

# ITU AUV Github Ödev Raporu

*Talha Karasu*

## Git Nedir ve Neden Kullanılır?

Git bir versiyon kontrol sistemidir. Genellikle yazılım projelerinde dosyaların farklı versiyonlarını yönetmek için kullanılır. Dosyaların üzerinde yapılan değişikliklerinin takibi ve kontrolünü kolaylaştırır. Farklı geliştiricilerin aynı proje üzerinde çalışmasını ve dosyaların farklı versiyonlarının birleştirilmesini sağlar. Takım çalışmasının düzenlenmesinde önemli bir araçtır.

## GitHub'ın Özellikleri ve Avantajları

GitHub, Git ile entegre çalışan bir web platformudur. Git'in birçok özelliğine görsel bir arayüz sağlar. Aynı zamanda Git kullanılarak geliştirilen projeleri internette saklamayı ve diğer geliştiriciler ile paylaşmayı da mümkün kılar. Komünitesi ve sağladığı araçlarla açık kaynak yazılım projelerinin en büyük buluşma noktasıdır. Aralıksız proje geliştirme sürecinin sıkıntısız ilerlemesindeki önemli araçlardan biri olan test sistemleri için otomasyon seçenekleri sunar. Bu sayede geliştiricilerin GitHub'a yükledikleri her kod test edilerek temel hatalardan anında haberdar olunur. Issue-Pull Request sistemiyle takım olarak geliştirilen projelerdeki hataların ve yeni özelliklerin takibini ve çözülmesini/implemente edilmesini kolaylaştıran bir arayüz sağlar.

## Temel Git Komutları ve Kullanımları

- `git init`: bulunulan konumda yeni bir git reposu oluşturmak için kullanılır
- `git clone <repository>`: bir repository'i klonlamak için kullanılır
- `git add <file>`: dosyaları commitlenmek üzere eklemek için kullanılır
- `git commit -m <commit>`: eklenen dosyaları bir mesaj girerek commitlemek için kullanılır
- `git commit --amend`: eklenen dosyaları önceki commite eklemek için kullanılır
- `git log`: önceki commitleri görmek için kullanılır
- `git status`: reponun şu anki durumunu görmek için kullanılır
- `git push <remote> <branch>`: repoyu belirtilen remote adrese pushlamak için kullanılır
- `git pull`: belirtilen remote adresten değişiklikleri getirmek için kullanılır
- `git checkout -b <branch>`: yeni bir branch oluşturur ve `git switch` ile o branche geçer

- `git merge <branch>` : İki branchi merge etmek için kullanılır
- `git rebase` : bir branchin/ birtakım commitin başlangıç noktasını değiştirmek için kullanılır. Örnek: feature branchinde ilerlendikten sonra main branchte feature'u etkileyen bir değişiklik yapıldıysa rebase ile feature branchi main'in son hali üzerine geliştirilmiş duruma getirilebilir.