**\*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 oturacaktır. İkinci hafta derslerimizi anlattıktan sonra tüm kavramları anlatacağımız bir video çekeceğiz ve Github öğrenmek isteyen herkesin Github üzerinde uygulama yaptığından emin olacağız.

Yeni katılan arkadaşların 1. Hafta anlatımı okumaları tavsiye edilir. Bunun için link: ```

<https://github.com/YOBU-Computer-Engineering/github-lecture-notes/blob/main/week1.md>

\*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

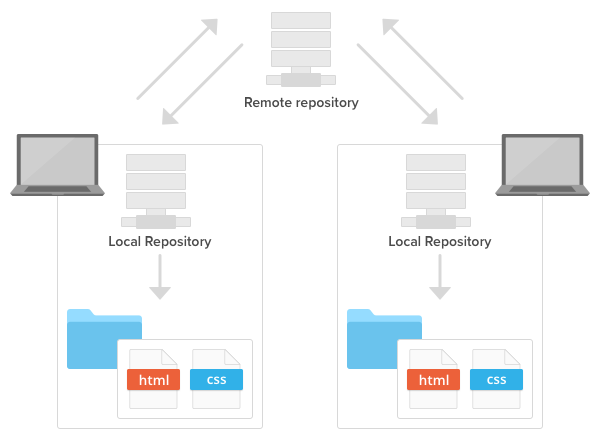
2.Commit

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.

Resimde de görüldüğü üzere Local repo ve Remote repo adında iki farklı alan vardır. Local kavramı geliştiricinin yani önünüzdeki bilgisayar olarak düşünülebilir(Local = Senin Bilgisayarın = geliştiricinin bilgisayarı). Github Desktop uygulaması kurulan bilgisayar local repository`dir. Yani geliştiricinin bilgisayarına kurulmuş Repository`dir.```



**\*Uygulama\***: OzgecmisRepo adında izlenebilir bir klasör yani repository oluşturalım.

```“OzgecmisRepo” adında bir proje klasörü ve içerisinde Ali.txt adındaki dosya oluşturalım.

OzgecmisRepo klasörünün repository olabilmesi için Git tarafından takip ediliyor olabilmesi gerekmektedir. Github Desktop uygulaması üzerinden OzgecmisRepo adında repository oluşturalım. ```

1. **```**Github desktop uygulaması açılır. **```**

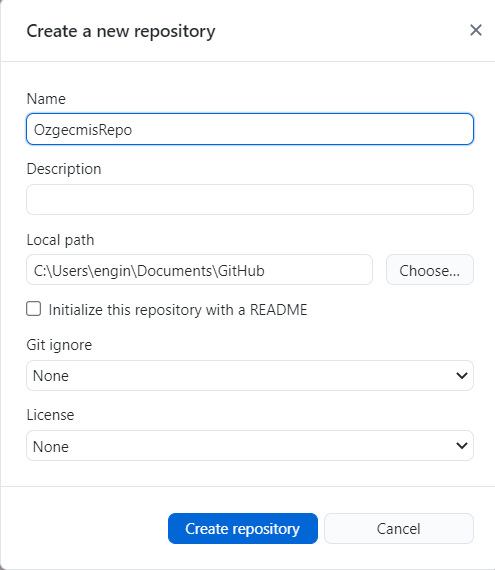


1. **```**File>New Repository yolunu takip edilir. **```**

A screenshot of a computer

Description automatically generated

1. “\*New Repository\*”**```**e tıklandıktan sonra gelen ekranda Name alanına OzgecmisRepo yazıldıktan sonra **```**\*“Create Repository”\* **```** (Repository Oluştur) butonuna basılır. **```**\*“Create Repository”\* **```**butona bastıktan sonra bir repository oluşturulur. **```**



1. **```**Bu işlem sonucunda şekildeki ekran gözükür. Bu ekran, geliştirme aşamasında en sık görülecek ekrandır. Remote repository`e kod göndermek ve Remote repository`deki son değişikliklerin kontrol edildiği ekrandır. Anasayfa niteliğindedir. **```**

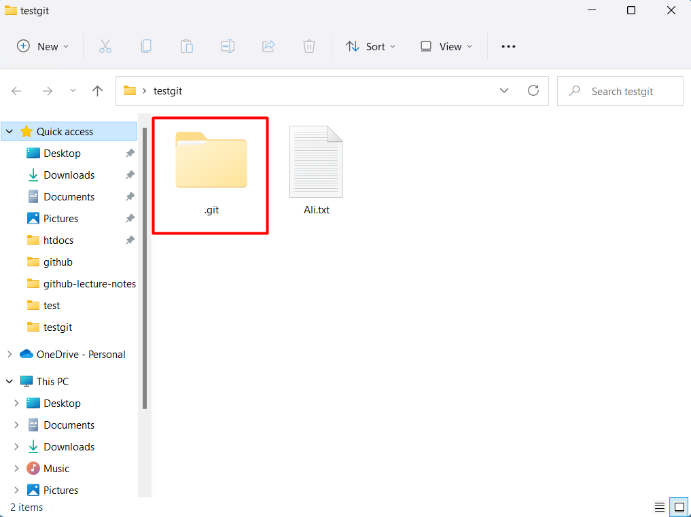
Graphical user interface, text, application, website

Description automatically generated

**```**Repository neydi? Git olayları takip edilen bir klasör. Oluşturduğu tek şey aslında budur. Bu klasörü görebilmek için yukarıdaki resimde işaretlenen KISIM 2`nin altında **```**\*“Show in Explorer”\***```**butonuna tıklanır veya kısayolu olan ctrl + shift + f yapılır. Create Repository butonuna tıklandıktan sonra şekildeki gibi **```**\***.git\*```** klasörü Dökümanlar>Github>OzgecmisRepo içerisinde otomatik oluşturulduğu görülür..

NOT: .git klasörü, komut satırına ```\***git init**\*``` yazılarak veya github desktop uygulamasından ```\*”Create Repository”\* ```diyilerek oluşturulan, Git sisteminin klasörü repository olarak algılamasını sağlayan bir klasördür```

**```**1. Haftada anlatılan Ali.txt dosyası örneğinde bahsedilen 4 satır komutun ilki olan git init komutu yazıldığında, ana proje klasörü içerisinde .git klasörü oluşur. **```**



**```**Sonuç: Create Repository butonuna tıklandığında, bir local repository(bir adet klasör ve bu klasör içerisinde .git klasörü) oluşturulur. 1. haftada anlatılan **```**\***git init\*```** komutu, Github Desktop uygulaması vasıtasıyla otomatik yürütülmüş olunur. Bu klasör Git`in çalışması için gerekli her şeyi tutar. **```**

**```**Bu aşamada repository sadece local bilgisayarda oluşturulmuş olunur. Remote Repository üzerine yapılan 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.

Not: Remote ve Local repository üzerindeki senkronizasyon, Kısım 1 olarak işaretlenen bölgedeki buton vasıtasıyla yapılır. **```**\*“Publish Repository”\* **```** butonuna tıklayalım. **```**

Text

Description automatically generated

**```**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 in Visual Studio Code”\* butonu belirir. Artık Remote ve Local repository`nin senkron(eşlenmiş repositoryler) olduğu anlamına gelir.

Bu butona tıklanması durumunda, Visual Studio Code programı çalışacaktır. Eğer VS Code uygulamasına sahip değilseniz bu linkten indirebilirsiniz: <https://code.visualstudio.com/download>

Not: Eğer geliştirme aşamasında kısım 1 de mavi bir buton görülmüyorsa, muhtemelen remote repository ile local repository senkron(birebir aynı) olmuştur. **```**

\*“Open in Visual Studio Code”\* **```**butonuna tıklandığında, VS Code programı içerisinde, Repository klasörü açılır. Açılan bu VS Code ekranında, Windows işletim sisteminden görüntülenen repository klasörünü aynen görülür**```**

A screenshot of a computer

Description automatically generated with medium confidence

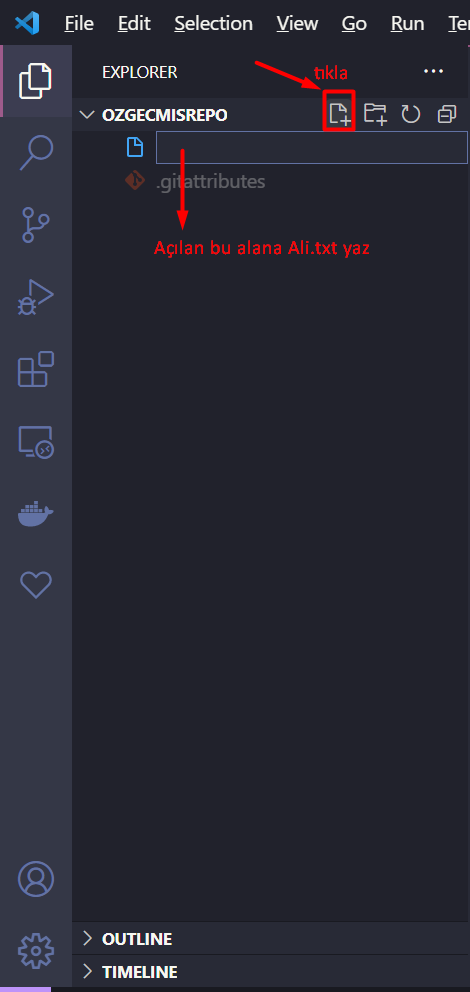
**\*VS Code nedir?\***

**```**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. **```**

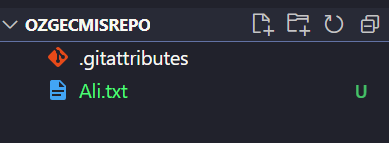
\*Uygulama\*:

**```**Local repository üzerinde Ali.txt dosyası oluşturup remote repository`e gönderelim.

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

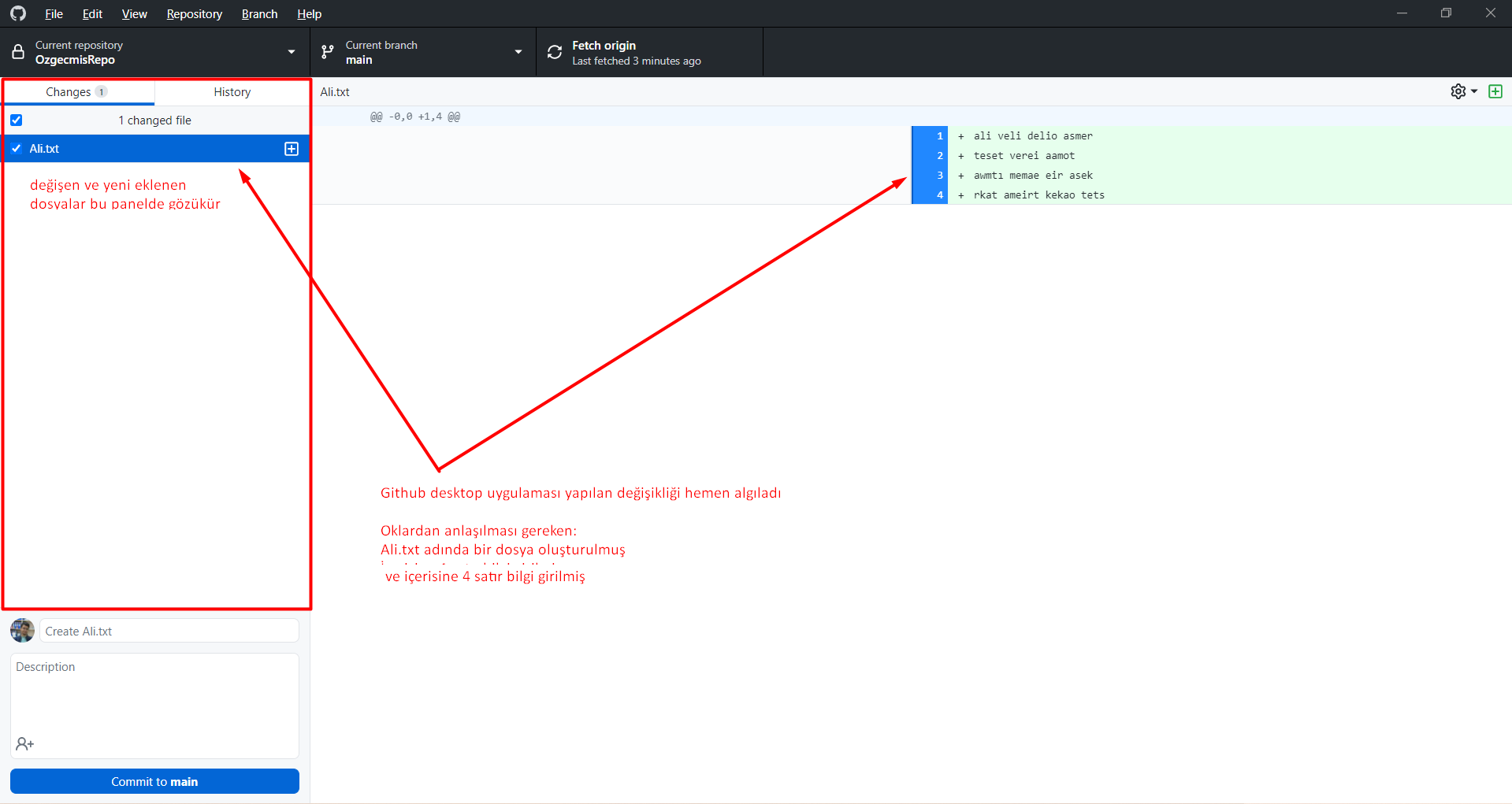


**```**Dosyayı oluşturduktan sonra içerisine 5 satırlık bir metin girelim ve ctrl + s yaparak kaydedelim. Veya File>Save yapalım. **```**



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.

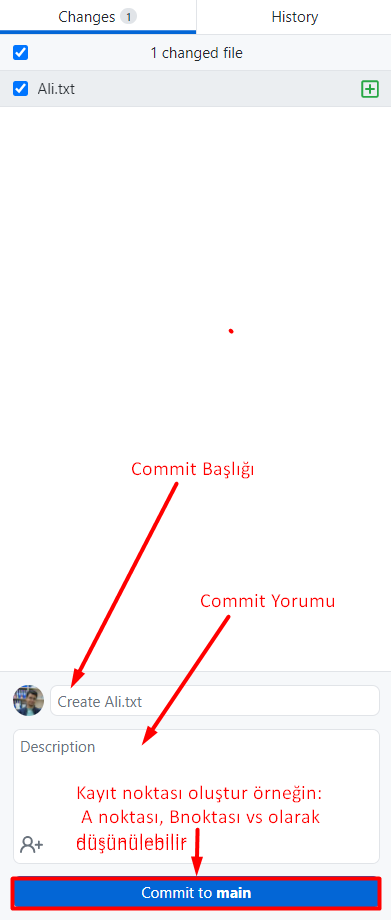
Commit Nedir? Commit o anki yapılan değişikliklerin kaydını tutar. Örnekte yer alan 1 adet dosya değişikliği olan Ali.txt dosyası commit edildikten sonra bir kayıt noktası oluşturur. 1. Hafta anlatılan A ve B noktaları commit demektir. Yukarıdaki resimde yorum yazdıktan sonra **```**\*“Commit to main”\* **```** 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. **```**

```Not: Commit, kayıt noktası aynı anlamda kullanılmıştır ve A ve B noktaları örnek kayıt noktalarıdır```

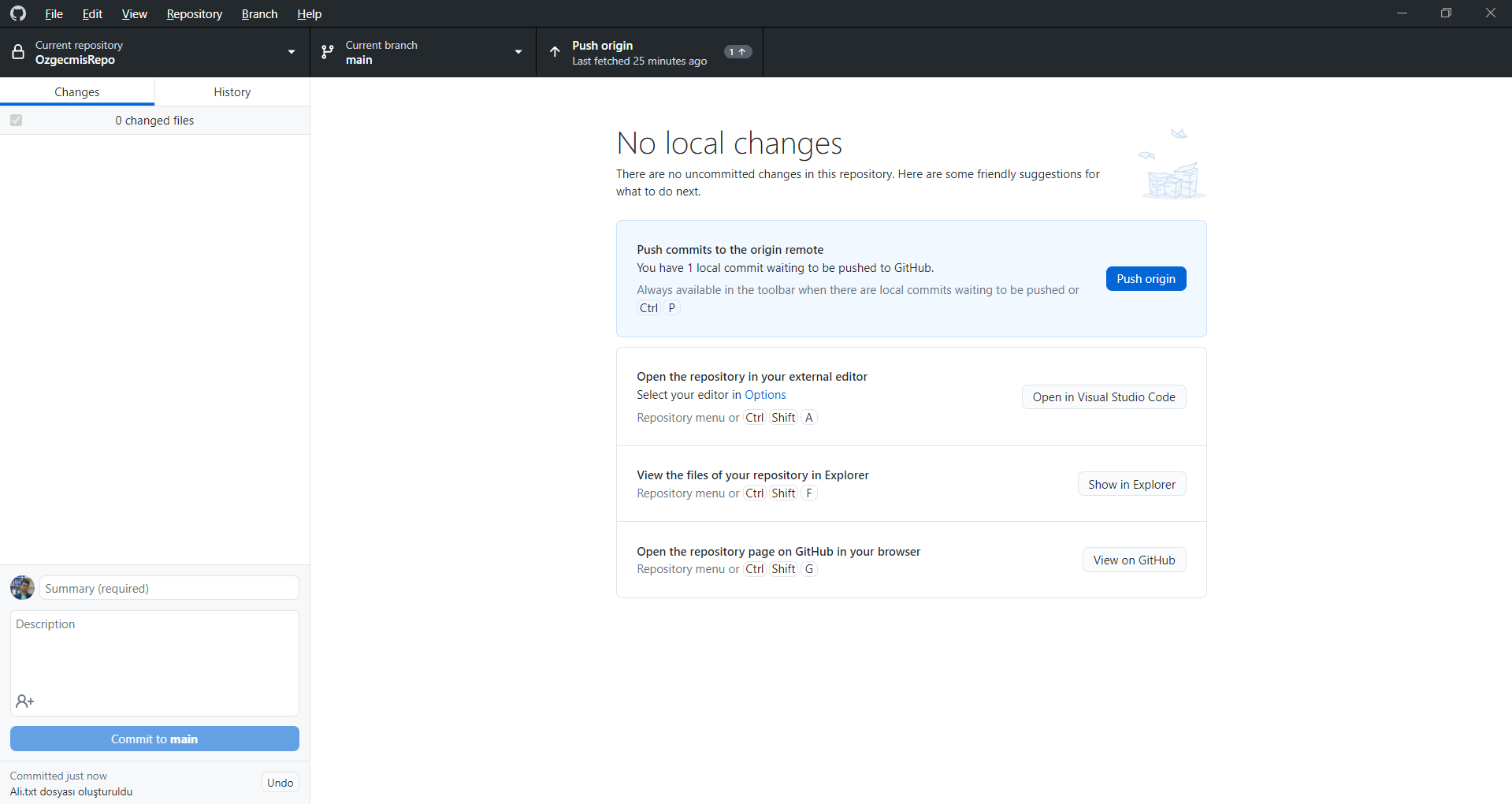
\*Uygulama - Commit ve Push\*:

**```**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

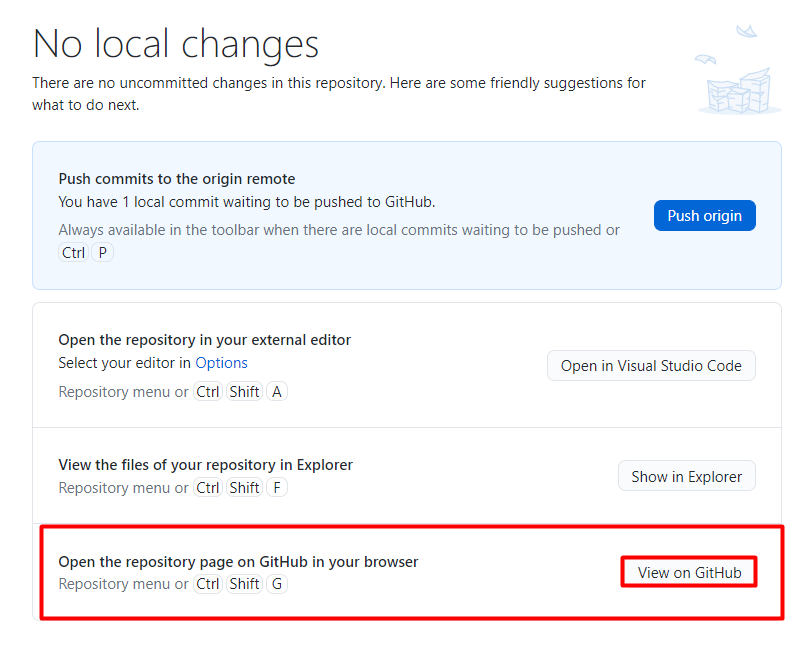
**```**Bu aşamadan sonra resimde görüldüğü gibi, KISIM 1 bölgesinde, **```**\*”Push Origin”\***```** butonu belirmiştir. **```**\*”Push Origin butonu”\***```**, Remote repository üzerine değişiklikleri atacaktır. Butona tıklayalım. **```**

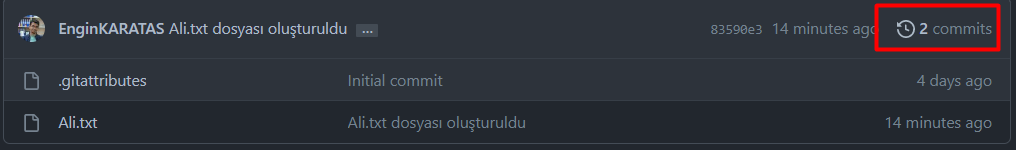


Graphical user interface, text

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 icona tıklayalım**```**

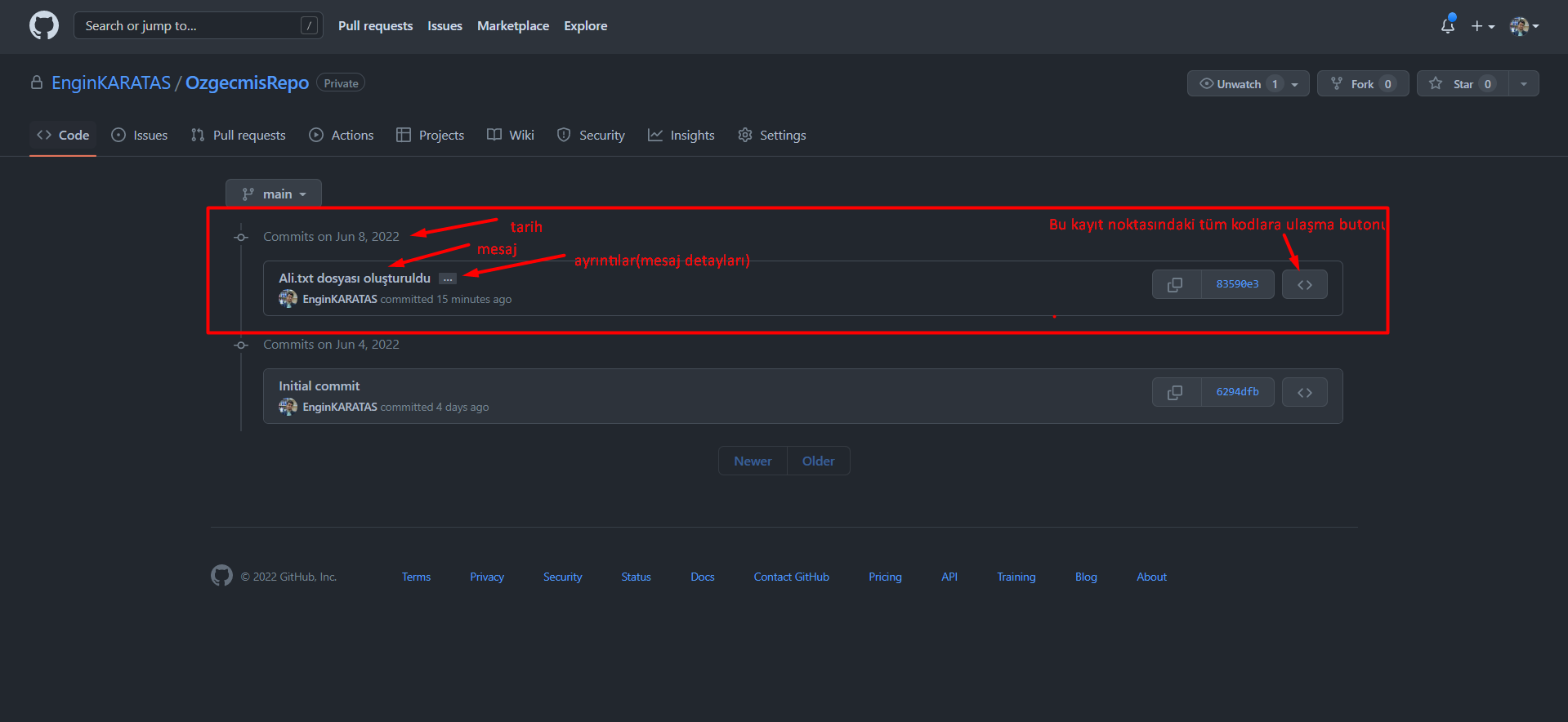
A screenshot of a computer

Description automatically generated

İkona tıklandığında aşağıdaki commit detaylarını gösteren bir sayfa açılır

A screenshot of a computer

Description automatically generated

**```**Commits iconuna 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