

-BiteSized Diet Plan-
Kalori Takip Programı

Yazılım Gereksinimleri Belirtimi

V1.0

26.11.2024

Pakize Mısra KIRIK

Revizyon Geçmiři

Tarih	Açıklama	Yazar	Yorumlar
25.11.2024	İlk yayın.	Pakize Mısra Kırık	Birinci gözden geçirme...

İçindekiler Tablosu

REVIZYON GEÇMİŞİ.....	II
İÇİNDEKİLER TABLOSU	III
1. GİRİŞ.....	1
1.1 AMAÇ	1
1.2 KAPSAM	1
1.3 KISALTMALAR.....	2
1.4 TANITIM	2
1.5 HEDEF VE BAŞARI KRİTERLERİ	2
1.6 REFERANSLAR	3
2. FONKSİYONEL GEREKSİNİMLER	3
2.1YAPISAL	4
2.2 VERİ GEREKSİNİMİ	4
2.3 İŞLEVSEL	5
3. FONKSİYONEL OLMAYAN GEREKSİNİMLER.....	6
3.1 PERFORMANS/HIZ.....	6
3.2 İZLENEBİLİRLİK	6
3.3 KULLANILABİLİRLİK.....	6
3.4 GİZLİLİK GEREKSİNİMLERİ	6
3.5 DESTEKLENEBİLİRLİK.....	6
3.6 ARAYÜZ.....	7
3.7 GÜVENLİK.....	7
4. SİSTEM MODELLERİ	7
4.1 AKTÖRLER.....	7
4.2 SENARYOLAR	8
4.3 USE-CASE MODELLERİ	9
5. GENEL.....	10
5.1 İş KISITLARI	10
6. TEKNİK.....	11
6.1 DONANIM.....	11
6.2 YAZILIM.....	12
7.TRANSITIONAL	12
7.1 ARŞİVLEME	12

1. Giriş

Bu doküman, kullanıcıların günlük kalori takibini kolaylaştıran ve sağlıklı yaşam alışkanlıklarını destekleyen BiteSizedDietPlan yazılımının geliştirilmesi için hazırlanmıştır. Program, çok katmanlı bir mimari ile tasarlanmış ve kullanıcıların yeme alışkanlıklarını izlemelerine, analiz etmelerine ve raporlamalar yoluyla karşılaştırmalar yapmalarına olanak tanır.

Modern yazılım geliştirme prensiplerine uygun olarak yapılandırılmış olan bu program, kullanıcıların tüketim alışkanlıklarını analiz etmelerine yardımcı olacak detaylı raporlama seçenekleriyle desteklenmiştir. Bu program, hem bireysel kullanıcıların hem de toplu verilerin etkili bir şekilde yönetilmesini sağlayan güçlü ve kullanıcı dostu bir araç olarak tasarlanmıştır. Sağlıklı yaşam yolculuğunuzu desteklemek ve yeme alışkanlıklarınızı optimize etmek için ideal bir çözümdür.

1.1 Amaç

Bu projenin amacı sağlıklı beslenmeyi hedefleyen kişilerin yedikleri besinleri öğün bazlı programa giriş yaparak kalori takibi yapabilmelerini sağlamaktır.

1.2 Kapsam

Bu proje kapsamında tüm kullanıcı verilerinin, işlemlerin ve ilgili bilgilerin depolanması için merkezi bir veritabanı olan SQL, kullanıcıların veritabanıyla etkileşim kurabileceği görsel kullanıcı arayüzü olarak Windows Forms Uygulaması kullanılmaktadır.

Bu yazılım ürünü aşağıdaki özellikleri yapabilecek kapasitede tasarlanmıştır.

- Kullanıcı bilgilerini güvenli bir şekilde saklama ve yönetme.
- Kullanıcılar için görsel olarak çekici ve kullanıcı dostu bir arayüz sağlama.
- Veritabanındaki bilgilere çıkarma, güncelleme ve silme gibi işlemleri yapabilme.
- Kullanıcıya veri tabanına işlenen bilgileri kullanarak rapor oluşturabilme.

Bu yazılım ürünün sahip olmadığı özellikler aşağıdaki sunulmaktadır.

- Program yalnızca SQL Server üzerinde çalışacak şekilde tasarlanmıştır, farklı veritabanı sistemleri desteklenmeyecektir.
- Windows Forms teknolojisi kullanıldığı için program yalnızca Windows işletim sistemlerinde çalışabilir.
- Aynı anda çok yüksek sayıda kullanıcıya hizmet vermek için optimize edilmemiştir, küçük ölçekli kullanıcı grupları için tasarlanmıştır.

1.3 Kısaltmalar

SQL : Structured Query Language

1.4 Tanıtım

Bu program kullanıcıların günlük yeme alışkanlıklarını düzenli bir şekilde takip etmelerine ve sağlıklı yaşam hedeflerine ulaşmalarına yardımcı olmak için tasarlanmış yenilikçi bir kalori takip uygulamasıdır. Windows Forms tabanlı bu uygulama, kullanıcıların kolaylıkla kayıt olmasını ve giriş yaparak yeme alışkanlıklarını detaylı şekilde takip edebilmesini sağlar.

Kullanıcılar, günlük yediklerini öğün bazında kaydedebilir ve tüketilen kaloriyi görüntüleyerek bilinçli seçimler yapabilir. Bunun yanı sıra, uygulama kullanıcıya kendi tariflerini ve yemek bilgilerini sisteme ekleyebilme esnekliği sunar. Program kullanıcıların sağlık hedeflerini daha iyi planlamalarına ve analiz etmelerine yardımcı olacak üç tür raporlama özelliği ile desteklenmiştir.

Gün Sonu Raporu: Kullanıcılar, gün sonunda tüketilen tüm besinleri ve toplam kalori miktarını detaylı bir şekilde görüntüleyebilir.

Kıyas Raporu: Kullanıcılar, belirli bir tarih aralığında diğer kullanıcılarla toplam kalori tüketimlerini karşılaştırabilir ve kendilerini değerlendirme imkanı bulur.

Yemek Çeşidi Raporu: Program, tüm kullanıcıların belirli bir tarih aralığında hangi öğünde hangi yemekleri en çok tercih ettiğini analiz eder ve raporlar.

Bu yazılım programı, hem bireysel hem de toplu kullanıcı verilerini güvenli bir şekilde saklar ve analiz eder. Kullanıcı dostu arayüzü, detaylı rapor seçenekleri ve esnek veri giriş özellikleriyle, sağlıklı yaşam yolculuğunuzda yanınızda olacak güçlü bir araçtır. Bu uygulama, yalnızca günlük kalorilerinizi takip etmekle kalmaz, aynı zamanda sağlıklı alışkanlıklar geliştirmenizi destekler.

1.5 Hedef ve Başarı Kriterleri

Bu programın temel hedefi, kullanıcıların günlük yeme alışkanlıklarını kolay ve etkili bir şekilde takip etmelerine olanak tanıyan, kullanıcı dostu ve güvenilir bir platform sunmaktır. Program, bireylerin sağlıklı yaşam hedeflerine ulaşmalarını destekleyecek özellikler ve analiz araçları ile donatılmıştır.

Belirlenen başarı kriterleri aşağıdaki gibidir:

1. Kullanıcı Kayıt ve Giriş İşlevselliği: Kullanıcılar, sorunsuz bir şekilde kayıt olup programa giriş yapabilmelidir. Giriş işlemleri güvenli ve hızlı bir şekilde gerçekleştirilmelidir.

2. Öğün ve Kalori Takibi: Kullanıcılar, tükettikleri yemekleri öğün bazında kolayca sisteme ekleyebilmelidir. Program, bu verileri doğru bir şekilde kaydetmeli ve günlük toplam kalori bilgilerini hesaplayabilmelidir.
3. Esnek Veri Girişi: Kullanıcılar, kendi tariflerini ve yemek bilgilerini sisteme ekleyerek kişiselleştirilmiş bir deneyim yaşamalıdır.
4. Raporlama Fonksiyonları:
 - Gün Sonu Raporu: Kullanıcıların, gün sonunda yedikleri yiyecekleri ve toplam kaloriyi detaylı olarak görebilmesi.
 - Kıyas Raporu: Kullanıcıların, belirli bir tarih aralığında diğer kullanıcılarla kalori tüketimlerini karşılaştırabilmesi.
 - Yemekler Çeşidi Raporu: Kullanıcıların, genel tüketim trendlerini öğrenebilmesi ve belirli tarih aralıklarında hangi yemeklerin öne çıktığını görebilmesi.
5. Kullanıcı Dostu Arayüz: Programın arayüzü, kullanıcıların kolaylıkla anlayabileceği ve kullanabileceği şekilde tasarlanmıştır. Karmaşık işlemler, basit ve erişilebilir yöntemlerle gerçekleştirilmelidir.
6. Veri Güvenliği ve Performansı: Kullanıcıların kişisel verileri ve beslenme alışkanlıkları, yüksek güvenlik standartlarıyla korunmalıdır. Programın performansı, yüksek veri hacminde bile hızlı ve kesintisiz işlem yapmaya uygun olmalıdır.

Bu hedeflere ulaşılması, yazılımın işlevselliğinin ve kullanıcı memnuniyetinin sağlanması açısından temel başarı kriterlerini oluşturur. Programın yaygın bir şekilde benimsenmesi ve sağlıklı yaşam hedeflerine katkıda bulunması, projenin başarısını belirleyecek en önemli faktörlerdir.

1.6 Referanslar

N/A

2. Fonksiyonel Gereksinimler

Bu bölüm, yazılımın temel fonksiyonlarını yerine getirebilmesi için gerekli olan yapısal, veri ve işlevsel gereksinimleri açıklamaktadır. Yazılım, çok katmanlı mimari yapısına uygun olarak tasarlanmış, modern geliştirme prensipleri benimsenmiştir ve güvenlik önlemleri ile desteklenmiştir.

2.1Yapısal

Bu yazılım, çok katmanlı mimari bir yapıda geliştirilmiş olup aşağıdaki yapısal özelliklere sahiptir:

- Katmanlı Mimari:
 - Repository ve Servis Katmanları: Veri erişim işlemleri Repository deseniyle gerçekleştirilir. İş mantığı ise servis katmanında işlenerek esneklik ve yeniden kullanılabilirlik sağlanır.
 - DTO'lar ve ViewModel'ler: Verilerin kullanıcı arayüzü ve servis katmanı arasında taşınması için Data Transfer Objects (DTO) ve kullanıcı arayüzüne özel ViewModel'ler kullanılır.
 - Mapping Profilleri: Veri dönüşümleri için AutoMapper veya benzeri bir araç ile mapping profilleri oluşturulmuş, ViewModel, DTO ve veri varlıkları arasında dönüşüm işlemleri otomatize edilmiştir.
- Kimlik Doğrulama ve Yetkilendirme:
 - Kullanıcı bilgileri güvenli bir şekilde saklanır; şifreler hashlenerek veritabanında tutulur. Hashleme algoritması, yüksek güvenlik standartlarına uygun olarak seçilmiştir (örneğin, SHA-256 veya bcrypt).
- Raporlama Modülü: Kullanıcıların yeme alışkanlıklarını analiz etmelerini sağlayacak üç tür raporlama desteği sunar (Gün Sonu, Kıyas, Yemek Çeşidi).
- Esnek Veri Yapısı: Kullanıcılar, kendi istedikleri yemek bilgilerini sisteme kolayca ekleyebilir.

2.2 Veri Gereksinimi

Programın işlevselliğini sağlamak için gerekli olan veri gereksinimleri:

- Kullanıcı Verileri:
 - Kullanıcı e-posta adresi ve hashlenmiş şifre gibi kimlik bilgileri.
- Yemek Verileri:
 - Yemek adı, porsiyon bilgisi, kalori değeri, öğün türü ve eklenme tarihi.
- Öğün Verileri:
 - Öğün adı (kahvaltı, öğle, akşam, ara öğün) ve bu öğünlere ait yemek kayıtları.
- Raporlama Verileri:

- Kullanıcıların ve genel sistem verilerinin analiz edilmesi için belirli tarih aralıklarındaki yemek ve kalori tüketim bilgileri.

2.3 İşlevsel

Programın sağlaması gereken temel işlevler:

- Kayıt ve Giriş İşlevleri:
 - Kullanıcılar sisteme kayıt olabilir, giriş yapabilir ve şifrelerini sıfırlayabilir.
 - Şifreler hashlenerek güvenli bir şekilde saklanır. Şifre sıfırlama işlemleri e-posta doğrulaması ile gerçekleştirilir.
- Öğün ve Kalori Takibi:
 - Kullanıcılar, her öğün için yedikleri yemekleri kaydedebilir.
 - Sistem, öğün bazında ve günlük toplam kalori tüketimini otomatik olarak hesaplar.
- Yemek Ekleme ve Düzenleme:
 - Kullanıcılar, kendi istedikleri yemekleri ve kalori bilgilerini sisteme ekleyebilir.
- Raporlama İşlevleri:
 - Gün Sonu Raporu: Kullanıcılar, gün sonunda yedikleri yiyeceklerin listesini ve toplam kalori tüketimlerini görüntüleyebilir.
 - Kıyas Raporu: Kullanıcılar, belirli bir tarih aralığında diğer kullanıcılarla kalori tüketimlerini karşılaştırabilir.
 - Yemek Çeşidi Raporu: Program, genel kullanıcı trendlerini analiz ederek, belirli tarih aralıklarında en çok tüketilen yemekleri öğün bazlı raporlar.
- Mapping ve Veri Dönüşümü:
 - Mapping profilleri, DTO'lar ve ViewModel'ler arasında veri dönüşüm işlemlerini yönetir ve tutarlılık sağlar.
- Kullanıcı Dostu Arayüz:
 - Arayüz, kullanıcıların kolayca yemek ekleyebilmesini, raporları görüntüleyebilmesini ve geçmiş verilerini analiz edebilmesini sağlar.

Bu güncellenmiş yapısal, veri ve işlevsel gereksinimler, yazılımın kullanıcı ihtiyaçlarını güvenli, etkili ve modern bir yaklaşımla karşılamasını sağlamaktadır. Kullanıcı dostu tasarımı ve güçlü altyapısı ile bu program, sağlıklı yaşam hedeflerine katkıda bulunmayı amaçlar.

3. Fonksiyonel Olmayan Gereksinimler

Bu bölüm, yazılımın işlevsellik dışında karşılaması gereken gereksinimleri kapsamaktadır. Yazılımın performansı, güvenliği, kullanılabilirliği ve izlenebilirliği gibi yönler ele alınarak, kullanıcı deneyimini ve yazılım kalitesini artırmaya yönelik standartlar belirtilmiştir.

3.1 Performans/Hız

- Program, kullanıcı işlemlerine hızlı yanıt verecek şekilde optimize edilmiştir. Veritabanı sorguları, Repository ve Service katmanları ile verimli bir şekilde çalışır.
- Raporlama işlemleri, belirli tarih aralıklarındaki verilerin analiz edilmesi ve işlenmesi sırasında optimize edilmiş algoritmalarla hızlıca tamamlanır.

3.2 İzlenebilirlik

N/A

3.3 Kullanılabilirlik

- Programın arayüzü, kullanıcı dostu olacak şekilde tasarlanmıştır. Windows Forms tabanlı bu arayüz, minimal tasarımı ve sezgisel menü yapısı sayesinde kullanıcıların kolayca işlem yapmasını sağlar.
- Kullanıcıların kolayca kayıt, giriş ve yemek ekleme gibi işlemleri gerçekleştirebilmesi için net yönlendirmeler ve açıklamalar bulunmaktadır.
- Raporlar açık, anlaşılır bir formatta sunulmakta ve kullanıcıların ihtiyaç duyduğu veriye hızlıca ulaşmasını sağlamaktadır.

3.4 Gizlilik Gereksinimleri

- Kullanıcı şifreleri, hashlenerek veritabanında saklanır. Şifreleme algoritması güvenlik standartlarına uygun olarak uygulanmıştır.

3.5 Desteklenebilirlik

- Program, hem mevcut işlevlerin sürdürülebilirliğini hem de gelecekte eklenebilecek yeni özellikleri destekleyecek şekilde tasarlanmıştır.

<BiteSizedDietPlan >

- Katmanlı mimari yapı, yazılımın bakımını kolaylaştırır ve modüler geliştirme imkanı sağlar.
- Kullanıcı geri bildirimlerine göre yeni özelliklerin eklenmesi veya mevcut özelliklerin iyileştirilmesi kolayca yapılabilir.

3.6 Arayüz

- Arayüz, Windows Forms kullanılarak geliştirilmiştir ve kullanıcıların yemek ekleme, rapor görüntüleme ve diğer işlemleri hızlı ve kolay bir şekilde yapmasını sağlar.
- Raporlama modülü, kullanıcıların tüm bilgileri liste formatında kolayca incelemesini sağlar.

3.7 Güvenlik

N/A

4. SİSTEM MODELLERİ

4.1 Aktörler

AKTÖR	ROLLER
Bireysel Kullanıcı	Giriş yapma
	Kullanıcı kaydı oluşturma
	Yemek ekleme
	Gün sonu raporu görüntüleme
	Kıyas raporu görüntüleme
	Yemek Çeşidi raporu görüntüleme
Sistem Yöneticisi	Kullanıcı yönetimi
	Sistem ayarlarını değiştirme
	Sistemde oluşan hataların düzeltilmesi
	Sürekli geliştirme ve iyileştirme
Veritabanı Yöneticisi	Veritabanı yedekleme
	Veritabanı kurtarma
	Performans izleme

4.2 Senaryolar

Bu başlık altında, yazılımın nasıl kullanılacağını açıklayan senaryolar tanımlanmaktadır. Senaryolar, kullanıcıların programla nasıl etkileşimde bulunduğunu adım adım gösterir. Genellikle kullanıcı hedeflerini ve bu hedeflere ulaşmak için izlenen yolu içerir. Örnek senaryolar aşağıdaki gibidir:

Senaryo 1: Kullanıcı Giriş Yapıyor

1. Kullanıcı, programın giriş ekranını açar.
2. E-posta ve şifresini ilgili alanlara girer.
3. Giriş butonuna tıklar.
4. Sistem, şifreyi hashleyerek veritabanında kontrol eder.
5. Eğer bilgiler doğruysa, kullanıcı ana sayfaya yönlendirilir.

Senaryo 2: Yemek Ekleme

1. Kullanıcı, ana ekrandaki “Yemek Ekle” butonuna tıklar.
2. Karşısına bir form açılır.
3. Yemek adı, öğün (sabah, öğle, akşam, atıştırmalık), kalori miktarı ve yemek resmi bilgileri doldurur.
4. “Ekle” butonuna tıklar.
5. Sistem, bu bilgileri veritabanına ekler ve kullanıcıya başarılı bir işlem mesajı gösterir.

Senaryo 3: Raporu Görüntüleme

1. Kullanıcı, ana ekrandaki “Gün Sonu Raporunu Görüntüle, Yemek Çeşidi Raporunu Görüntüle, Kıyas Raporunu Görüntüle ” sekmelerinden birine tıklar.
2. İstenilen tarih yada tarih aralığı seçilir.
3. Sistem, ilgili bilgileri kullanıcıya listeler.
4. Kullanıcı bu veriyi inceleyebilir.

Senaryo 4: Öğün Ekleme

<BiteSizedDietPlan >

1. Kullanıcı, ana ekranında öğün eklemek istediği tarih ve öğün türünü seçer.
2. "Öğün Ekle" butonuna tıklar.
3. Ana ekranın sağ tarafında yer alan yemek seçeneklerinden seçmek istenilen öğünün üzerine tıklanır.
4. Seçilen yemek ana ekranın alt tarafında gözüken listede gözüktür.
5. "+", "-" butonları ile porsiyon arttırılıp azaltılabilir.
6. Sil seçeneği ile istenilen öğün ve yemek seçildikten sonra "SİL" butonuna tıklanarak silinir.

4.3 Use-Case Modelleri

Aşağıda, aktörün sistemi nasıl kullandığı ve sistemin bu taleplere nasıl cevap verdiği göstermektedir. Aşağıda bu yazılım için örnek Use-Case açıklamaları ve modelleri verilmiştir:

Use-Case 1:

- Aktör: Kullanıcı
- Ön Koşul: Kullanıcının programa daha önce kayıtlı olması gerekir.
- Adımlar:
 1. Kullanıcı, giriş bilgilerini girer ve gönderir.
 2. Sistem, bilgileri doğrular.
 3. Kullanıcı başarılı bir şekilde sisteme giriş yapar.
- Başarı Durumu: Kullanıcı ana ekrana yönlendirilir.
- Hata Durumu: Hatalı giriş bilgilerinde kullanıcıya hata mesajı gösterilir.

Use-Case 2:

- Aktör: Kullanıcı
- Ön Koşul: Kullanıcının giriş yapmış olması gerekir.
- Adımlar:
 1. Kullanıcı, yeni yemek ekleme ekranına gider.
 2. Yemek bilgilerini doldurur ve kaydeder.

<BiteSizedDietPlan >

3. Sistem, bilgileri veritabanına ekler ve başarı mesajı gösterir.

- Başarı Durumu: Yeni yemek başarıyla kaydedilir.
- Hata Durumu: Eksik veya hatalı bilgi girilmesi durumunda kullanıcıya uyarı mesajı gösterilir.

5. Genel

5.1 İş Kısıtları

İş kısıtları bölümleri, yazılımın geliştirilmesi, dağıtılması ve işletilmesi sırasında göz önünde bulundurulması gereken sınırlamaları ve şartları içermektedir. Bu proje için iş kısıtları aşağıdaki şekilde tanımlanabilir:

1. Teknolojik Kısıtlar:

- Platform Bağımlılığı: Uygulama yalnızca Windows işletim sistemi üzerinde çalışacak şekilde tasarlanmıştır.
- Veritabanı Teknolojisi: SQL Server kullanımı zorunludur ve başka bir veritabanı sistemi için destek sağlanmamaktadır.
- Katmanlı Mimari: Uygulama, katmanlı mimari yapısına uygun olarak geliştirilmiştir. Bu nedenle, katmanlar arasında herhangi bir doğrudan erişim kısıtlanmıştır ve yalnızca belirlenen servisler aracılığıyla iletişim sağlanmaktadır.
- Teknolojik Bileşenler: Uygulama, C# dili ve Windows Forms teknolojisiyle geliştirilmiştir. Kullanıcı deneyimi ve görünüm açısından sınırlamalar bulunmaktadır.

2. Güvenlik Kısıtları:

- Şifreler hashlenerek saklanmaktadır. Ancak, ileri düzey saldırılara karşı tam kapsamlı güvenlik önlemleri (örneğin, çok faktörlü kimlik doğrulama) bu sürümde sağlanmamıştır.

3. Kullanıcı Kısıtları:

- Uygulama, bireysel kullanıcılar tarafından kullanılmak üzere tasarlanmıştır.
- Sistem yöneticisi dışında kullanıcılar, programın işleyişinde yapısal değişiklik yapamazlar (örneğin, yemek türleri veya öğün kategorileri gibi sabit veriler düzenlenemez).

4. Donanım Kısıtları:

- Uygulama, düşük sistem kaynaklarına sahip bilgisayarlarda da çalışacak şekilde optimize edilmiştir.

5. Genişletilebilirlik Kısıtları:

- Sistem, yeni özelliklerin entegre edilebilmesi için esnek bir yapıda olsa da, mevcut sürümde tüm kullanıcıların ihtiyaçlarını karşılamayı hedefleyen genel bir yapıya sahiptir.
- Gelecekteki genişletme çalışmalarında mevcut mimarinin yeniden yapılandırılması gerekebilir.

6. Teknik

6.1 Donanım

Uygulamanın çalıştırılabilmesi için minimum ve önerilen donanım gereksinimleri aşağıdaki gibidir:

Minimum Donanım Gereksinimleri

- İşlemci: Intel Core i3 veya dengi.
- RAM: 4 GB.
- Depolama: 500 MB boş disk alanı (veritabanı ve uygulama dosyaları için).
- Ekran Çözünürlüğü: 1280x720 piksel.
- İşletim Sistemi: Windows 10 veya daha üstü.

Önerilen Donanım Gereksinimleri

- İşlemci: Intel Core i5 veya üstü.
- RAM: 8 GB veya daha fazla.
- Depolama: 1 GB boş disk alanı (yüksek veri yoğunluğu için).
- Ekran Çözünürlüğü: 1920x1080 piksel veya daha yüksek.

<BiteSizedDietPlan >

- İşletim Sistemi: Windows 10/11 Pro.

Bu gereksinimler, uygulamanın istikrarlı çalışmasını sağlamak için belirlenmiştir. Daha düşük donanımlarda uygulama performansında düşüş yaşanabilir.

6.2 Yazılım

Geliştirme Ortamı

- Programlama Dili: C#
- IDE: Microsoft Visual Studio 2022 veya daha güncel bir sürüm.
- Veritabanı: Microsoft SQL Server 2019 veya üstü.
- Bağımlılıklar: NuGet paket yöneticisi aracılığıyla yüklenen kütüphaneler (örneğin, Entity Framework Core).

Çalışma Ortamı

- İşletim Sistemi: Windows 10 veya daha yeni bir sürüm.
- .NET Runtime: .NET Framework 4.8 veya .NET 6 Runtime (tercihe bağlı).
- Veritabanı Sunucusu: Microsoft SQL Server Express (geliştirme/test için) veya SQL Server Standard/Enterprise (üretim için).
- Yedekleme ve Geri Yükleme Araçları: SQL Server Management Studio (SSMS).

Üçüncü Taraf Araçlar ve Bileşenler

- Hashing ve Güvenlik: Şifrelerin hashlenmesi için System.Security.Cryptography.
- DTO ve Mapping: AutoMapper kütüphanesi.
- UI: Windows Forms teknolojisi kullanılmıştır.

7.Transitional

7.1 Arşivleme

N/A