

PROJE TANITIMI

Bu proje, üniversitelerdeki öğrenci otomasyon sistemlerini modernize etmek ve süreçleri daha verimli hale getirmek amacıyla tasarlanmıştır. Ders kayıtları, öğrenci-öğretim görevlisi eşleştirmeleri ve veri yönetimi gibi işlemleri kapsayan sistem, kullanıcı dostu bir arayüz ve güvenli bir veri altyapısı sunmaktadır. Süreçler hızlandırılarak, üniversite yönetiminin iş yükü azaltılacak ve hem öğrenciler hem de öğretim görevlileri için daha kolay bir deneyim sağlanacaktır.

PROJE TASARIM VE GELİŞTİRME AŞAMALARI

Öncelikle veritabanı için gerekli olan tablolar oluşturuldu ve aralarındaki ilişkiler kuruldu. Sonrasında projede ne tür işlemlerin olacağına karar verildi . Ardından arayüz işlemlerine kullanıcı giriş ekranı yapılarak başlandı. Projemizde 3 tür kullanıcı olacağı için rol sistemiyle farklı formların açılması sağlandı.

Öğrenci ve Öğretim görevlisi kullanıcıları için işlemler genel olarak listeleme ve güncelleme olduğu için basit sayılırdı. Ancak yönetici için oluşturulan arayüzde hem çok daha fazla işlem hem de daha fazla arayüze erişim olacağı için daha karmaşık bir yapı kullanıldı.

Burada Yönetici işlemlerini gruplayarak daha kolay bir erişim sağlamayı amaçladık. Esas işlemler her bölümün altındadır.

KULLANILAN TEKNOLOJİLER

Arayüz ve backend kısmında c# form kullanarak basit ve kolay kullanılabilir bir arayüz tasarlamayı hedefledik.

Veritabanı için Postgresql tercih ettik , sebebi de daha az donanım gereksinimi ve ücretsiz olmasıydı.

# VERİTABANI TASARIM AŞAMALARI

Kavramsal Tasarım

Varlıklar;

Ders , Öğrenci , Öğretim Görevlisi , Kullanıcı Girişi , Öğrenci-ders ilişkisi , Görevli-ders ilişkisi

Öznitelikler;

Ders: dersAdi, dersKodu, ogrenciSayisi, dersDonemi

Öğrenci: kimlikNo, ad, soyad, dtarih, adres, telefon, email

Öğretim Görevlisi: kimlikNo, ad, soyad, dtarih, adres, telefon, email

Kullanıcı Girişi: kullaniciAdi, sifre, rol

Öğrenci-Ders İlişkisi: kimlikNo, dersKodu

Görevli-Ders İlişkisi: kimlikNo, dersKodu

Mantıksal Tasarım

Güçlü Varlıklar : Öğrenci , Öğretim Görevlisi , Ders

Zayıf Varlıklar : Öğrenci-Ders , Görevli-Ders

İlişkiler;

Öğrenci ile Ders arasında çoktan çoka ilişki

Öğretim görevlisi ile ders arasında çoktan çoka ilişki

Fiziksel Tasarım

CREATE TABLE ders (

dersKodu SERIAL PRIMARY KEY,

dersAdi VARCHAR(50),

ogrenciSayisi INTEGER,

dersDonemi INTEGER

);

CREATE TABLE ogrenci (

kimlikNo VARCHAR(50) PRIMARY KEY,

ad VARCHAR(50),

soyad VARCHAR(50),

dtarih DATE,

adres VARCHAR(50),

telefon VARCHAR(50),

email VARCHAR(100)

);

CREATE TABLE ogretimgorevlisi (

kimlikNo VARCHAR(50) PRIMARY KEY,

ad VARCHAR(50),

soyad VARCHAR(50),

dtarih DATE,

adres VARCHAR(50),

telefon VARCHAR(50),

email VARCHAR(100)

);

CREATE TABLE giris (

kullaniciAdi VARCHAR(50) PRIMARY KEY,

sifre VARCHAR(50),

rol VARCHAR(50)

);

CREATE TABLE ogrenci\_ders (

kimlikNo VARCHAR(50),

dersKodu INTEGER,

PRIMARY KEY (kimlikNo, dersKodu),

FOREIGN KEY (kimlikNo) REFERENCES ogrenci(kimlikNo),

FOREIGN KEY (dersKodu) REFERENCES ders(dersKodu)

);

CREATE TABLE gorevli\_ders (

kimlikNo VARCHAR(50),

dersKodu INTEGER,

PRIMARY KEY (kimlikNo, dersKodu),

FOREIGN KEY (kimlikNo) REFERENCES ogretimgorevlisi(kimlikNo),

FOREIGN KEY (dersKodu) REFERENCES ders(dersKodu)

);

Tablolara veriler arayüz yardımıyla uygulama üzerinden eklenmiştir.

# GÜNCEL VERİTABANI VE ER DİYAGRAMI

# SONUÇ

Projemiz hedeflediğimiz işlemleri düzgün bir şekilde gerçekleştiriyor ancak buna benzer çok daha ayrıntılı sistemler olduğu için daha üzerinde bir çok geliştirme yapılması gerekiyor . Örneğin derslerin günleri , saatleri belirlenip her öğrenci ve öğretim görevlisinin ders programı tanımlanabilir.

Projeyi yaparken daha çok veritabanı kısmında ilişki kurma deneyimi kazandık . Arayüz ve kalan diğer işlemlerde zaten deneyimimiz olduğu için çok zorlanmadan gerçekleştirdik.

KAYNAKÇA

https://tembo.io/docs/getting-started/postgres\_guides/connecting-to-postgres-with-c-sharp