**1.Github üyeliği oluşturma ve organizasyona katılma:**

**Github Kayıt olma**

Github.com`a girdikten sonra Sign Up kısmına emailinizi yazarak kaydolabilirsiniz.

**Github Organizasyona katılma**

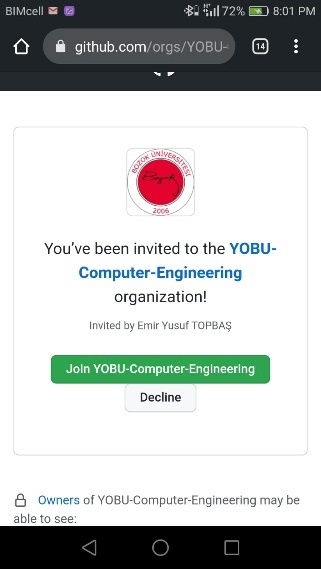
Organizasyona katılmak için organizasyon adminlerinden birinin sizi github organizasyonuna davet etmesi gerekmektedir. Emir arkadaşımız`a Github üzerinden istek gönderdikten sonra size email yoluyla gruba katılma isteği gönderecek. Organizasyon katılma isteği **E-mail`inize** gönderilecek.

Github hesabınızın kayıtlı olduğu e-maili kontrol etmeyi unutmayın.

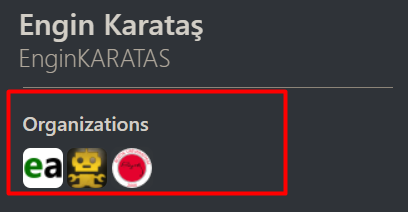
E-mail gönderildikten sonra, e-mail içerisindeki Join yazısına tıklayın.



1. Linke tıkladıktan sonra Github sitesine yönlendirileceksiniz. Uygulamayı Mobil mağzanızdan(App Store, Play Store) kurmanız durumunda uygulama üzerinden tarayıcıda giriş yapmadan kabul edebilirsiniz.



Not: Organizasyona girseniz bile profilinizde organizasyonda olduğunuzu belirtecek resmi siz hariç kimse göremez(başka hesaplardan).



Şekildeki organizasyon resimlerinin gözükmesini isterseniz, github organizasyon>peoples sekmesinden isminizin yanındaki private yazan yeri public yapmalısınız.

A screenshot of a computer

Description automatically generated with medium confidence

**2. Git nedir?**

Git, küçükten çok büyük projelere kadar her kodu hızlı ve verimli bir şekilde ele almak için tasarlanmış ücretsiz bir sürüm kontrol sistemidir. Farklı sürüm kontrol sistemleri(bunlara VCS denir) de vardır fakat Git oldukça kullanışlı ve performanslı çalışan bir VCS`dir. Git'i öğrenmesi kolaydır.

**2.1.Git ve GitHubda asıl yapmak istediğimiz şey ne?**

Kodları kontrol etmek.. Bir geliştiricinin geliştirdiği kodlarda genellikle “keşke önceki yaptığım duruma geri dönebileydim” ifadesi sık sık kullanılır. Ve Git, sizin önceden yazdığınız kodların tam o anki anına gitmenizi sağlar. Böylece korkulu geliştirme süreci sorularından “Flash diske yedekleme yaptım mı?, acaba ctrl z mi yaptım yoksa ctrl shift z mi yaptım?” kurtulmuş olunur. Bunun yerine “Önceden şöyle bir kod yazmıştım acaba bunu yeni koduma eklesem nasıl olur? gibi daha verimli sorular sormaya başlanılır.

Örnek git kullanımı:

Ali.txt isminde metin dosyası (.txt) olsun. Bu dosyada Ali`nin özgeçmişi anlatan toplam 5 satır metin olsun. Eğer Ali 3. Satırı silerse ve Git`ten yardım alıp değişiklikleri gör derse: Git, toplam 5 satır olan metnin 4 satır kaldığını ve 3. Satırın yok olduğunu söyleyecektir. Git`ten bu durumu kaydetmesi istenirse, git bu olayı kaydeder. Buna A noktası diyelim. Sonrasında buna benzer değişiklikler yapılırsa B diyelim. Git, bu A ve B gibi durumlar arasında geçiş yapılabilmesini, yapılan değişikliklerin algılanmasını ve yapılan değişikliklerin yönetilmesini kolaylaştırır. Metin dosyası örneği, kod dosyası örneği olarak anlaşılabilir. Hiçbir farkı yoktur.

Yukarıdaki örnekte A noktasını kaydetmek için git kodları aşağıdadır. Genel olarak işleyiş bu şekilde. Bu Kodları ezberlemenize gerek yok. Mantığını anlamanız yeterli. Github kullanırken bu kodlarla uğraşmayacağız. Github bu adımları kolaylaştırır.:

Yukarıdaki örnekte A noktasını kaydetmek için git kodları aşağıdadır. Genel olarak işleyiş bu şekilde. Bu Kodları ezberlemenize gerek yok. Mantığını anlamanız yeterli. Github kullanırken bu kodlarla uğraşmayacağız. Github bu adımları kolaylaştırır.:

Yukarıdaki Ali.txt örneği için Ali.txt dosyası oluştur ve 5 satır ekledikten sonra, Git komutları(cmd`ye yazılır):

\*git init\* //Git başlatmak için kullanılır.

\*git add\* . //klasör içerisindeki tüm dosyaların hareketlerini izlemeye başlar. Ve hepsinde yapılan değişiklikleri algılar

Burada Ali.txt`de 3. satırı sil :

\*git diff\* //dosyalardaki değişiklikleri algılar. Ve ekteki resimdeki sonucu verir.

Text

Description automatically generated

\*git commit -m “ilk commit”\* //değişikliklerin yapıldığı anı tarihe kaydet. Buna A noktası diyebiliriz.

**3. Github nedir?**

* Hub, sosyal ağ anlamı taşır. GitHub ise yazılımların sosyal ağıdır. Sosyal bir ortam sunmasıyla birlikte Git altyapısını kullanarak kodları yönetmemizi sağlar.
* Github, git komutlarını yazmadan, bir arayüz vasıtasıyla Git işlemleri yapılmasını kolaylaştırır.
* Git altyapısını kullanarak projede yapılan değişiklikleri komut satırı(cmd) ekranından görmek yerine, kullanıcıların rahatça anlayabileceği ekranlarda yani UI(user interface/grafiksel kullanıcı arayüzü)`da izlemesine olanak tanır.
* Dünya üzerindeki milyarlarca sayfa kodu, milyonlarca projeyi hızlıca arayabilmeyi sağlar.
* Yapılan projelerin yayımlanmasını, lisanslanmasını, yönetilmesini sağlar.
* Ekipçe kod geliştirmeyi sağlar.
* Git öğrenim süresini oldukça düşürür. Teknik bilgi gerektirmez. Geliştirme aşamalarındaki sıkıntıları en aza indirir.
* Ali.txt dosyasındaki değişiklikler istenmeyen noktaya geldiğinde, Github bunları birkaç tıklamayla kaydedilen noktalara (A, B) geri getirmeyi kolaylaştırır ve projenin önceki sürümü geri getirilir.

Özetle, GitHub Git komutlarını kullanarak riskleri ve çok fazla hata yapma korkusunu ortadan kaldırır. İş birliği yapma ve geliştirme olanağını arttırır.