**[GOLDBERG\_MED\_INT1\_PROJECT](https://github.com/Duygu-Jones/GOLDBERG_MED_INT1_PROJECT" \t "_blank) GITHUP REPO**

**GitHub Desktop Kullanım Protokolü:**

1. **Kendi Branch'inizde Çalışın:**
   * Kendi adınıza **yeni bir branch** açın ve **sadece bu branch'te** çalışın.
   * Başkalarının branch'ine veya main branch'ine asla müdahale etmeyin.
   * Yeni bir branch açarken:
     + GitHub Desktop’ta "Current Branch" menüsünden "New Branch" seçeneğini kullanın.
2. **Güncellemeleri Almadan Çalışmaya Başlamayın:**
   * Günlük çalışmaya başlamadan önce, main branch'teki son değişiklikleri kendi branch'inize aktarmayı unutmayın:
     + **GitHub Desktop'ta** Fetch origin butonuna tıklayın. Bu, main branch'teki son güncellemeleri yerel kopyanıza alır.
     + Ardından, **GitHub Desktop'ta** Branch menüsünden Merge into Current Branch seçeneğini kullanarak main branch'ini kendi branch'inize merge edin.
     + **Önemli:** main branch'teki güncel değişiklikleri aldıktan sonra kendi branch'inizde çalışmaya devam edin.
3. **Commit Mesajlarını Düzgün Yazın:**
   * Sık sık commit yaparak ilerleyin ve açıklayıcı mesajlar kullanın.
   * Örnek: "Erdem - veri temizleme tamamlandı."
   * **GitHub Desktop'ta** Commit to [branch\_name] butonunu kullanarak değişikliklerinizi commit edin.
4. **Merge Conflict'leri Çözme:**
   * main branch'teki güncel değişiklikleri kendi branch'inize merge ettiğinizden emin olun.
   * main branch'i kendi branch'inize merge ederken **merge conflict** oluşursa, GitHub Desktop'ta conflict'leri çözün ve daha sonra commit yapın.
   * **GitHub Desktop'ta** Resolve conflicts ve Commit merge seçeneklerini kullanarak bu durumu yönetebilirsiniz.
5. **Başkalarının Dosyalarına Müdahale Etmeyin:**
   * **Kendi klasörünüzde ve kendi dosyalarınızda** çalışın.
   * Başkalarının dosyalarına veya branch'lerine müdahale etmeyin.
6. **Dosya İsimlerini Standartlaştırın:**
   * Dosya isimlerinize **isim baş harflerinizi** ekleyin (örnek: E\_analysis.ipynb).
   * İsimleri **kısa ve öz** tutun, çünkü GitHub uzun dosya isimlerinde sorun çıkarabilir.
7. **Gereksiz Dosyaları Commit Etmeyin:**
   * Büyük dosyaları veya geçici dosyaları (örneğin .csv yedek dosyaları) .gitignore listesine ekleyin.
   * Bu dosyalar gereksiz yere repoyu şişirir ve merge işlemlerinde sıkıntı yaratabilir.

**Özet:**

* **Kendi branch'inizde çalışın** ve main'den güncellemeleri düzenli olarak **merge ederek** alın.
* **Calismaniz bittikten sonra gunluk commit yapın** ve commit mesajlarınızı açıklayıcı tutun.
* **Dosya isimlerini düzenli tutun** ve **başkalarının çalışmalarına müdahale etmeyin**.

Bu güncellemelerle, main branch'teki güncellemelerin kendi branch'lere doğru bir şekilde aktarılmasını ve merge edilmesini sağlamış oluruz. Bu süreç, projenin güncel ve sorunsuz bir şekilde ilerlemesine yardımcı olacaktır.