## Proje Büyüklüğünü Tahminlemesi Nasıl Yapıldığı:

## Task uzunluğunun 10 ile modu kadar gün olarak tahmin ediliyor.

## Yazılım geliştirmede yazılım geliştirme kalıbı kullanıldı mı?:

## Evet, Express tabanlı rest api ve react ile beraber yazıldı.

## Tasarım ve Görsellik için ilave bir araç kullanıldı mı ?

## Kullanımı kolay mı?: Evet, Vuexy adlı react template ve react-beautiful-dnd npm kütüphanesi kullanıldı.

## Test Cases

**Test Case:1**

|  |  |
| --- | --- |
| Risk Level | Yüksek |
| Purpose | Bu test durumunda kullanıcı giriş yapmadan tasks kısmına erişmeye çalışıyor. |
| Inputs | Klavye, Fare |
| Expected Outputs | Tasks kısmına erişilmeden giriş kısmına gönderilmesi. |
| Pass Criterias | Tasks kısmı karşılanmadan önce giriş kontrolü yapılmalı, eğer giriş yok ise giriş kısmına gönderilmesi. |
| Fail Criterias | Giriş yapılmadan tasks kısmına erişilmesi. |
| Test Procedure | Yazılım ve desteklenen sistem, test kullanıcısı tarafından cihazlarda çalıştırarak  belirtilen şekilde testi gerçekleştirmektedir. Test işleminin ardından test sonucunu Pass/Fail olarak belirterek nedenleri ile birlikte raporlamalıdır. |

**Test Case:2**

|  |  |
| --- | --- |
| Risk Level | Düşük |
| Purpose | Bu test durumunda kullanıcı boş bir task oluşturmaya çalışıyor. |
| Inputs | Klavye, Fare |
| Expected Outputs | Boş task oluşturulmamalı. |
| Pass Criterias | Boş task oluşturulmadan önce kontrol edilmeli ve kullanıcıya bir uyarı vermedilir. |
| Fail Criterias | Boş taskin oluşturularak veritabanına eklenmesi. |
| Test Procedure | Yazılım ve desteklenen sistem, test kullanıcısı tarafından cihazlarda çalıştırarak  belirtilen şekilde testi gerçekleştirmektedir. Test işleminin ardından test sonucunu Pass/Fail olarak belirterek nedenleri ile birlikte raporlamalıdır. |

**Black Box Test**

1. *Gereksinim*: Task açıklaması olması.
2. *Girdiler*:
   1. Geçerli: Fırına git.
   2. Geçersiz:
3. *Çıktılar*:
   1. Geçerli task için hata mesajı gösterilmez.
   2. Geçersiz task için “Lütfen bilgilerinizi kontrol ediniz” mesajı gösterilir.
4. *Testler*:
   1. Geçerli task ekleme durumunda mesajın gösterilmeyeceğini belirten test
   2. Geçersiz tesk ekleme durumunda ”Lütfen bilgilerinizi kontrol ediniz” mesajının gösterileceğini belirten test
5. Uygulama açılır ve testler koşulur.
6. 4. adımda belirlenen testlerin çalışması sonucunda elde edilen çıktılar ile 3. adımda belirtilen çıktılar karşılaştırılır.
7. Her iki testte başarılı ise uygulama doğru çalışıyor kabul edilir aksi takdirde uygulama başarısızdır.

**White Box Test**

React-app-rewider adlı npm pakedinin test komutu sayesinde proje teste sokuluyor ve belirtilen test kısımlarından bir hata olup olmadığına bakılıyor.

**Database Diagram**

