**2.Hafta Konuları**

1. Repository nedir, ne işe yarar?
2. Kendi profilimde ilk repositorymi nasıl oluştururum?
3. Commit nedir, push nedir?
4. Kendi kodumu Githuba nasıl eklerim ve nasıl çalışırım?
5. UYGULAMA: Kişisel Github profili üzerinde repository oluşturma ve commit.

**2. Hafta Giriş**

Arkadaşlar merhabalar. Bundan sonraki örneklerde kavram yanılgılarını en aza indirgemek için Git yerine Github kullanılacaktır. Kavram yanılgılarından korkulmamalıdır. Pratik yaptıkça kavramlar oturacaktur. İkinci hafta derslerimizi anlattıktan tüm kavramları anlatacağımız bir video çekeceğiz ve Github öğrenmek isteyen herkesin Github üzerinde uygulama yaptığından emin olacağız.

**Github Desktop**

Github Desktop uygulaması ile tüm git işlemleri kolay bir şekilde yapılabilir ve anlaşılabilir olmaktadır. Pratik olarak Github Desktop uygulaması ile Git kullanımını anladıktan sonra teorik kısmını anlaşılması oldukça kolaylaşacaktır.

Github Desktop uygulamasını kurmak için aşağıdaki linke tıklayabilirsiniz.

<https://desktop.github.com/>

Bu haftaki 3 temel konu Github Desktop uygulaması üzerinden anlatılacaktır:   
1.Repository

2.Commit

3.Push

**Repository nedir, ne işe yarar?**

Repository(repo) ya da türkçesi ile depo, geliştirme yapılan proje klasörünü izlenebilir kılar. Proje klasörü içerisindeki tüm dosyalarda, klasörlerde yapılan değişiklikleri izlenebilir kılar.

* “OzgecmisRepo” adında bir proje klasör ve içerisinde Ali.txt adındaki dosya olsun

Resimde de görüldüğü üzere Local repo ve Remote repo adında iki farklı alan vardır. Local kavramı geliştiricinin bilgisayarı olarak düşünülebilir. Github Desktop uygulaması kurulan bilgisayar local alandır ve yapılan değişiklikler (OzgecmisRepo klasörü) izlenebilir. OzgecmisRepo klasörünün repository olabilmesi için Git tarafından takip ediliyor olabilmesi gerekmektedir.

OzgecmisRepo adında izlenebilir bir klasör yani repository oluşturalım.

1. Github desktop uygulaması açılır.

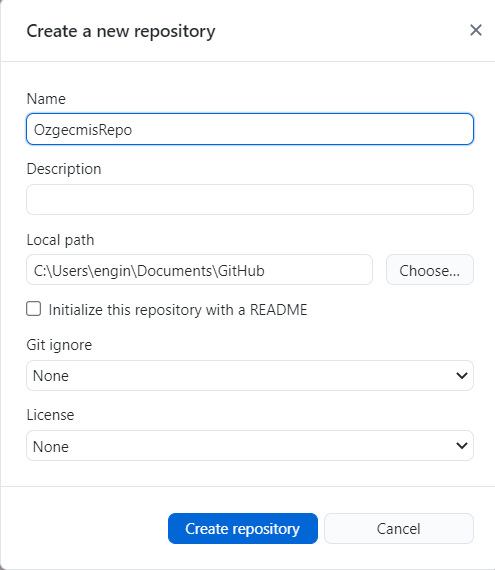


1. File>New Repository yolunu takip edilir.

A screenshot of a computer

Description automatically generated

1. Proje adına(Name alanı) OzgecmisRepo yazıldıktan sonra Create Repository(Repository oluştur) butonuna basılır. Create Repository butona bastıktan sonra bir repository oluşturur.



1. Bu işlem sonucunda şekildeki ekran gözükür.Graphical user interface, text, application, website

   Description automatically generated

Not: Bu ekran, geliştirme aşamasında en sık görülecek ekrandır. Remote repository`e kod göndermek ve Repote repository`de ki son değişiklikleri alınacak olan sayfadır. Anasayfa niteliğindedir.

Repository neydi? Git olayları takip edilen bir klasör. Oluşturduğu tek şey aslında budur. Bu klasörü görebilmek için resimde işaretlenen KISIM 2`nin altında “Show in Explorer” butonuna tıklanır veya kısayolu olan ctrl + shift + f yapılır. Butona tıklandıktan sonra şekildeki gibi bir klasör oluşturmuştur Create Repository butonu.

Graphical user interface, text, application, email

Description automatically generated

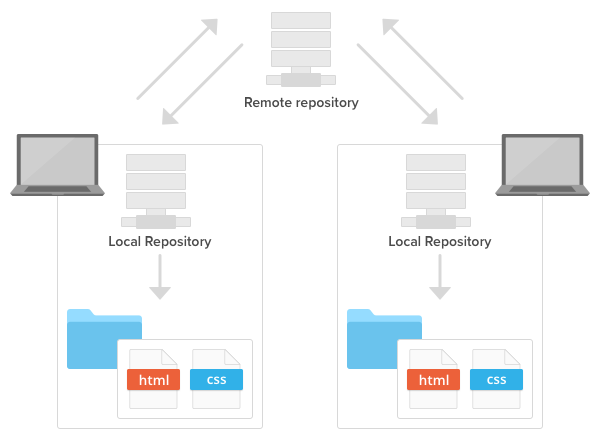
Ctrl + Shift + F yapıldığında, resimde görüldüğü üzere OzgecmisRepo adındaki klasör Dökümanlar>Github>OzgecmisRepo içerisinde oluşturulmuştur. İçerisinde otomatik olarak oluşturulan bir klasör ve bir dosya mevcuttur. 1. Haftada anlatılan Ali.txt dosyası örneğinde bahsedilen 4 satır komutun ilki olan **git init** komutu yazıldığında **.git klasörü** oluşuyor. Create Repository butonuna tıklanıldığında bu komut arka planda otomatik olarak yürütülür. Yani ilk haftada anlatılan **git init** komutu, Github Desktop uygulaması vasıtasıyla otomatik oluşturulur ve **.git** klasörü şekildeki gibi görülür **git init** yazmaya gerek kalmadan tıklama yaparak hızlı bir şekilde oluşturulur. Bu klasör Git`in çalışması için gerekli her şeyi tutar.

Not: Gitattributes dosyası daha sonra anlatılacaktır.

Text

Description automatically generated

Create Repository butonuna tıklandığında, bir local repository oluşturulur. Bu aşamada repository sadece local bilgisayarda oluşturulmuş olunur.



Remote Repository üzerine kendi yaptığımız değişiklikleri göndermek, orada bir repo oluşturmak ve local repodaki değişiklikleri oraya göndermek çok kolaydır. Kısım 1 olarak gösterilen yerde publish repository butonuna tıklandığında, Github desktop uygulaması, Github uygulamanıza bağlı olduğundan, remote repository üzerinde bir proje oluşturur.

Text

Description automatically generated

Publish repository butonuna tıklayalım.

Resimdeki ekranda repository üzerinde çeşitli ayarlamalar yapılır ve Publish Repository butonuna basılır.

Graphical user interface, text, application

Description automatically generated

Bu işlemi yaptıktan sonra resimdeki ekran gözükür

Graphical user interface, text, website

Description automatically generated

Bu aşamadan sonra KISIM 1 olarak işaretlenilen alanda “Publish Repository” butonu yerine “Open the repository in your external editor” yazısı ile gelen “Open in Visual Studio Code” butonu belirir. Artık Remote ve Local repository`nin senkron(eşlenmiş repositoryler) olduğunu gösterir. Bu butona tıklanması durumunda, Visual Studio Code programı çalışacaktır. Eğer VS Code uygulamasına sahip değilseniz bu linkten indirebilirsiniz: <https://code.visualstudio.com/download>

“Open in Visual Studio Code” butonuna tıklandığında, VS Code programı içerisinde, Repository klasörü açılır. Açılan bu VS Code ekranında, Windows işletim sisteminden görüntülediğimiz repository klasörünü aynen görürüz

A screenshot of a computer

Description automatically generated with medium confidence

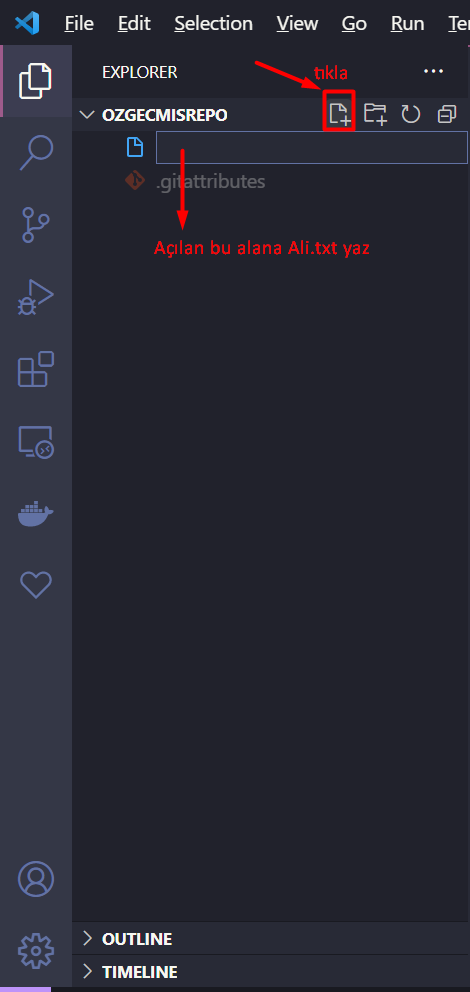
VS Code nedir?

Windows pencerelerinden(Windows yerine Linux`ta olabilir. Genel kullanım Windows olduğu varsayıyoruz) yapılabilen her türlü dosya veya klasör oluşturma, düzenleme işlemlerini daha hızlı bir şekilde yapılmasını sağlayan, çeşitli eklentiler ile kod geliştirmek için hazırlanmış, yazılım geliştirme aşamasındaki problemleri en aza indiren bir düzenleme aracıdır.

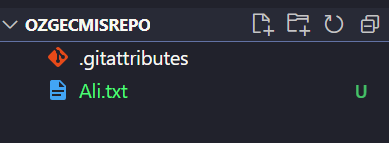
Uygulama:

local repository üzerinde Ali.txt dosyası oluşturup remote repository`e gönderelim.

Sol taraftaki panele dosya oluşturma simgesine tıkladıktan sonra, gösterilen alana Ali.txt yazalım.

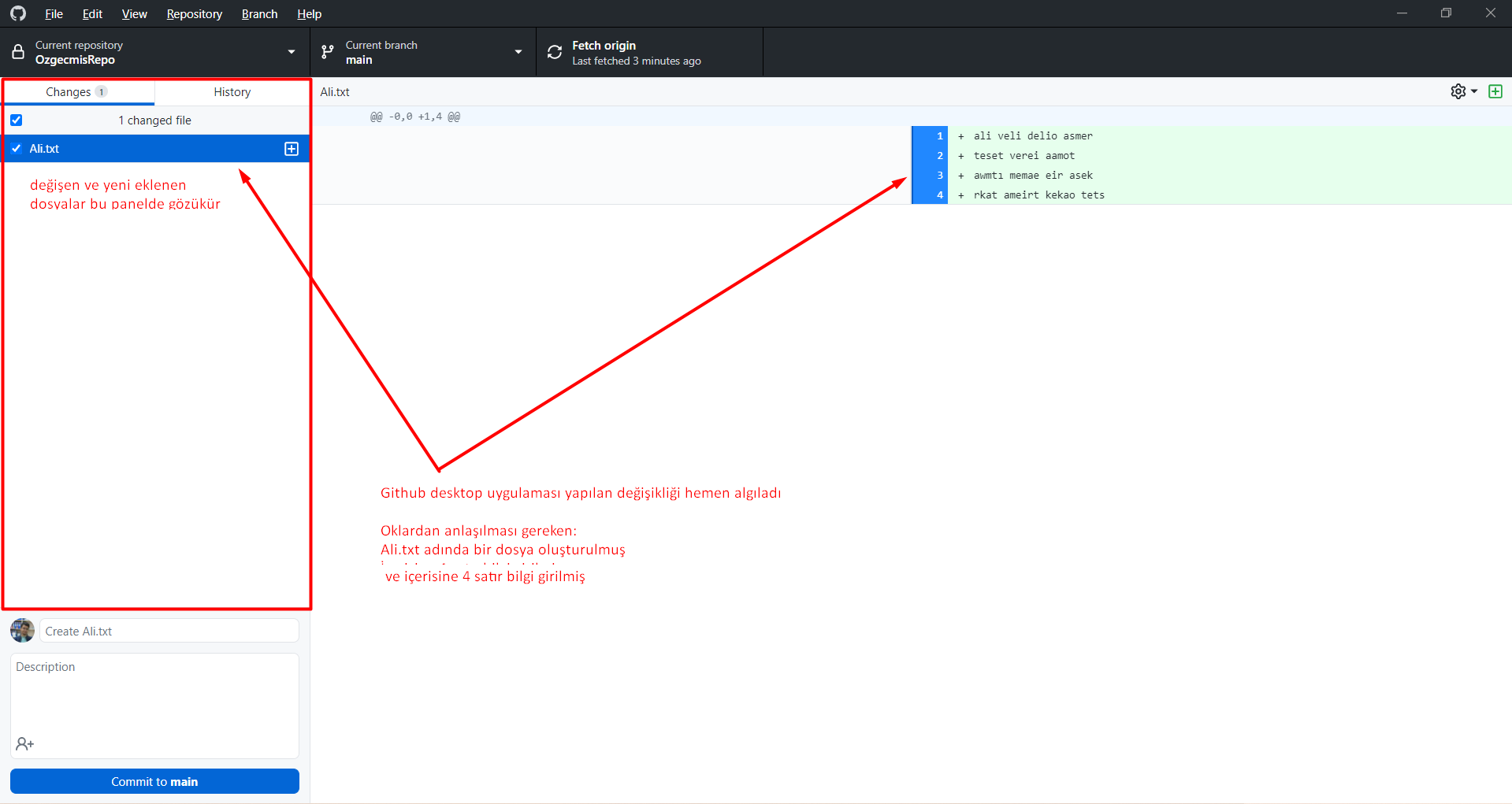


Dosyayı oluşturduktan sonra içerisine 5 satırlık bir metin girelim ve ctrl + s yaparak kaydedelim. Veya File>Save yapalım.



Graphical user interface, text

Description automatically generated

Bu aşamadan sonra Github Desktop uygulamasını açalım. Resimde görüldüğü üzere Github desktop uygulaması değişiklikleri algıladı.

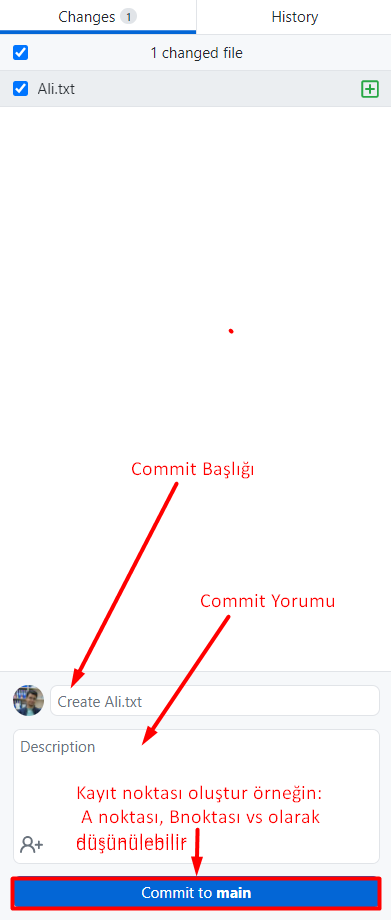
Resimde yapılan değişiklikleri Remote Repository`e atmak için yani Ali.txt dosyasının Remote repository(Github) üzerinde görünmesini sağlamak için yapılan değişiklikleri **Commit** ve ardından Push etmemiz gerekmektedir. Bu iki kavramın öğrenilmesi elzemdir.

Commit Nedir? Commit o anki yapılan değişikliklerin baydını tutar. Örnekte yer alan 1 adet dosya değişikliği olan Ali.txt dosyası commit edildikten sonra bir kayıt noktası oluşturur. 1. Hafta anlatılan A ve B noktaları commit demektir. Yukarıdaki resimde yorum yazdıktan sonra “Commit to main” butonuna bastığımızda, yapılan her değişikliği bir kayıt noktası olarak kaydedecektir. Bu kayıt noktaları projenin her bir sayfa kodunun o anki zaman dilimindeki kaydını tutar. Yani geçmişteki bir zamandaki, tam o andaki tüm kodlara erişilebilinir.

Push nedir? Push o anki commit edilen her bir kayıt noktasını, local repository`den remote repository`e gönderim işlemini sağlar.

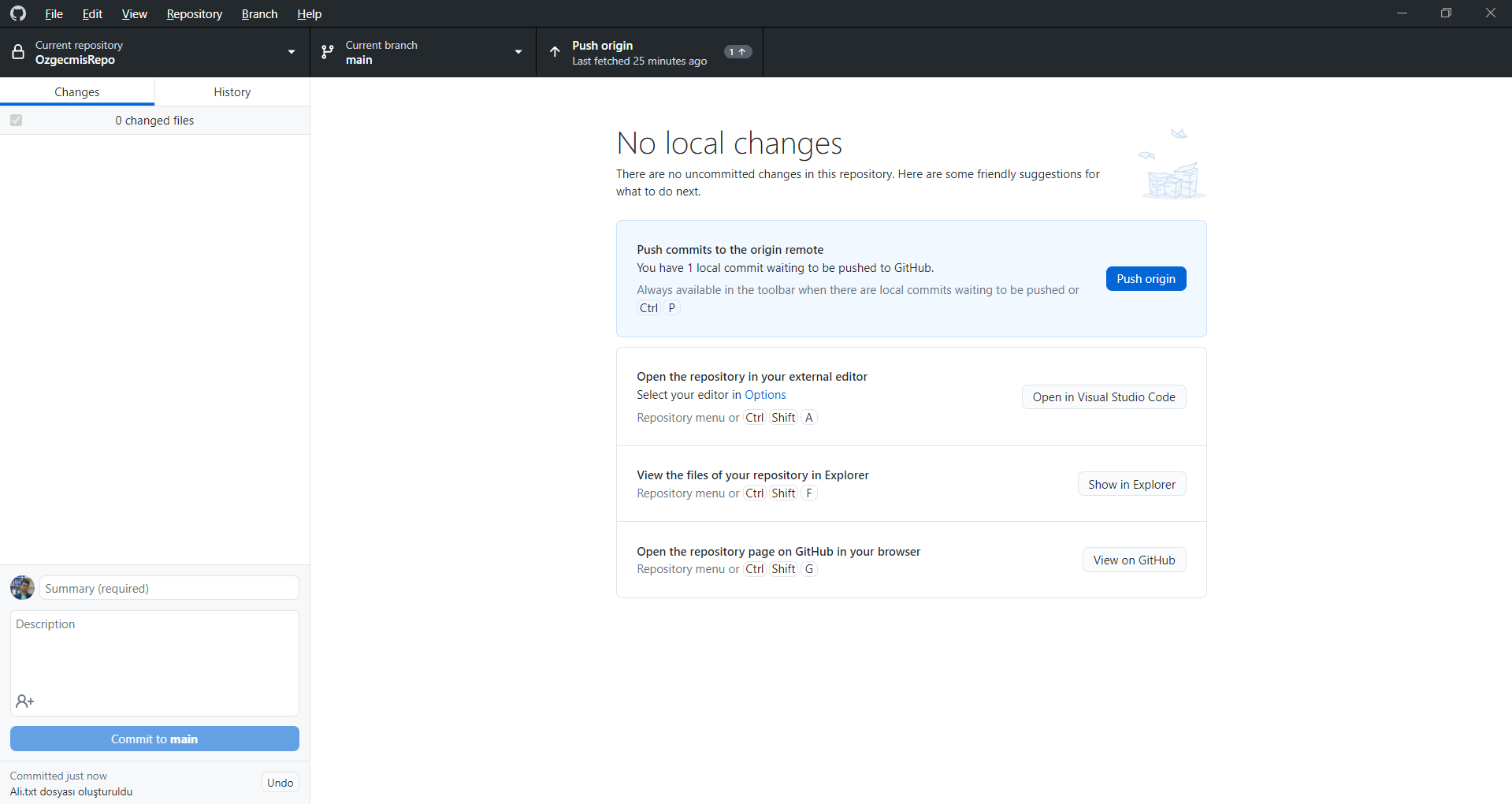
Uygulama - Commit ve Push:

Ali.txt dosyasını düzenleyip değişiklikerin Github tarafından görülmesini sağlamıştık. Tam bu anda yapılan değişiklikleri kayıt noktası olarak kaydetmek istenilidiğinde, bir commit mesajı yazdıktan sonra “Commit to Main” butonuna tıklandığında Kayıt noktası işlemi gerçekleşir.

 Graphical user interface, text, application, email

Description automatically generated

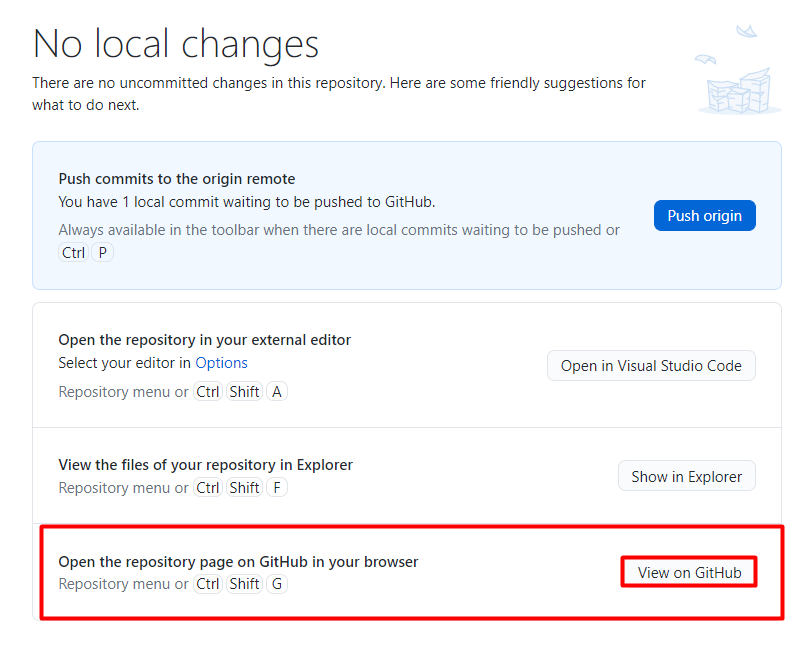
Bu aşamadan sonra resimde görüldüğü gibi, KISIM 1 bölgesinde, Push Origin butonu belirmiştir. Push Origin butonu, Remote repository üzerine değişiklikleri atacaktır. Butona tıklayalım.

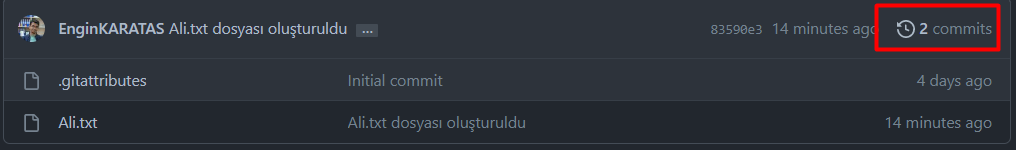


Graphical user interface, text

Description automatically generated

Yapılan değişiklikleri Github üzerinde görmek için Kısım 2 olarak işaretlenen bölgenin altındaki bölgede View on Github butonuna tıklayalım.



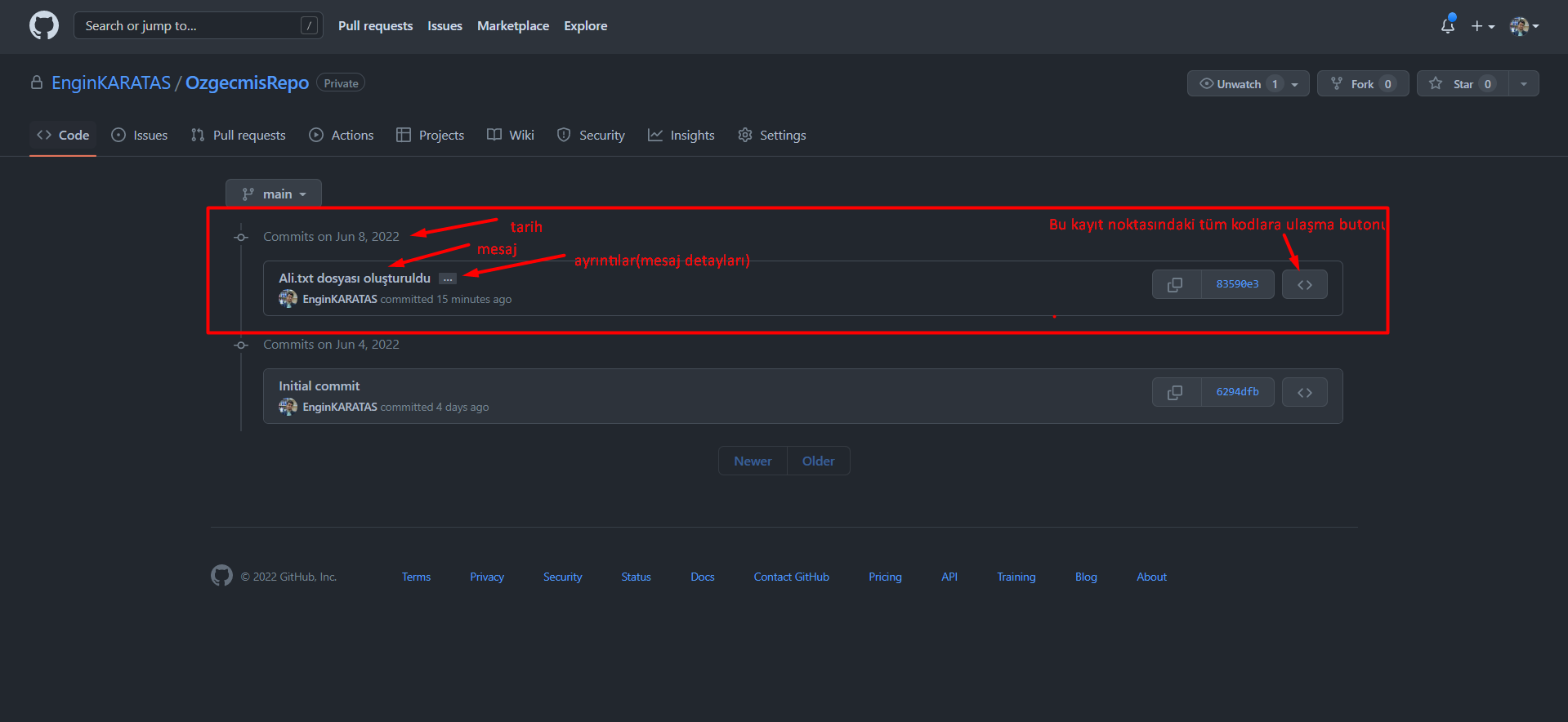
Github`da görüleceği üzere Ali.txt dosyası oluşmuştur. Commitlerimizi görmek için 2.resimde görülen butona tıklayalım

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Commits butonuna tıklanınca resimdeki gibi commits ekranı açılır. Bu noktada “Ali.txt dosyası oluşturuldu” notuyla yapılan commit işlemini görebilmekteyiz. 

Bu commit mesajının içeriği üzerine tıklandığında yapılan değişiklikler görülebilir.

A screenshot of a computer

Description automatically generated

Örnek:

* 1. Github Desktop uygulaması

Text

Description automatically generated with low confidence

**Yeni bir klasör oluşturalım**

A screenshot of a computer

Description automatically generated

Visual Studio`da HelloWorld adında bir proje oluşturalım ve lokasyonunu oluşturduğumuz klasör olarak seçelim.

Graphical user interface, application

Description automatically generated

Bu klasörü bir Repository yani depo olarak adlandırılabilir. Fakat Git şu anda yaptığımız değişiklikleri izlemiyor. **git init** ile Github-yobu-ce-hello-world klasörünü git için

İşletim sisteminizde (Windows, Mac, Linux) oluşturulan klasör içerisine bir proje oluşturulsun.