



Yiğit Öner

H5210016

Yazılım Geliştirme Ortam ve Araçları

Projenin Tüm Süreçleri ve Adımları

Milletvekili Seçimi Hakkında Genel Bilgi

2017 Türkiye anayasa değişikliği referandumu ile ülkenin yönetim şekli başkanlık sistemi olarak belirlendi. 16 Mart 2018 tarihinde, 26 maddelik "Seçimlerin Temel Hükümleri ve Seçmen Kütükleri Hakkında Kanun ile Bazı Kanunlarda Değişiklik Yapılmasına Dair Kanun" yürürlüğe girdi. Kanunla, seçimlere katılma yeterliliği taşıyan siyasi partilerin ittifak yaparak milletvekili seçimine katılmalarına imkân tanınırken, siyasi partilerin seçimlerde başka bir siyasi partiyi destekleme kararı almalarını yasaklayan hüküm kaldırıldı.

Kasım 2017'de Milliyetçi Hareket Partisi Genel Başkanı Devlet Bahçeli'nin Adalet ve Kalkınma Partisi ile ittifak yapmayı önermesi sonucunda Şubat 2018'de Cumhur İttifakı kuruldu.

Seçimin normal şartlarda 3 Kasım 2019'da yapılması gerekiyordu ancak, Bahçeli'nin 17 Nisan 2018 tarihinde partisinin TBMM'deki grup toplantısında erken seçimi gerekli gördüğünü belirtmesi ve Cumhurbaşkanı Recep Tayyip Erdoğan'ın ertesi gün buna olumlu yaklaştığını açıklaması ile seçim tarihi erkene alındı. Aynı gün olağanüstü hâl seçim tarihini de içine alacak şekilde üç ay daha uzatıldı. 20 Nisan 2018'de erken seçim önerisi Türkiye Büyük Millet Meclisi'nde kabul edildi.

Çıkan Kanunun 4. maddesine göre; milletvekillerinin seçim çevrelerine dağılımı yapılırken; toplam milletvekili sayısından (600) önce her ile bir milletvekili tahsis edilir. Son genel nüfus sayımı ile belli olan Türkiye nüfusu, kalan milletvekili sayısına (519) bölünmek suretiyle bir sayı elde edilir. İl nüfusunun bu sayıya bölünmesi ile her ilin ayrıca çıkaracağı milletvekili sayısı tespit olunur. Bu suretle hesaplanan milletvekillerinin sayısı 600'ü bulmadığı takdirde, nüfusu; milletvekili çıkarmaya yetmeyen illerin nüfusları ile artık nüfus bırakan illerin, artık nüfusları büyüklüklerine göre sıraya konulur ve ilk hesapta iller arasında bölüştürülmemiş bulunan milletvekilleri bu sıra esas alınarak dağıtıma tabi tutulur. Yapılan tespit sonunda çıkaracağı milletvekili sayısı (18)'e kadar olan iller bir seçim çevresi sayılır. Çıkaracağı milletvekili sayısı (19)'dan (35)'e kadar olan iller iki, (36) ve daha fazla olan iller üç seçim çevresine bölünür. Bu seçim çevreleri numara sırasına göre adlandırılır. 2839 sayılı Kanunun 4. maddesinin yukarıda belirtilen hükümleri ile 2016 yılı nüfus bilgileri göz önünde bulundurulmak suretiyle, 600 milletvekilinin 81 ile dağılımı yapılmıştır. İllerin çıkardığı milletvekili sayıları aşağıdaki gibidir: ([Wikipedia'dan alıntı yapılmıştır](#)).

Seçim çevresi	MŞ	Seçim çevresi	MŞ	Seçim çevresi	MŞ	Seçim çevresi	MŞ	Seçim çevresi	MŞ	Seçim çevresi	MŞ	Seçim çevresi	MŞ	Seçim çevresi	MŞ
Adana	15	Artvin	2	Bursa	20	Erzurum	6	İstanbul (III)	35	Kırklareli	3	Muş	4	Tekirdağ	7
Adıyaman	5	Aydın	8	Bursa (I)	10	Eskişehir	7	İzmir	28	Kırıkkale	3	Nevşehir	3	Tokat	5
Afyonkarahisar	6	Balıkesir	9	Bursa (II)	10	Gaziantep	14	İzmir (I)	14	Kırşehir	2	Niğde	3	Trabzon	6
Ağrı	4	Bartın	2	Çanakkale	4	Giresun	4	İzmir (II)	14	Kocaeli	13	Ordu	6	Tunceli	2
Aksaray	4	Batman	5	Çankırı	2	Gümüşhane	2	Kahramanmaraş	8	Konya	15	Osmaniye	4	Şanlıurfa	14
Amasya	3	Bayburt	1	Çorum	4	Hakkâri	3	Kars	3	Kütahya	5	Rize	3	Şırnak	4
Ankara	36	Bilecik	2	Denizli	8	Hatay	11	Kastamonu	3	Malatya	6	Sakarya	7	Uşak	3
Ankara (I)	13	Bingöl	3	Diyarbakır	12	Iğdır	2	Karabük	3	Manisa	10	Samsun	9	Van	8
Ankara (II)	11	Bitlis	3	Düzce	3	Isparta	4	Karaman	3	Mardin	6	Silirt	3	Yalova	3
Ankara (III)	12	Bolu	3	Edirne	4	İstanbul	98	Kayseri	10	Mersin	13	Sinop	2	Yozgat	4
Antalya	16	Burdur	3	Elazığ	5	İstanbul (I)	35	Kilis	2	Muğla	7	Sivas	5	Zonguldak	5
Ardahan	2			Erzincan	2	İstanbul (II)	28								

Proje Tanım ve Amaç

Projenin araştırma konusu olan Kayseri Milletvekilleri ; sayısı , milletvekillerinin belgeleri ve gerekli bilgilerine ulaşabilmek için araştırma sürecini Arama motorlarından araştırmalar sonucunda dahili ve kapsamlı bilgilerin bulunduğu [Vikipedi](#) sitesinden yararlanıldı. Bilgilerin gerçek veriler ile uyuşması adına [CNN Türk](#), [Sabah](#) gazetesi gibi denk haber sitelerinden, [YSK](#) gibi gerekli karar merceği ile geçmiş ve bu seçimdeki veriler elde edildi ve süreç buna göre uyarlandı.

Proje'nin Tanımı : Türkiye Cumhuriyeti Milletvekillerinin belirleneceği 14 Mayıs 2023 Seçimlerinin simülasyonudur(Taslak). Her Partinin belli bir sayıda kontenjan sınırı olması ile birlikte her parti kendi içinde adayları aday listesi ve aday sayılarını kontenjan sayısı artmayacak şekilde belirleyebilmektedir.

Proje'nin amacı: Türkiye Cumhuriyeti Vatandaşlarının Milletvekili seçimlerinde oy verdikleri partinin belirlemiş olduğu Adayların Meclise girebilmesini mümkün kılan bu seçimin simülasyonudur. Bu yapı sayesinde oy damgası üzerinde geçerli olan pusulaların kaybolma , geçersiz hale gelmesi vs. gibi durumlara karşı önlem alınmış olunacaktır. Ayrıca kullanıcıların güncel Oy istatistiğini , hangi partiden hangi Milletvekillerin olduğunu görebilecektir.

Kullanım tasarımı şu şekilde olması planlanmıştır: Oy sandığı başındaki gerekli yetkili merci sisteme giriş yaparak oy sayım anında hangi partide kaç oy almış ise bunu sisteme kayıt ederek resmi hale getirmiş olacaktır. Daha sonrasında Türkiye Cumhuriyeti Vatandaşları sistemde güncel oyları görecektir (Hangi partiye kaç oy ve Toplam Kullanılan oy sayısı).

Kullanılan Yazılımın Süreç Model Tanımı

Kullan Yazılım süreç modeli Waterfall'dır. Şelale yönteminde (yaygın kullanılan adı WaterFall Model) yazılım geliştirme süreci analiz, tasarım, kodlama, test, sürüm ve bakım gibi safhalardan oluşur. Geleneksel yazılım metodlarında bu safhalar şelale modelinde olduğu gibi doğrusal olarak işler. Her safha, başlangıç noktasında bir önceki safhanın ürettiklerini bulur. Kendi bünyesindeki değişiklikler doğrultusunda teslim aldıklarını bir sonraki safhanın kullanabileceği şekilde değiştirir.

Özellikleri

- Şelalenin her basamağında yer alan aktiviteler eksiksiz olarak yerine getirilir. Bu bir sonraki basamağa geçmenin şartıdır.
- Her safhanın sonunda bir doküman oluşturulur. Bu yüzden şelale modeli doküman güdümlüdür.
- Yazılım süreci doğrusaldır, yani bir sonraki safhaya geçebilmek için bir önceki safhada yer alan aktivitelerin tamamlanmış olması gerekir.
- Kullanıcı katılımı başlangıç safhasında mümkündür. Kullanıcı gereksinimleri bu safhada tespit edilir ve detaylandırılır. Daha sonra gelen tasarım ve kodlama safhalarında müşteri ve kullanıcılar ile diyaloga girilmez.

Modelin Getirdiği Problemler

- Safhaların birbirinden kesin olarak ayrı tutulmaları gerçekçi değildir. Projelerde safhalar arasındaki bu sınırlar yok olabilir.
- Teoride safhalar birbirlerini takip ederler. Projelerde bunun bazen mümkün olmadığını ve önceki safhalara geri dönmek zorunda kalındığını görebiliriz.
- Safhalar arası geri dönüş yetersizdir. Model değişikliğe açık değildir.
- Müşteri gereksinimlerinin proje öncesi detaylı olarak kağıt üzerinde oluşturulması ilerde sorun yaratabilir. Müşteri gereksinimleri değişikliğe uğrayabileceği için, yazılım sisteminin de yapısal değişikliğe uğraması kaçınılmaz olabilir. Böyle bir durum maliyeti artırır, çünkü yeni ve değişen gereksinimleri implemente edebilmek için modelde yer alan safhaların birkaç kere uygulanması gerekebilir.
- Sistemin kullanılır hale gelmesi uzun zaman alabilir.
- Başlangıçta yapılan hataların tespiti çok uzun zaman alabilir. Bu hataların giderilmesi maliyeti yükseltir.
- Modül implementasyonları için zaman tahminleri proje planlarını oluşturan yöneticiler tarafından yapılır. Teknik bilgiye sahip olmayan şahıslar tarafından yapılan bu tahminler çoğu zaman doğru değildir. Bu durum proje planlama sürecini negatif etkiler.

1.Çözümleme

Sistem ihtiyaç ve işlevlerinin ayrıntılı bir şekilde çıkarıldığı ve belirli kurallar çerçevesinde kağıda döküldüğü aşamasıdır. Geliştirilecek olan yazılım ürününden tam olarak ne istendiği net bir şekilde belirlenir.

2.Tasarım

Belirlenen istekler çerçevesinde yazılım sisteminin temel yapısının oluşturulduğu aşamadır.

3.Gerçekleştirme

Kodlama, test etme ve kurulumun yapıldığı aşamadır:

3.1)Kodlama

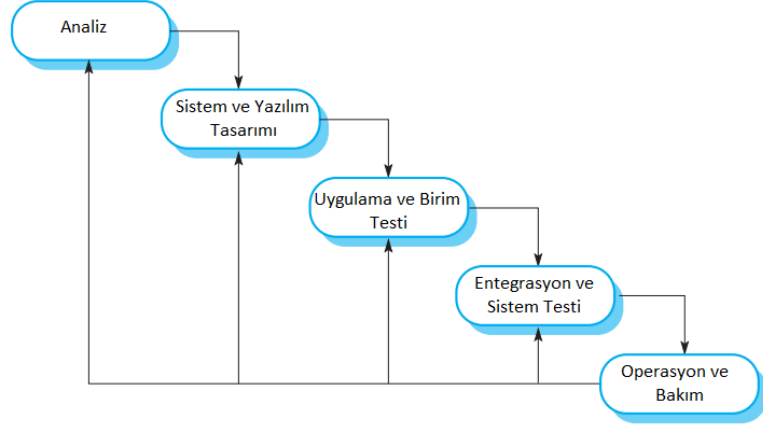
Oluşturulan yazılım ürünün programlama aşamasıdır. Basit ,okunabilir ve bakımı kolay olan kodlama tekniğiyle programlama yapılmalıdır.

3.2)Test Etme

Önceden belirlenen ihtiyaçları karşılayıp karşılamadığı ,doğru çıktı üretip üretmediği testleri yapılarak sistemdeki açıklıklar kapatılmalıdır.

4.Teslim ve Bakım

Tüm test aşamaları tamamlandıktan sonra ürün müşteriye teslim edilir. Yazılım ürünü kullanılmaya başlandıktan sonra bakım aşaması da başlar. Ürün kullanıldıktan sonra yazılımın desteklenip desteklenmemesine göre hata giderilir ya da yeni eklentiler yapılır.



Yazılımın Süreç Kontrolü ve Detayları (Adımları)

Uygulamanın neye ihtiyaç olduğunu belirlemek amacı ile benzer uygulamaların raporlarına ulaşılıp karşılaştırma yapıldıktan sonra gerekli sitelerden süreçlerin araştırması yapılmıştır. Daha sonrasında **Waterfall** modeli seçilmiştir. Nedeni uygulama tekdüze ve yekpare olduğu için kullanıcıların neye ihtiyacı olduğunu paydaşlar tarafından belirlenmiştir.

ÇÖZÜMLEME

Türkiye Cumhuriyeti'nin Kayseri İline ait "Milletvekili Seçimi" için gerek araştırmalar yapıldı. Milletvekili Seçimlerinde ihtiyaç olan kişiler öğelerin neler olduğu niçin olduğu belirlendi. İhtiyaçlar doğrultusunda yapılar çıkarıldı. İhtiyaç Duyulan Yapılar (Ekranlarla Yekpare açıklama).

1) Giriş Ekranı

Sistemde tüm yetkili, Parti görevlilerinin sisteme girip işlem yapmasını sağladığı ekrandır. Veriler sistemde mevcutsa ilgili ekrana yönlendirilir.

2) Güncel verilerin olduğu Ekranı

Sisteme giriş yapmadan da etkileşim olan sayfa tüm Türkiye Cumhuriyeti Vatandaşları tarafından kurulabilecek ekran mevcut kullanılan oyları ve hangi partinin ne kadar oy aldığını gösterecektir. Geriye kalan saati ve tüm kayseri nüfusunu, Milletvekili kontenjanını gösterecektir.

3) Partilerin Olduğu Ekranı

Sisteme giriş yapabilen Şahsiyetler sisteme temsil ettikleri partinin bilgilerini sisteme ekleme işlemini yapar.

4) Seçili Partinin Milletvekilleri Ekranı

Sisteme giriş yapan Şahsiyetlerin ait oldukları partideki Milletvekillerinin bilgilerini ve Milletvekili Aday Listesinin ekleme yapıldığı yeri temsil eder.

5) Oy Girişinin yapıldığı Ekranı

Sisteme giriş yapabilen Seçim yetkilisi (Sandık Başkanı) Oy sayım anında sisteme oy ekleme işlemini yapar.

6) İtiraz Ekranı

Sisteme giriş yapabilen Parti temsilcisi(görevlisi) veya Sandık başkanının eğer bir "gayri resmi" veya oluşan bir problemten ötürü oluşan durumların sisteme eklenmesini sağlamaktadır.(saat, açıklama).

TASARIM

Bu adımda Çözümlemede yapılan işlemlerin tasarıma dökülme işlemini gerçekleştirilmesi durumunu ele almaktadır. Amaç uygulamayı kullanan kişilere yönelik basit ve kullanışlı yapıda olup kullanıcıların isteklerini karşılaması ön plandadır. Sistemde bulunacak yapılar, öğeler ve özelliklerin ne olduğu kararlaştırıldı. Kağıt ortamda UML Use Caseleri değerlendirip çizildikten sonra UML'deki sınıfların aralarındaki ilişkiler değerlendirilip düzenlendi.

GERÇEKLEME

Uygulamanın Hangi dil ile yazılacağını belirledikten sonra Dotnet(C#) ' den faydalanıp uygulamayı masaüstü şeklinde kodlandırılmıştır. Burada programlama dilleri arasında karşılaştırma yapılması ve kaynak veri olmasını araştırdıktan sonra kodlama kısmına geçilmiştir.

SINAMA

Bu adım da Kod yazılırken oluşan zafiyetleri ve hataların maliyetinin en az olması amacı ile yapılmıştır. Tasarım ve gerçekleştirme evresinde oluşan hatalar veya ekran tasarımlarının mantıksızlığı gibi yapılar revize edildi .gerekli mercilerin bu hataları veya tasarımları düzenlemesi ile uygulama teslim hazır evresine geçmeye hak kazanmıştır.(Sinama evresinde mevcut test senaryoları yardımı ile tekrar tekrar sinama evresine sokulmuştur.)

TESLİM VE BAKIM

Bu adım da ortaya çıkan Seçim uygulamasının çalışma faaliyetine geçtikten teslim hale gelmiştir. Yazılım ürünü kullanılmaya başlandıktan sonra bakım aşaması da başlar. 14 Mayıs 2023 Milletvekili seçim uygulaması kullanıma açıldıktan sonra yazılımın desteklenip desteklenmemesine gibi hatalar süreç içerisinde çözüme ivedilik ile kavuşturuldu ve yeni eklentiler yapıldı.

Proje Yazılım Süreç Modeli ve Nasıl Kullanıldığı

Bu kısımdaki konu başlığı projede hangi süreci kullanarak projenin oluşturulduğu ve yararlanılan modelin adımlarını nasıl izlediğim hakkında bilgileri içermektedir.(Bu konu başlığı altındaki metinler Kahraman bakış açısı ile yazılmıştır.

Projedeki yararlanılan süreç Waterfall (Şelale modelidir). Bu yapının seçilme nedeni büyük bir organizasyon halinde geliştirmede ve uygulamanın müşteri ile kontak içerisinde olmadan gerçekleştirilerek tasarlandığı için tekdüze ve yekparedir. Yani ürünü isteyen kişinin aşağı yukarı nasıl bir istekte bulunduğunu anladığım için Waterfall modelini tercih ettim.

Kullanılan Adımlar :

Çözümleme > Tasarım > Gerçekleme > Sınama > Bakım

I. Çözümleme

Çözümleme adımında benzer uygulamaların , temel ihtiyaçların ne olduğunu ve ne olması gerektiğini araştırdık. Hangi Şahıslar nesneler veya tiplerin olduğu belirlendi (**Giriş Ekranı ,Güncel Verilerin Olduğu Ekran , Partilerin olduğu Ekran , Seçili Partinin Milletvekilleri Ekranı , Oy Giriş Ekranı , İtiraz Ekranı**) .

Bu uygulamanın ana amacı ve hangi amaca hizmet etmesi gerektiği saptandı . Hangi Süreç tipinin kullanılması gerektiği paydaşlar tarafından tartışıldıktan sonra Süreç modeli belirlendi.

II. Tasarım

Bu Yazılım Sürecinde Hazırlanan plan ve akış doğrultusunda kullanılacak Sınıflar , ihtiyaçları karşılayacak roller , hangi programla dili hangi veri tabanı ve ne türde uygulama geliştirilecek gibi durumlar belirlenmesi halinde kağıt üzerine aktarıldı.

Burada kullanılacak tasarım yönetme sisteminde diyagramlar ile uygulamanın nasıl çalışması neye hizmet etmesi gerektiği yapıları düzenlendi . Daha sonrasında hangi tip veri maksimum ve minimum değerlerinin bulunduğu Tasarım gerçekleştirildi.

III. Gerçekleme

Bu Yazılım Sürecinde Tasarım adımında belirlenen yapının kodsız olarak gerçekleştirildiği yani yazılıma döküldüğü kısımdır . Bu kısımda hazırlanan UML Use Case'lerden yararlanarak uygulamayı hayata geçirme evresini gerçekleştirildi.

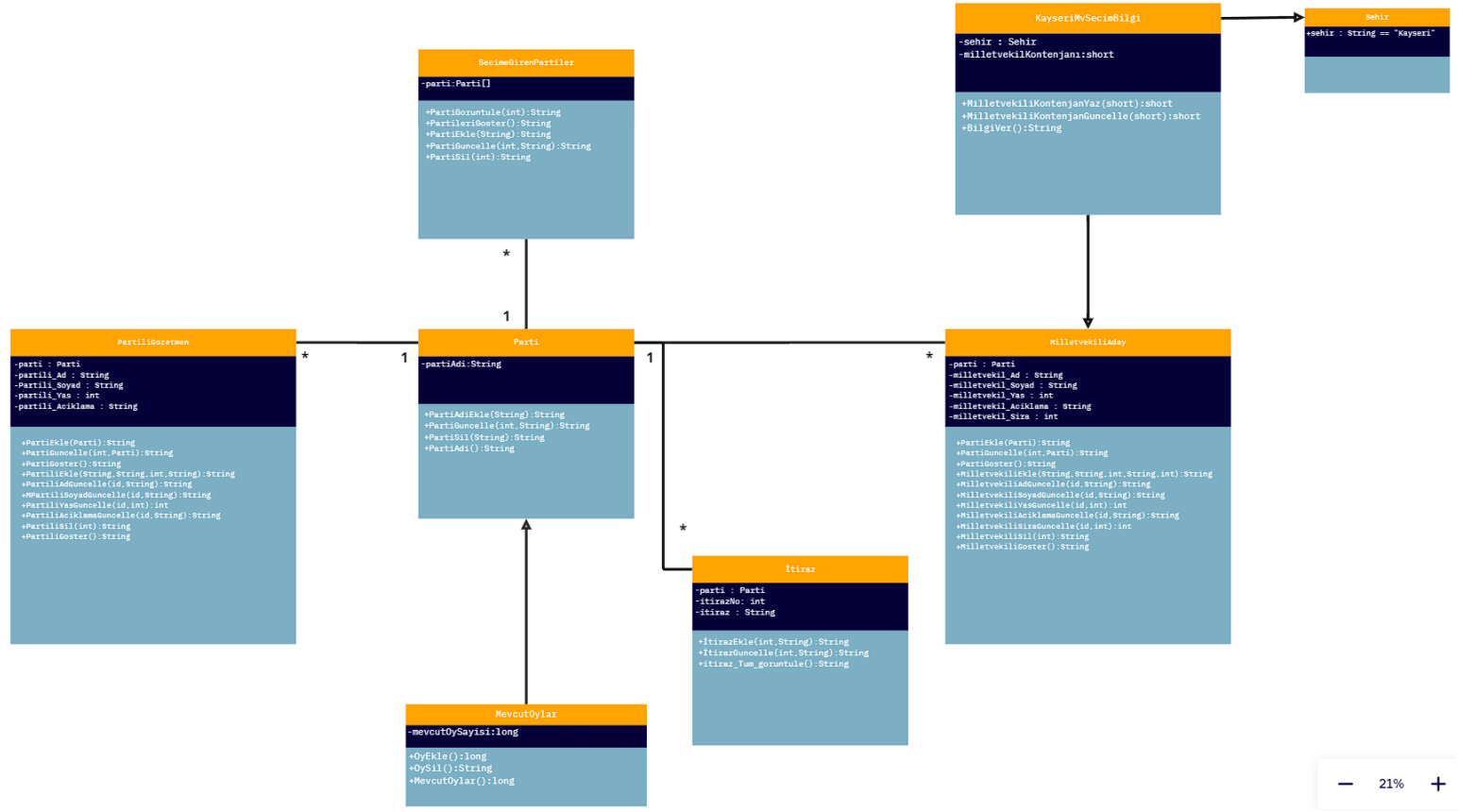
IV. Sınama & Test Etme

Bu Yazılım sürecinde Hazırladığım & hazırladığımız uygulamanın önceden(Tasarım Adımı) hazırlanan Test yapısını kullanarak oluşturulan uygulamayı testini yapacak paydaşlara ilettikten sonra uygulamadaki hataların olmadığına emin olduk .Daha sonrasında ekip gerekli bilgileri ilettikten sonra Uygulamanın hazır olduğunu ifade etti.

V. Bakım

Bu Yazılım sürecinde hazır olan ve hali hazırda piyasaya sunduğumuz Seçim uygulamasının bakımlarını ve optimizasyon , performans , verim gibi yapılarını süreç içinde takip edip fark ettiğimiz değişiklikleri eklediğimiz yama sayesinde düzelttik.

UML Diyagramı ve Açıklamaları



Yukarıda bulunan Sınıflar Uygulamanın ihtiyaç duyduğu UML sınıflarını temsil etmektedir.

I) Parti

Bu sınıf sayesinde sisteme parti hakkında veri girme gibi temel işlemleri yönetebiliyoruz.

Bu sınıf Seçime Giren partiler ile 1 - * ilişkisi içermektedir

Özellikler:

- PartiAdi : String

Metodlar:

- PartiAdiEkle(String) : String
- PartiAdiGuncelle(String) : String
- PartiAdiSil(String) : String
- PartiAdi():String

II) **SecimeGirenPartiler**

Bu sınıf sayesinde sistemdeki partilerin Listelenmesi ve tekrardan ekleme güncelleme vb işlemlerin yapılması , veri girme gibi işlemleri yönetebiliyoruz.

Bu sınıf parti sınıfı ile * - 1 ilişkisi içermektedir

Özellikler:

- parti: Parti []

Metotlar:

- PartiGoruntule(int) : String
- PartileriGoster():String
- PartiEkle(String):String
- PartiGuncelle(int,String):String
- PartiSil(int):String

III) **PartiliGozetmen**

Bu sınıf sayesinde sistemdeki Oy sayma işlemini gerçekleştiren partili gözetmen – personelin işlemlerini bulunduruyor.

Bu sınıf parti ile 1 - * (parti – Partili Gözetmen) ilişkisi içermektedir

Özellikler:

- parti : Parti
- partili_Ad : String
- partili_Soyad : String
- partili_Yas : int
- partili_Aciklama : String

Metodlar:

- PartiEkle(Parti):String
- PartiGuncelle(int,Parti):String
- PartiliGoster():String
- PartiliEkle(String,String,int,String):String
- PartiliAdGuncelle(id,String):String
- PartiliSoyadGuncelle(id,String):String
- PartiliYasGuncelle(id,int):int
- PartiliAciklamaGuncelle(id,String):String
- PartiliSil(int):String
- PartiliGoster():String

IV) **MevcutOylar**

Bu sınıf sayesinde sisteme Oy girme gibi temel işlemleri yönetebilmektedir.

Bu sınıf parti sınıfından kalıtım almıştır.

Özellikler:

- mevcutOySayisi:long

Metodlar:

- OyEkle():long
- OySil(): String
- MevcutOylar():long

V) **İtiraz**

Bu sınıf sayesinde sisteme itiraz etme işlemleri yönetebilmektedir.

Bu sınıf parti sınıfı ile * - 1 ilişki vardır (itiraz – parti).

Özellikler:

- parti : Parti
- itirazNo : int
- itiraz(Yapıcı değil) : String

Metodlar:

- İtirazEkle(int,String): String
- İtirazGuncelle(int,String): String
- İtiraz_Tum_goruntule ():String

VI) MilletvekiliAday

Bu sınıf sayesinde sisteme Kayseri Ekleme silme Güncelleme işlemlerini bulunduruyor.

Bu sınıf parti ile * - 1 (MilletvekiliAday –Parti) ilişkisi içermektedir.

Bu Sınıfı KayseriMvSecimBilgisinin parent'ıdır.

Özellikler:

- parti : Parti
- Milletvekili_Ad : String
- Milletvekili _Soyad : String
- Milletvekili _Yas : int
- Milletvekili _Aciklama : String
- Milletvekili _Sira:int

Metodlar:

- PartiEkle(Parti):String
- PartiGuncelle(int,Parti):String
- PartiliGoster():String
- Milletvekili Ekle(String,String,int,String,int):String
- MilletvekiliAdGuncelle(id,String):String
- Milletvekili SoyadGuncelle(id,String):String
- MilletvekiliYasGuncelle(id,int):int
- MilletvekiliAciklamaGuncelle(id,String):String
- MilletvekiliSiraGuncelle(id,String):String
- MilletvekiliSil(int):String
- MilletvekiliGoster():String

VII) KayseriSecimBilgi

Bu Sınıf Kayseri Milletvekili seçimi hakkında bilgi vermektedir.

Bu Sınıf Milletvekili Sınıfından kalıtım almıştır child'dır.

Bu Sınıf Sehir Sınıfının Parent'ıdır.

Özellikler:

- sehir : Sehir
- milletvekilKontenjanı:short

Metodlar:

- MilletvekiliKontenjanYaz(short):short
- MilletvekiliKontenjanGuncelle(short):short
- BilgiVer():String

VIII) Sehir

Bu Sınıf Belirtilen Sehir verisinin bulunduğu sınıftır.

KayseriSecimBilgi sınıfından kalıtım almıştır.

Özellikler:

- sehir : String == "Kayseri"

Use Case Diyagramı



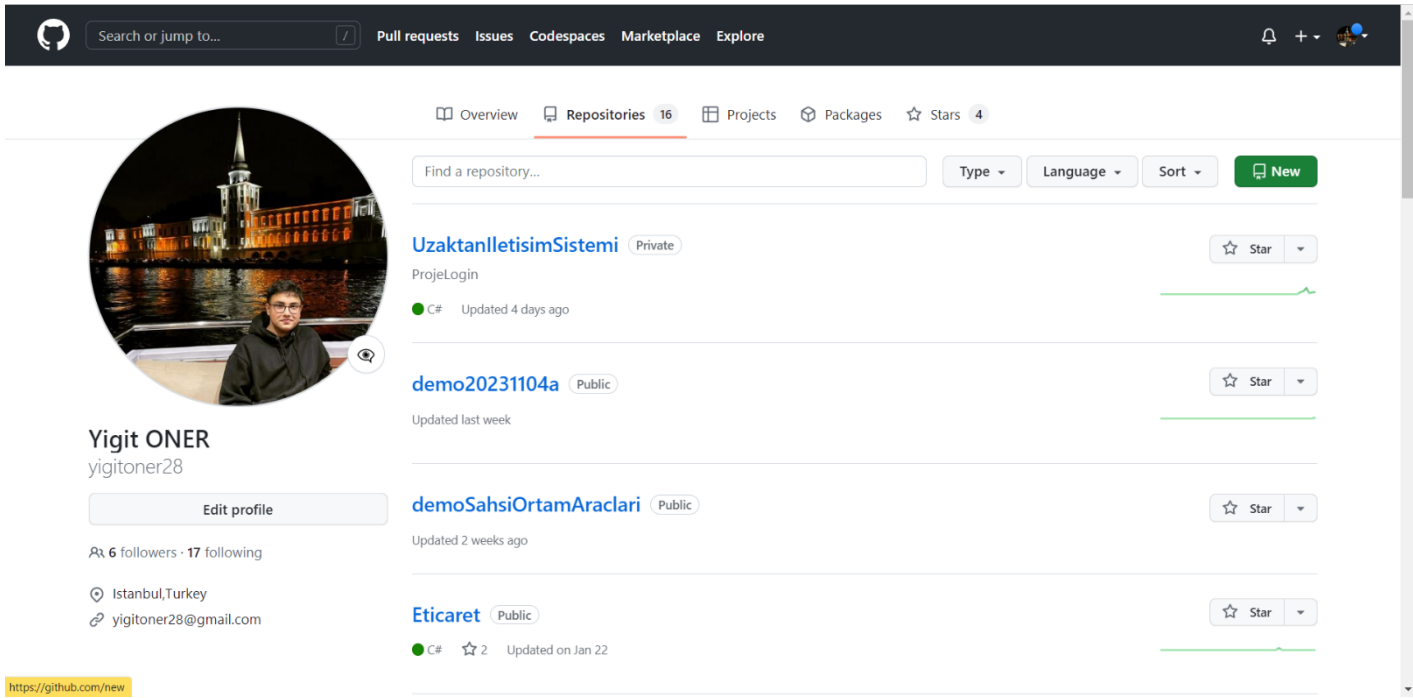
Projeyi Git İle Uzak Sunucuya Ekleme İşlem Evreleri

Bu başlık altındaki evreler Proje raporunun nasıl git vesilesi ile Uzak sunucuya yani Github'a nasıl iletildiği hakkında bilgi vermektedir.

Uzak Sunucuda Repository Açma

Bu evre Projenin yüklenecek olduğu uzak sunucudaki depoyu açma işlemini gösterir.

New tuşuna basarak yeni bir depo oluşturunuz.



The screenshot shows the GitHub profile of Yigit ONER (yigitoner28). The profile includes a circular profile picture of a man in a dark jacket, a bio stating "Istanbul, Turkey" and "yigitoner28@gmail.com", and a link to "https://github.com/new". The repository list shows four repositories: "UzaktaniletisimSistemi" (Private), "demo20231104a" (Public), "demoSahsiOrtamAraclari" (Public), and "Eticaret" (Public). The "demo20231104a" repository is highlighted with a green bar.

Repository Name	Visibility	Language	Updated	Stars
UzaktaniletisimSistemi	Private	C#	Updated 4 days ago	0
demo20231104a	Public	C#	Updated last week	0
demoSahsiOrtamAraclari	Public	C#	Updated 2 weeks ago	0
Eticaret	Public	C#	Updated on Jan 22	2

Search or jump to... Pull requests Issues Codespaces Marketplace Explore

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner **yigitoner28** / Repository name **2022-2023H5210016Yevo** ✓

Great repository names are short and memorable. Need inspiration? How about [automatic-enigma](#)?

Description (optional)

☐ Public
Anyone on the internet can see this repository. You choose who can commit.

☒ Private
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: **None**

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

License: **None**

ⓘ You are creating a private repository in your personal account.

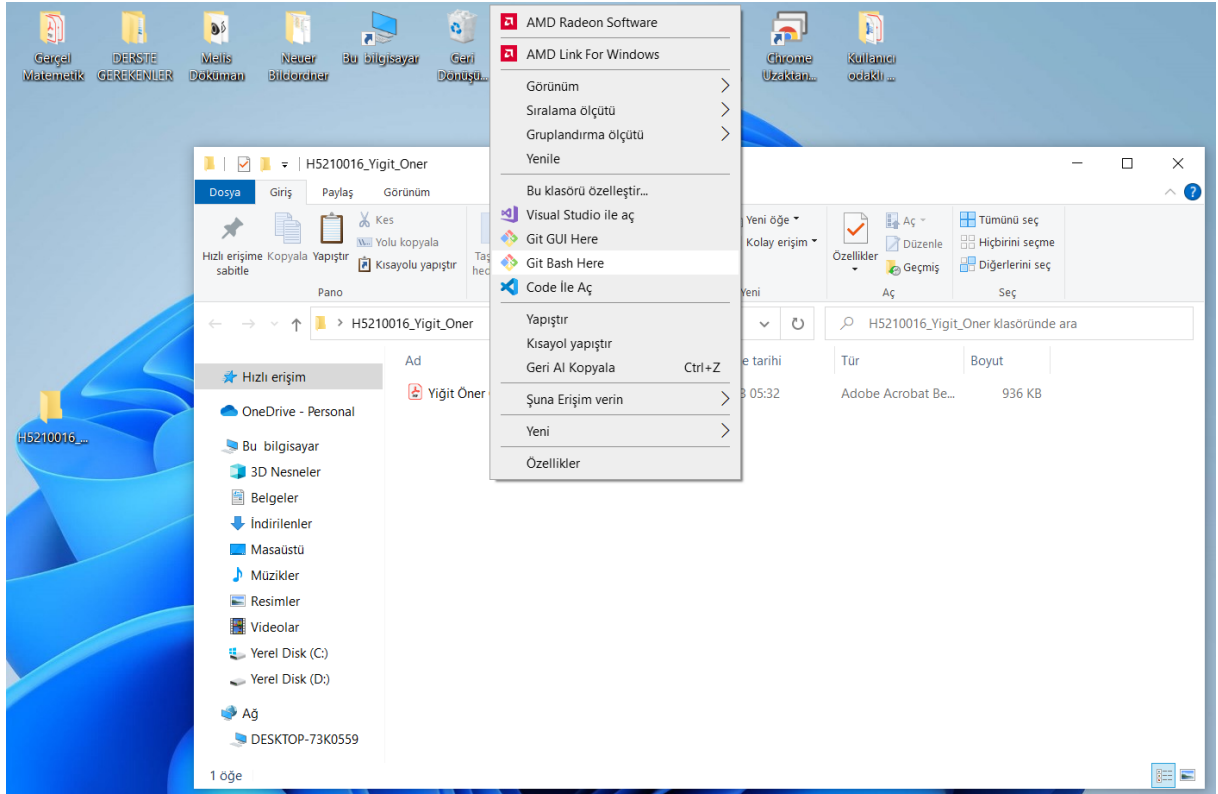
Create repository

New tuşu ile Create a new repository ekranına geçtikten sonra;

- Repository Name kısmını dolduruyoruz
- Erişim kısmını Private olarak seçiyoruz
- Son olarak Create repository tuşuna basıyoruz

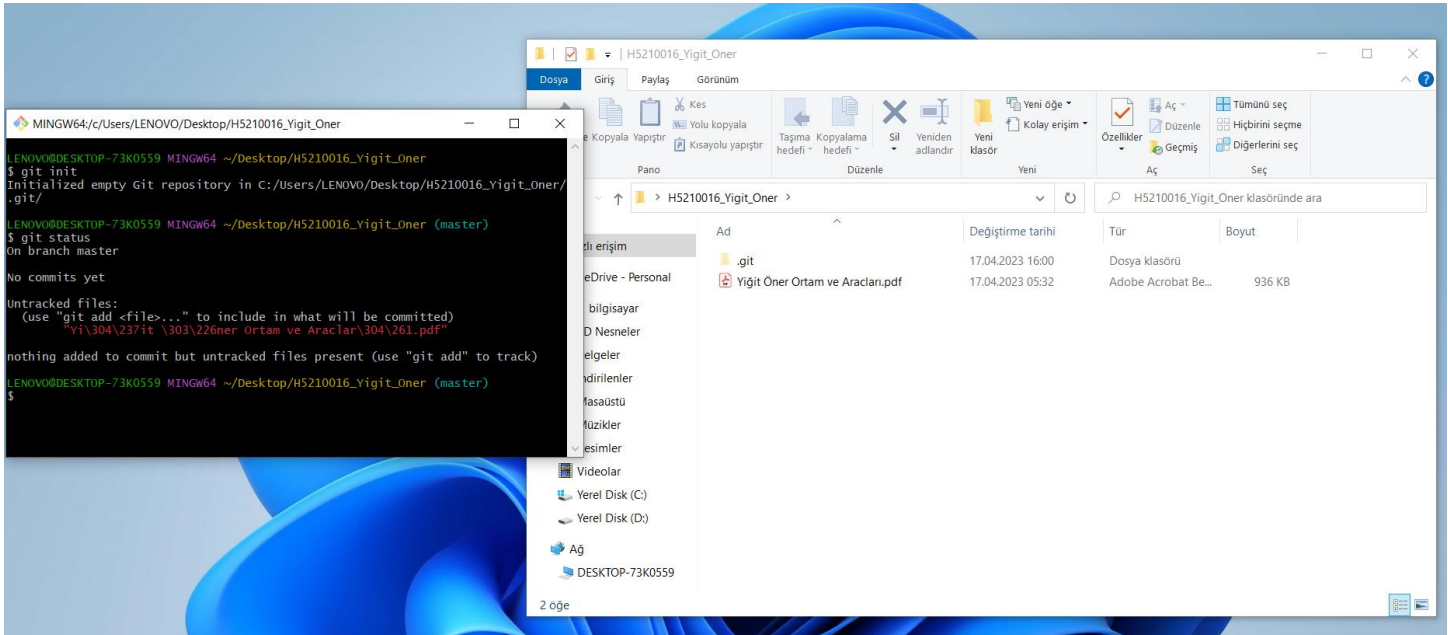
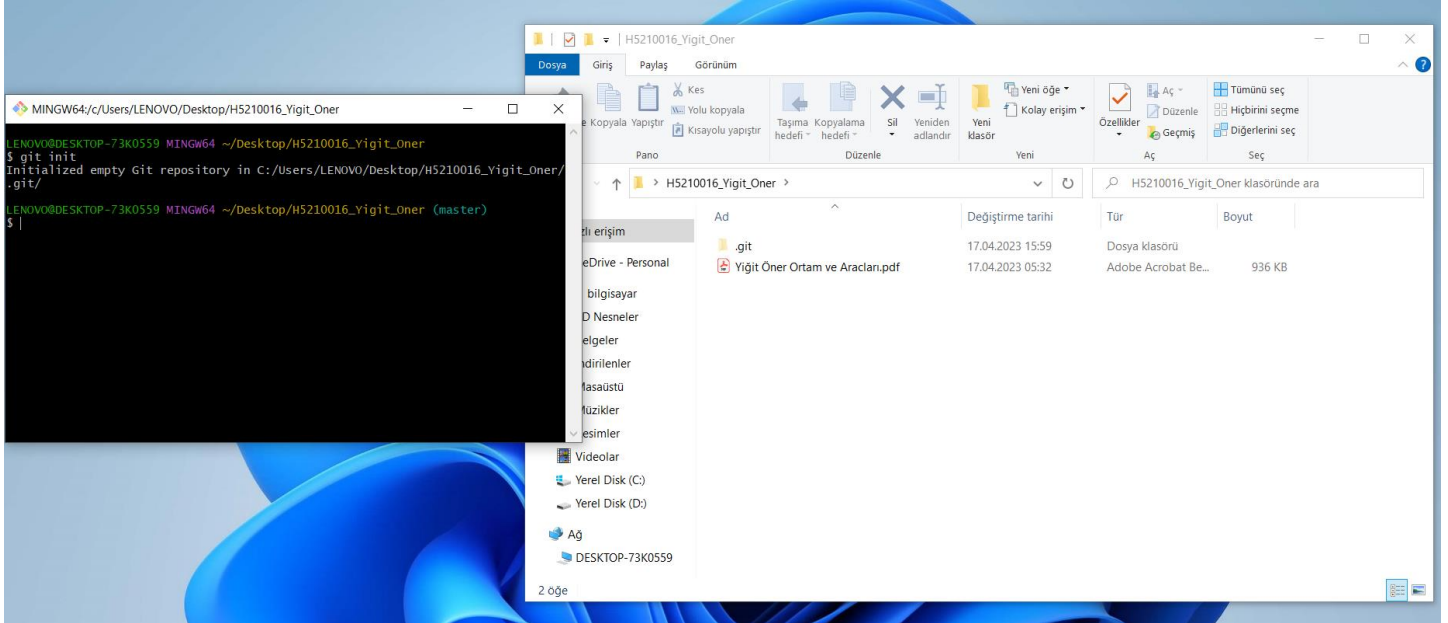
Git Bash Kısmı

Bu evrede git komutlarını o dosyada yapmak için gerek ilk evreyi yani git terminalini açıyoruz



Git Init

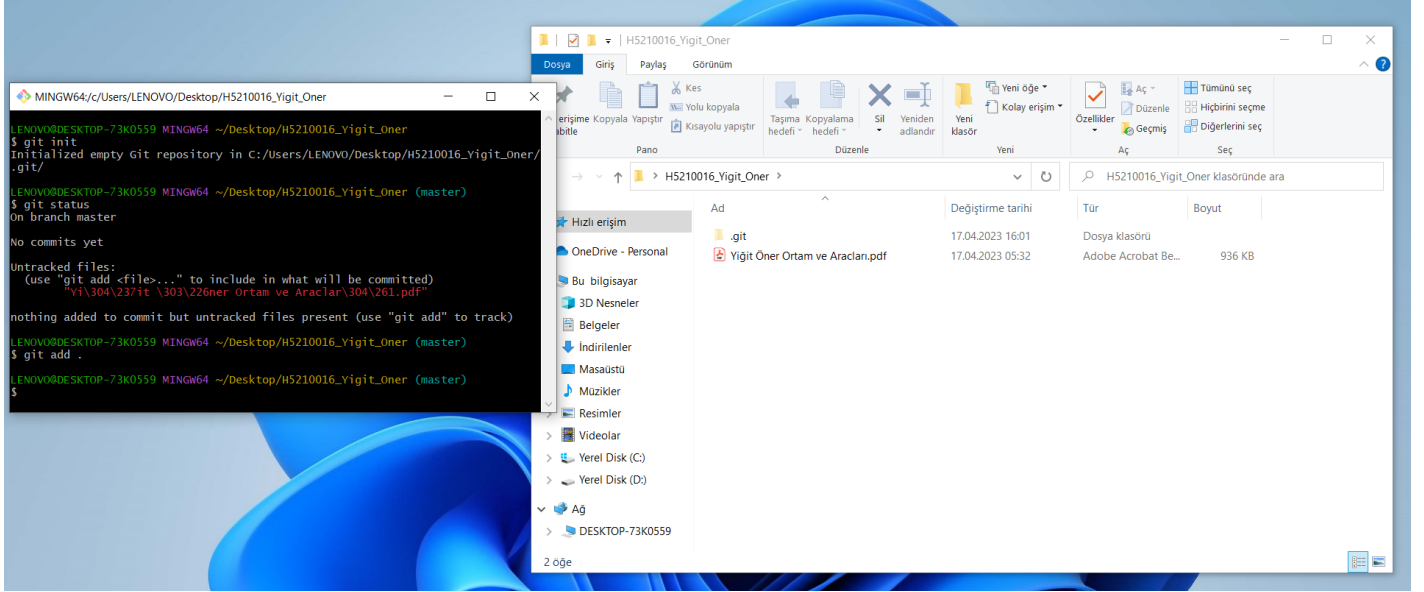
Bu Evre Git Bash işlemini yani terminali açtıktan sonra dosya içindeki elemanları git komutu ile yönetmemizi sağlayan ilk komut işlemidir.



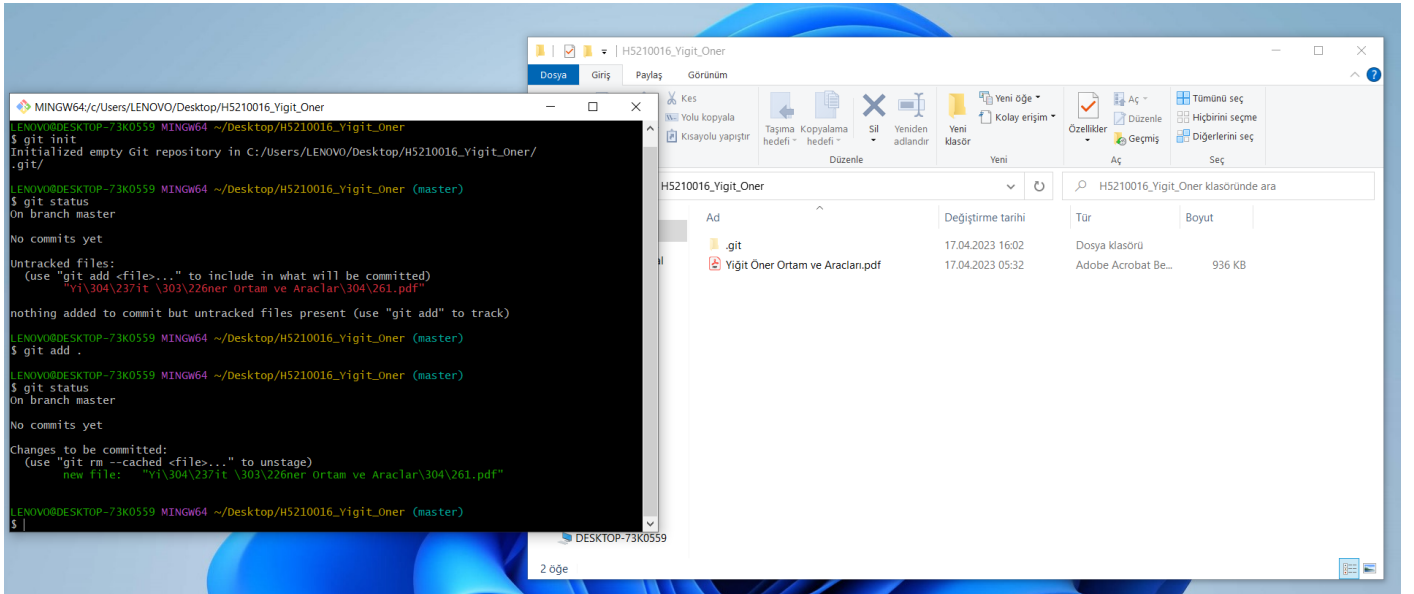
Daha Sonrasında işlem commit edilmiş mi veya sorun var mı diye git status komutunu giriyoruz. Bu kod bize dosyanın daha takip altında olmadığını belirtiyor.

Git Add

Bu Evre işlemlerin lokal repository'e yollamadan önceki son işlemi ifade ediyor



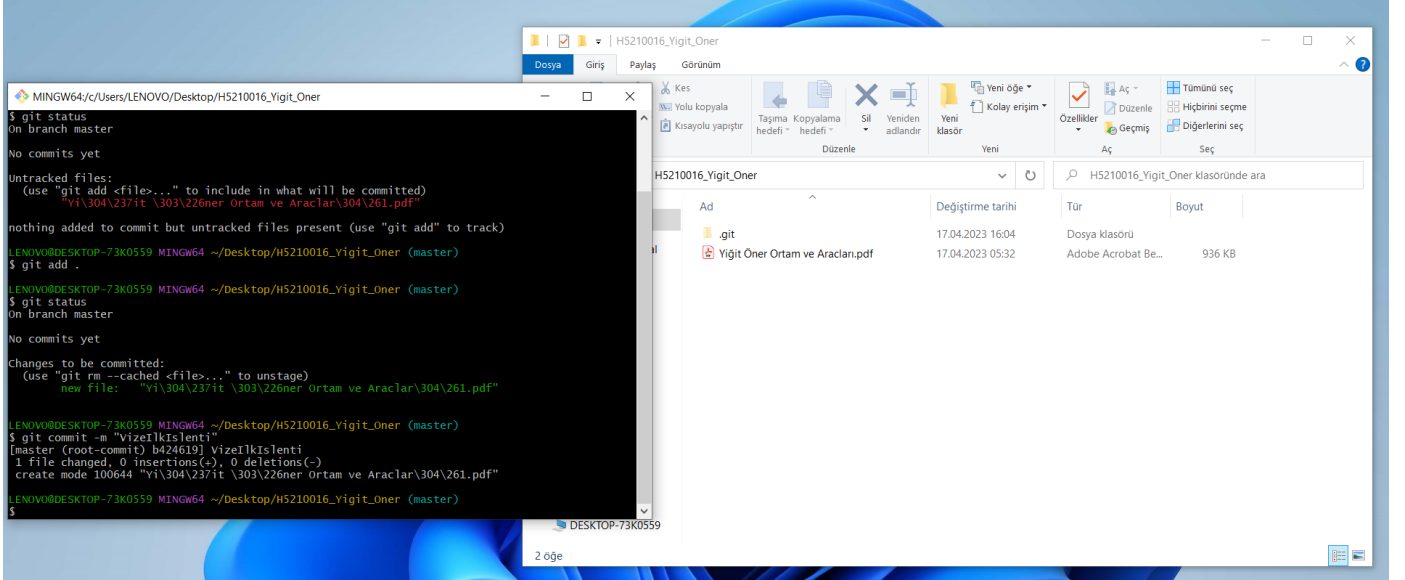
Git add Commit işleminden önceki son işlemdir **add** . kullanılmasının nedeni klasör içerisindeki tüm dosyaları ekleme işlemini yapmayı sağlamaktadır.



Başarılı şekilde ekleme işlemi gerçekleştirdik :Bu işlem Git status sayesinde oldu.

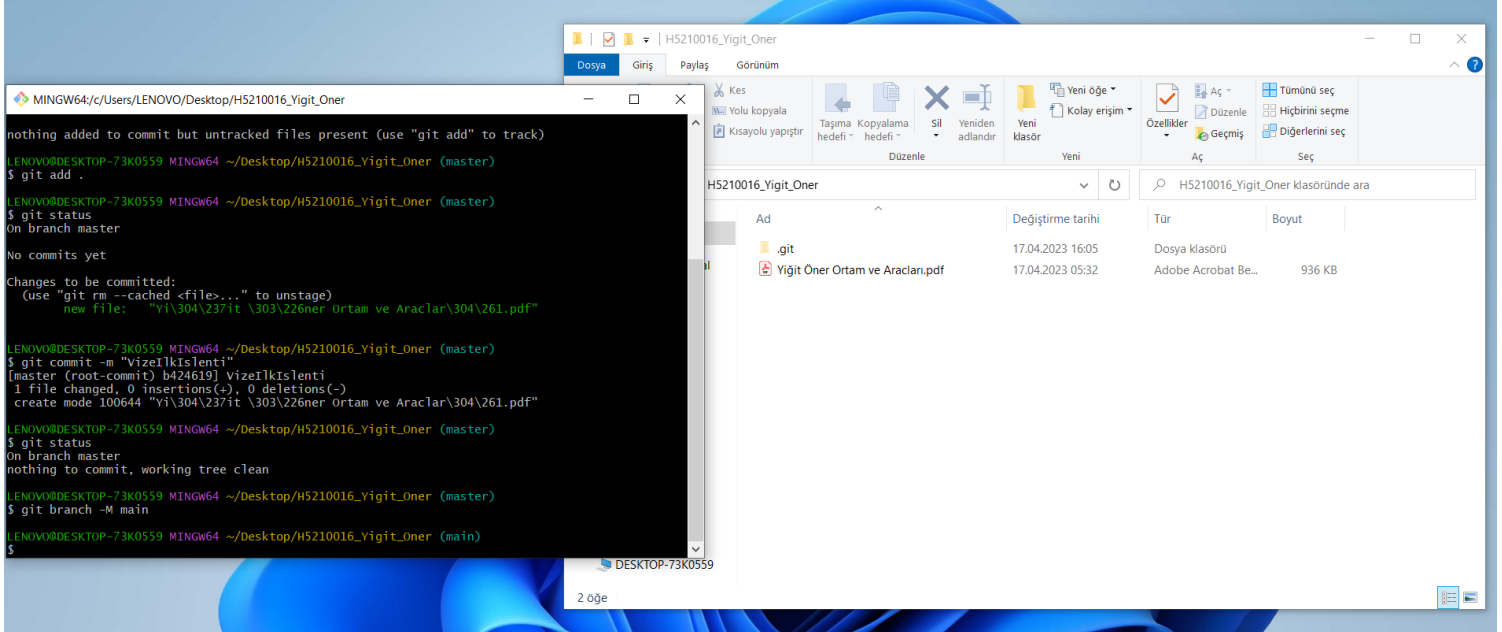
Git Commit

Bu evre lokal depoya ekleme işlemini gerçekleştirmeyi sağlamaktadır . (staging areaya aktarma işlemi).



Git Branch

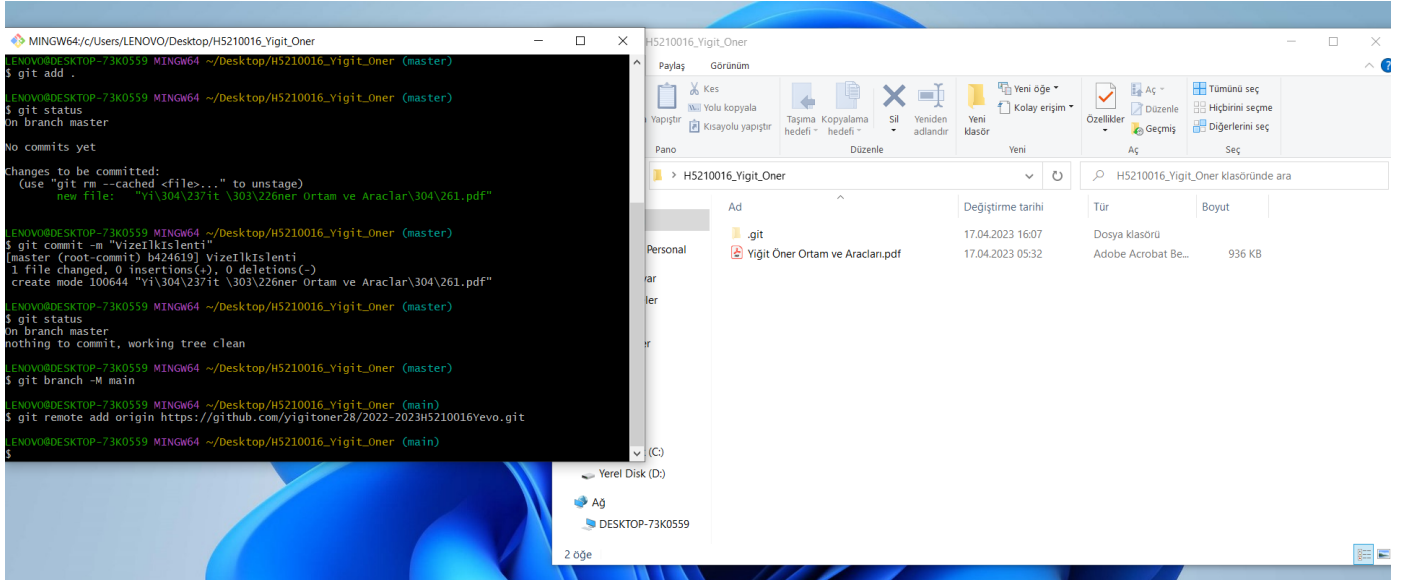
Bu evre Lokaldeki dosya(ları) hangi dalda olacağını belirttiğimiz yerdır.



Git branch -m main (main) dalına ekleme işlemini gerçekleştirdikten sonra dosya lokal dalda kaydedildi .

Git Remote

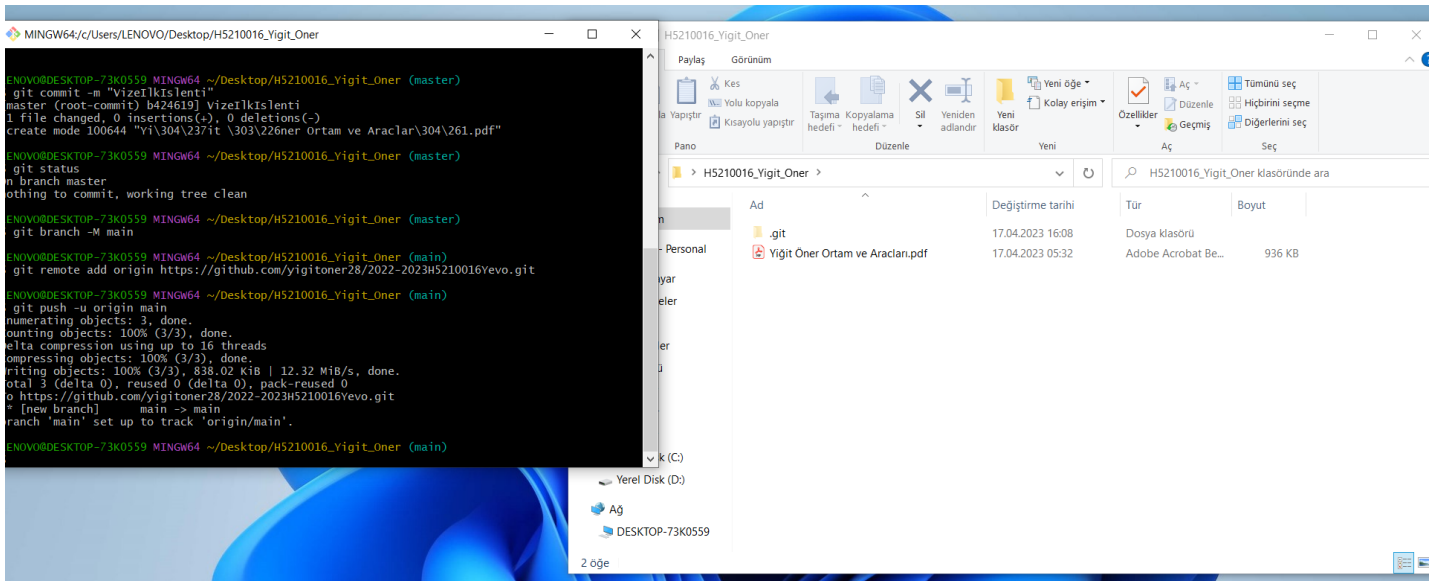
Bu evre Lokaldeki eklenmiş dosyaları uzaktaki hangi sunucuya ekleneceğini belirttiğimiz evredir.



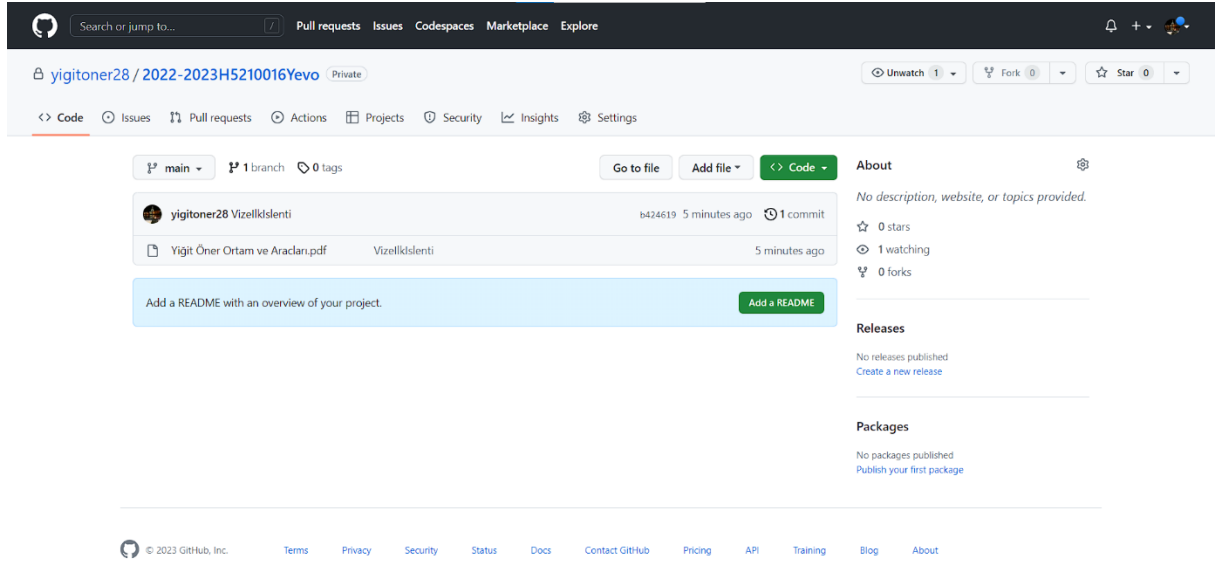
Git remote add origin (Uzaktaki sunucu https'i dir.)

Git Push

Bu işlem hangi uzak sunucu kısmını gerçekleştirdikten sonra lokaldeki commitleri uzak sunucuya yollama işlemini tasvir eder.



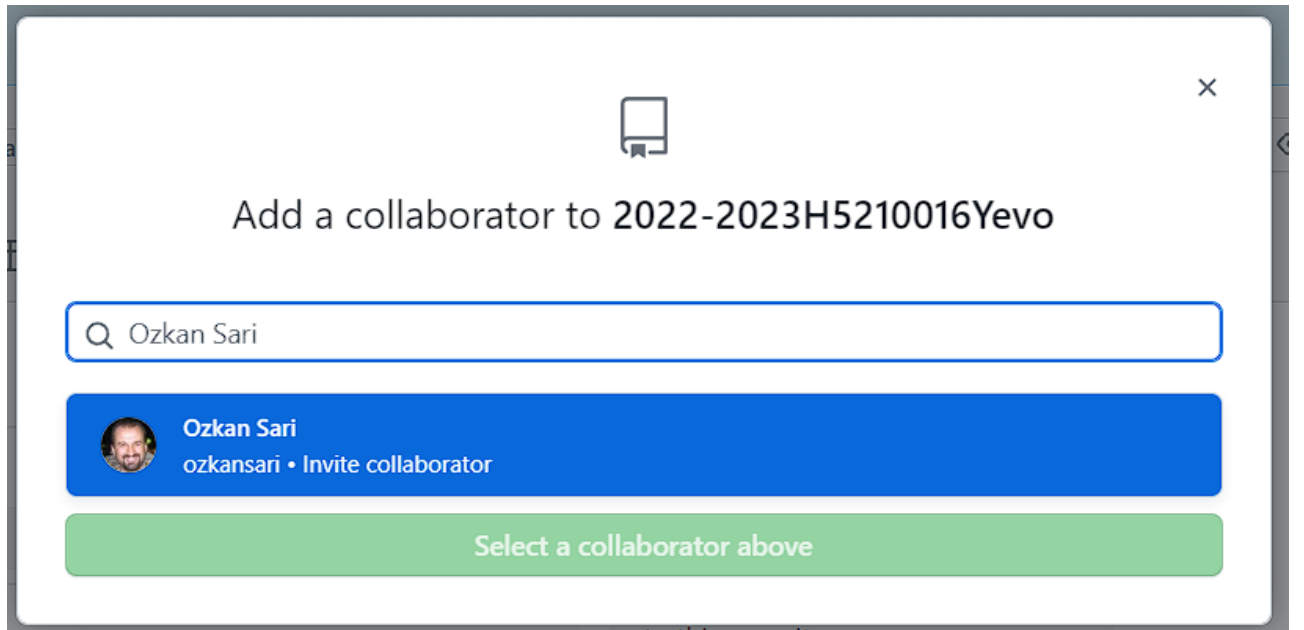
İşlemler Gerçekleştikten Sonraki Git Hub Bölümleri



Yaptığımız init > add > commit > branch > remote > push işlemleri gerçekleştikten sonra github kısmına gelen dosyalardır.

Collaborators Ekleme Kısmı

Bu Evre projeye katkı sağlamasını istediğimiz kullanıcıları ekleme işleminin bulunduğu kısımdır.



ozkansari has been added as a collaborator on the repository.

yigitoner28 / 2022-2023H5210016Yevo Private

Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights Settings

General

Access

Collaborators

Code and automation

Branches

Tags

Actions

Webhooks

Codespaces

Pages

Security

Code security and analysis

Deploy keys

Secrets and variables

Integrations

GitHub Apps

Email notifications

Who has access

PRIVATE REPOSITORY

Only those with access to this repository can view it.

[Manage](#)

DIRECT ACCESS


1 has access to this repository, 0 collaborators, 1 invitation.

Manage access

[Add people](#)

☐ Select all Type

Find a collaborator...

<input type="checkbox"/>	 Ozkan Sari Awaiting ozkansari's response	Pending invite	Remove
--------------------------	--	----------------	------------------------

[< Previous](#) [Next >](#)

Tüm Komutlar

```
MINGW64~/c/Users/LENOVO/Desktop/H5210016_Yigit_Oner
LENOVO@DESKTOP-73K0559 MINGW64 ~/Desktop/H5210016_Yigit_Oner
$ git init
Initialized empty Git repository in c:/Users/LENOVO/Desktop/H5210016_Yigit_Oner/.git/
LENOVO@DESKTOP-73K0559 MINGW64 ~/Desktop/H5210016_Yigit_Oner (master)
$ git status
On branch master
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        "Yi\304\237it \303\226ner ortam ve Araclar\304\261.pdf"
nothing added to commit but untracked files present (use "git add" to track)
LENOVO@DESKTOP-73K0559 MINGW64 ~/Desktop/H5210016_Yigit_Oner (master)
$ git add .
LENOVO@DESKTOP-73K0559 MINGW64 ~/Desktop/H5210016_Yigit_Oner (master)
$ git status
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   "Yi\304\237it \303\226ner ortam ve Araclar\304\261.pdf"
LENOVO@DESKTOP-73K0559 MINGW64 ~/Desktop/H5210016_Yigit_Oner (master)
$ git commit -m "VizeIlkIsilenti"
[master (root-commit) b424619] VizeIlkIsilenti
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "Yi\304\237it \303\226ner ortam ve Araclar\304\261.pdf"
LENOVO@DESKTOP-73K0559 MINGW64 ~/Desktop/H5210016_Yigit_Oner (master)
$ git status
On branch master
nothing to commit, working tree clean
LENOVO@DESKTOP-73K0559 MINGW64 ~/Desktop/H5210016_Yigit_Oner (master)
$ git branch -M main
LENOVO@DESKTOP-73K0559 MINGW64 ~/Desktop/H5210016_Yigit_Oner (main)
$ git remote add origin https://github.com/yigitoner28/2022-2023H5210016Yevo.git
LENOVO@DESKTOP-73K0559 MINGW64 ~/Desktop/H5210016_Yigit_Oner (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (0/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 838.02 KiB | 12.32 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
to https://github.com/yigitoner28/2022-2023H5210016Yevo.git
* [new branch]    main -> main
branch 'main' set up to track 'origin/main'.
LENOVO@DESKTOP-73K0559 MINGW64 ~/Desktop/H5210016_Yigit_Oner (main)
$
```