

1) Repository nedir?

Cevap:

Repository (kısaca *repo*), bir projenin tüm dosyalarının, klasörlerinin, geçmiş kayıtlarının (*commit’ler*) ve sürümlerinin tutulduğu depodur.

Bilgisayardaki bir klasörün GitHub üzerindeki karşılığıdır.

2) Commit ne işe yarar?

Cevap:

Commit, projede yaptığın değişiklikleri kaydetme işlemidir.

Her commit:

- O anki değişikliklerin fotoğrafını çeker,
 - Açıklama (commit message) içerir,
 - Geri dönülmesini sağlar.
-

3) Fork ile clone arasında fark ne?

Cevap:

İşlem Açıklama

Fork Başkasının GitHub’daki reposunu hesabına kopyalarsın. Online ortamda sana ait yeni bir repo oluşur.

Clone Bir GitHub reposunu kendi bilgisayarına indirirsin. Repo senin hesabında değildir.

Kısa özet:

Fork → Hesabımı kopyala

Clone → Bilgisayarıma indir

4) Branch ne zaman kullanılır?

Cevap:

Branch, ana projeyi bozmadan yeni bir özellik geliştirmek, hata düzeltmek veya deneme yapmak istediğiinde kullanılır.

Ana (main/master) kod güvende kalır, sen yan kolda çalışırsın.

 **UYGUN 20 SORU & CEVAP**

 **BÖLÜM 1 — Temel Kavramlar**

5) Git nedir?

Cevap:

Kodun geçmişini kayıt altına alan bir sürüm kontrol sistemidir.

6) GitHub nedir?**Cevap:**

Git ile oluşturduğu projeleri internette saklayabileceğin, paylaşabileceğin platformdur.

7) README.md ne için kullanılır?**Cevap:**

Projenin tanımını, kurulumunu, kullanımını anlatan açıklama dosyasıdır. Projenin vitrinidir.

8) .gitignore dosyası ne işe yarar?**Cevap:**

Git tarafından takip edilmesini istemediğin dosyaları/klasörleri belirtirsin (ör. log, geçici dosyalar, node_modules).

9) Pull nedir?**Cevap:**

GitHub'daki son değişiklikleri kendi bilgisayarına çekmektir.

10) Push nedir?**Cevap:**

Bilgisayarında yaptığı değişiklikleri GitHub'daki repoya göndermektir.

11) Merge conflict nedir?**Cevap:**

İki kişi aynı satırı farklı şekilde düzenlerse, Git hangi değişikliğin doğru olduğunu bilemez → buna merge conflict denir.

12) Commit mesajı neden önemlidir?**Cevap:**

Yapılan değişikliği açıkça anlatır.

Takım içinde anlaşılması kolaylaştırır ve proje geçmişini okunaklı yapar.

● BÖLÜM 2 — Branch, Fork, PR Kavramları

13) Pull Request (PR) nedir?

Cevap:

Bir projeye yaptığın değişiklerin ana projeye dahil edilmesini istedigin talebe pull request denir.

14) Pull Request neden önemlidir?

Cevap:

Kodları ana projeye eklemeden önce kontrol etmeyi, yorumlamayı ve onaylamayı sağlar.

15) Branch adı seçerken nelere dikkat edilmelidir?

Cevap:

Kısa, anlaşılır ve işlevi açıklayan olmalıdır:

- feature/login-page
 - bugfix/navbar-crash
-

16) Fork yapılan projede nasıl değişiklik geri gönderilir?

Cevap:

Fork → Değişiklik yap → Commit → Push → Pull Request aç

Ana proje sahibi PR'ı görür.

17) Clone edilen projeye değişiklik gönderilebilir mi?

Cevap:

Hayır.

Clone, proje sahibinin izni olmadan yalnızca lokal çalışma yapmanı sağlar.

● BÖLÜM 3 — İş Akışı ve Gerçek Senaryolar

18) “Local repository” ve “Remote repository” farkı nedir?

Cevap:

Local → Bilgisayardaki proje

Remote → GitHub'daki proje

19) Projede birden fazla kişi varsa neden branch kullanılır?

Cevap:

Aynı dosya üzerinde çakışma (conflict) olmaması ve herkesin bağımsız çalışabilmesi için.

20) Issue nedir?

Cevap:

Hata bildirimi, iyileştirme isteği veya görev takip biletidir.

21) “Stash” ne işe yarar?**Cevap:**

Yarım kalan değişiklikleri geçici olarak saklar; commit yapmak istemeyen biri stash kullanır.

22) Commit’ler neden küçük ve sık yapılmalıdır?**Cevap:**

- Geri dönmek daha kolay olur
 - Değişiklikler anlaşılır olur
 - Sorun tespiti hızlı yapılır
-

23) GitHub'da “Star” vermek ne işe yarar?**Cevap:**

Beğendiğin veya izlemek istediğiniz projeyi işaretlemiş olursun. Favorilere almak gibidir.

● BÖLÜM 4 — Ekstra Kavramlar (Öğretmen için)**24) “Forked from ...” yazısı ne anlama gelir?****Cevap:**

O repo başka birinin projesinden kopyalanarak oluşturulmuş demektir.

25) Tag nedir?**Cevap:**

Projenin belirli bir sürümüne verilen işaret (ör. v1.0).

26) Release nedir?**Cevap:**

Tag'lenmiş sürümün kullanıcılar için paketlenmiş halidir.
Örneğin bir yazılımın “indirilebilir” versiyonudur.

27) Contributor kimdir?**Cevap:**

Projeye kod, düzeltme, geliştirme, dökümantasyon vb. katkı sağlayan kişidir.

28) GitHub Actions nedir?

Cevap:

Projede otomatik görevler oluşturmaya yarar (testler, deploy vb.)