

TEST RAPORU

1. Ön İnceleme

Yapılan çalışma Ubuntu 22.04 işletim sistemi üzerinde, Google Chrome Version 127.0.6533.72 kullanılarak gerçekleştirilmiştir.

İncelenilmesi istenilen web sitesinin iki arayüzden oluştuğu, ilk sayfanın asıl uygulama sayfasına giriş yapabilmek için bir LOGIN sayfası diğer sayfanın ise TO-DO APP isimli uygulama sayfası olduğu görülmüştür.

LOGIN PAGE analizleri: Bu arayüzde sayfanın sol kısmında bir resim ve hemen sağ tarafında input alanlarının bulunduğu bir form elemanının olduğu görülmüştür. İlk olarak buradaki beklentinin asıl uygulama sayfasına geçiş yapabilmek için doğrulama işlevine sahip bir giriş sayfası olduğu değerlendirilmiştir.

TO-DO APP analizleri: Başarılı login işleminin ardından karşılaşılan yeni arayüzde bir yapılacaklar listesinin oluşturulmak istenildiği düşünülmektedir.

2. Tespit Edilen Hatalar

2.1. Login Page

2.1.1. Placeholder ile yıldızlı olarak gösterim sağlanan isim ve e-mail input alanları zorunlu alan olarak algılanmasına rağmen;

2.1.1.a. İsim ve e-mail alanları her ikisinde boş bırakıldığında giriş yapılmamasına rağmen ekranda herhangi bir uyarı mesajı ile karşılaşılmamaktadır.

2.1.1.b. İsim alanı boş e-mail alanı dolu olduğunda giriş yapılmamasına rağmen herhangi bir uyarı mesajı ile karşılaşılmamaktadır.

2.1.1.c. İsim alanı dolu e-mail alanı boş olduğunda giriş yapılabilir. Beklenti email alanında dolu olması halinde giriş yapılabilmesi yönündedir.

2.1.1.ç. İsim alanı ve e-mail alanlarının beraber dolu olduğu durumda giriş yapılabildiği görülmüştür. Bu durumda başarılı giriş yapıldığına dair uç kullanıcının bilgilendirilmesinin daha iyi olacağı değerlendirilmektedir..

2.1.2. Login sayfasında bulunan isim ve e-mail alanlarının önyüz validasyon kontrolleri;

2.1.2.a. İsim alanında sadece string bir input girişi yapılabilmesi gerektiği algılanmasına rağmen harici durumların (rakam, özel karakter , whitespace trimleme, html elements) kontrol edilmediği gözlemlenmiştir.

2.1.2.b. E-mail alanında bir mail adresi girişi yapılması gerektiği anlaşılmasına rağmen html elemanın type özelliği text olarak belirtilmiş ve bu alana geçerli bir e-mail yazımı olmasa dahi login işlemleri yapılabildiği gözlemlenmiştir.

2.1.2.c. Input alanlarının min karakter uzunluğunun belirtilmediği görülmüştür.

2.1.2.ç. Input alanlarında herhangi bir regex validation bulunmadığı ve olması gerektiği düşünülmektedir.

2.1.3. Html elemanları arasında bir img bulunmasına rağmen image boyutu 0*0 ve src belirtilmediği için arayüzde herhangi bir image görüntülenememektedir.

2.1.4. Buton elementinin enter tuşuna basıldığında etkileşiminin olmadığı görülmüştür.

TEST RAPORU

2.2. To-Do App Page

2.2.1. Başarılı login yapıldığı kabul edildiğinde yönlendirilen sayfa için herhangi bir endpoint tanımlanmadığı ve bu durumda sayfa yenilendiğinde yeniden login sayfasına geçiş yapıldığı görülmüştür.

2.2.2. Body elementinin sayfanın tamamını kapsamadığı görülmüştür.

2.2.3. Input alanına girilecek değerlerde;

2.2.3.a. Özel karakter validasyonu bulunmadığı,

2.2.3.b. Minimum karakter kontrolü yapılmadığı,

2.2.3.c. Maksimum karakter kontrolü yapılmadığı,

2.2.3.ç. Html etiketlerinin girilebildiği görülmüştür.

2.2.4. Ekleme butonunun enter tuşu ile etkileşimde bulunmadığı görülmüştür.

2.2.5. Liste elemanlarının;

2.2.5.a. Uzun karakter sayısına sahip olduğunda ilgili alan dışına taşıdığı word wrap özelliğinin bulunmadığı,

2.2.5.b. Sahip oldukları “-” butonlarının ilgili elemanı değil her zaman listenin ilk elemanını listeden çıkardığı görülmüştür.

3. Başarılı Durumlar

3.1. Login Page

3.1.1. Sayfa genel yapısının responsive olduğu,

3.1.2. Sayfanın title ifadesine ve icona sahip olduğu,

3.1.3. Login butonunun başarılı giriş yapıldığı varsayıldığında ilgili to-do app sayfasının görüntülenmesini sağladığı görülmüştür.

3.2. To-Do App Page

3.2.1. Sayfa genel yapısının responsive olduğu,

3.2.2. Add butonu aracılığıyla input alanına girilen ifadelerin listeye eklendiği,

3.2.3. Liste elemanları eklenirken kullanılan whitespacelerin trimlendiği görülmüştür.

4. Tasarlanan Test Senaryoları

Tasarlanan test senaryoları Login sayfası ve To-Do App sayfası için ayrı ayrı tablolara yerleştirilmiştir.

4.1. Login Page Test Senaryoları

Test Senaryo ID	Test Senaryo Adı	Ön Koşul	Test Adımları	Beklenen Sonuç	Alınan Sonuç	Test Metodolojisi	Statü
1	Boş alanlarla giriş denemesi	Login sayfasına gidilmesi.	1.İsim alanını boş bırakın. 2.E-mail alanını boş bırakın. 3.Giriş butonuna tıklayın.	Kullanıcıya, alanların doldurulması gerektiğini belirten bir uyarı mesajı gösterilmesi.	Başarısız	Fonksiyonel Test: Formun doğru çalıştığının kontrolü.	Yüksek

TEST RAPORU

2	İsim alanı boş, email alanı dolu	Login sayfasına gidilmesi.	1.İsim alanını boş bırakın. 2.E-mail alanına geçerli bir e-mail girin. 3.Giriş butonuna tıklayın.	Kullanıcıya, isim alanının doldurulması gerektiğini belirten bir uyarı mesajı gösterilmesi.	Başarısız	Fonksiyonel Test: Form alanları zorunluluğunun kontrolü.	Yüksek
3	İsim alanı dolu, email alanı boş	Login sayfasına gidilmesi.	1.İsim alanına geçerli bir isim girin. 2.E-mail alanını boş bırakın. 3.Giriş butonuna tıklayın.	Kullanıcıya, email alanının doldurulması gerektiğini belirten bir uyarı mesajı gösterilmesi.	Başarısız	Fonksiyonel Test: E-posta alanı validasyonu kontrolü.	Yüksek
4	Mİnimum karakter sayısına sahip isim girdisi	Login sayfasına gidilmesi.	1.İsim alanına tek karakter girin. 2.E-mail alanına geçerli bir e-mail girin. 3.Giriş butonuna tıklayın.	Kullanıcıya, minimum karakter sayısına uygun isim girilmesi gerektiğini belirten bir uyarı mesajı gösterilmesi.	Başarısız	Birim Testi: İsim alanı için giriş minimum karakter kontrolü.	Orta
5	Geçersiz e-mail formatı ile giriş	Login sayfasına gidilmesi.	1.İsim alanına geçerli bir isim girin. 2.E-mail alanına geçersiz bir e-mail formatı girin. 3.Giriş butonuna tıklayın.	Kullanıcıya, geçerli bir e-mail formatı girilmesi gerektiğini belirten bir uyarı mesajı gösterilmesi.	Başarısız	Birim Testi: E-posta doğrulama fonksiyonu test edilir.	Orta
6	İsim alanına özel karakter girilmesi	Login sayfasına gidilmesi.	1.İsim alanına özel karakterler girin. 2.E-mail alanına geçerli bir e-mail girin. 3.Giriş butonuna tıklayın.	Kullanıcıya, isim alanında sadece harflerin kabul edildiği yönünde bir uyarı mesajı gösterilmesi.	Başarısız	Birim Testi: İsim alanı için giriş doğrulama kontrolü.	Orta
7	Başarılı giriş yönlendirme	Login sayfasına gidilmesi.	1.İsim alanına geçerli bir isim girin. 2.E-mail alanına geçerli bir e-mail girin. 3.Giriş butonuna tıklayın.	Kullanıcı başarılı bir şekilde To-Do App Page'e yönlendirilmesi.	Başarılı	Fonksiyonel Test: Başarılı giriş sonrası yönlendirme kontrolü.	Yüksek

TEST RAPORU

4.2. To-Do App Test Senaryoları

Test Senaryo ID	Test Senaryo Adı	Ön Koşul	Test Adımları	Beklenen Sonuç	Alınan Sonuç	Test Metodolojisi	Statü
1	Listeye yeni eleman eklenmesi	Login sayfasından başarılı giriş yapıp, to-do app sayfasına ulaşılması.	1.İlgili input alanına bir text girin. 2.Ekleme butonuna tıklayın.	Girilen ifadenin listeye eklenmesi.	Başarılı	Fonksiyonel Test: Listeye eleman ekleme kontrolü.	Yüksek
2	Özel karakter ve HTML etiketi ile liste elemanı eklenmesi	Login sayfasından başarılı giriş yapıp, to-do app sayfasına ulaşılması.	1.İlgili input alanına bir html scripti girin. 2.Ekleme butonuna tıklayın.	Kullanıcıya, geçersiz karakterler veya etiketlerin kullanılamayacağı bir uyarı mesajı gösterilmesi.	Başarısız	Güvenlik Testi: HTML/JS Injection güvenliği kontrolü.	Yüksek
3	Uzun metin eklenmesi	Login sayfasından başarılı giriş yapıp, to-do app sayfasına ulaşılması.	1.İlgili input alanına bir sayfalık bir metin girin. 2.Ekleme butonuna tıklayın.	Karakter sınırını aşma uyarı mesajı gösterilmesi.	Başarısız	Fonksiyonel Test: Karakter sınırı kontrolü.	Orta
4	Word wrap kontrolü yapılması	Login sayfasından başarılı giriş yapıp, to-do app sayfasına ulaşılması.	1.İlgili input alanına bir sayfalık bir metin girin. 2.Ekleme butonuna tıklayın.	Liste elemanının, alan dışına taşmadan düzgün bir şekilde görüntülenmesi	Başarısız	Kullanılabilirlik Testi: Word Wrap kontrolü.	Orta
5	Liste elemanı silinmesi	Login sayfasından başarılı giriş yapıp, to-do app sayfasına ulaşılması.	1.Listenin herhangi bir elemanı için “-” butonuna tıklayın.	Tıklanan elemanın listeden başarılı bir şekilde çıkarılması ve diğer elemanlar doğru şekilde sırada bulunması.	Başarısız	Fonksiyonel Test: Eleman silme fonksiyonu kontrolü.	Yüksek
6	Sayfa yenilendiğinde girişin korunması	Login sayfasından başarılı giriş yapıp, to-do app sayfasına ulaşılması.	1.Sayfanın yenilenmesi.	Kullanıcı, giriş yapmış haliyle To-Do sayfasının yenilenmesi.	Başarısız	Fonksiyonel Test: Oturum yönetimi kontrolü.	Yüksek
7	Boşluk trimleme kontrolü	Login sayfasından başarılı giriş yapıp, to-do app sayfasına ulaşılması.	1.İlgili input alanına ardarda birden fazla boşluk içeren yazı girin. 2.Ekleme butonuna tıklayın.	Listeye eklenen girdinin whitespacelerden arındırılmış olarak listede görünmesi.	Başarılı	Kullanılabilirlik Testi: Boşluk trimleme kontrolü.	Düşük

TEST RAPORU

5. Playwright ile Yapılan Testler Çıktıları

```
7 Failed
[chromium] › loginpagetests.spec.js:10:9 › Login Page Tests › Missing name and e-mail -----
[chromium] › loginpagetests.spec.js:27:9 › Login Page Tests › Missing name and valid e-mail ----
[chromium] › loginpagetests.spec.js:44:9 › Login Page Tests › Valid name and missing e-mail ----
[chromium] › loginpagetests.spec.js:61:9 › Login Page Tests › Image Detection -----
[chromium] › todoapptests.spec.js:32:9 › To-Do App Page Tests › Remove item from list -----
[chromium] › todoapptests.spec.js:58:9 › To-Do App Page Tests › Add a HTML tag to the list ----
[chromium] › todoapptests.spec.js:76:9 › To-Do App Page Tests › Keep logged in after reload the page
5 passed (1.0s)
```

5.1. Login Page : Bu kısım, giriş sayfasındaki çeşitli senaryoları test etmek için tasarlanan bir dizi testten oluşmaktadır. Her testten önce tarayıcı belirtilen URL'e yönlendirilir. Bu sayfada test edilen senaryolar:

5.1.1. İsim ve E-posta Girdilerinin Eksik Olduğu Durum : Amaç girdilerin eksik olduğu durumda hata mesajının görüntülenip görüntülenmediğini test etmektir. İsim ve e-posta alanları boş bırakılır, "Login" butonuna tıklanır ve hata mesajının görünür olması beklenir.

Sonuç: Başarısız.

```
1) [chromium] › loginpagetests.spec.js:10:9 › Login Page Tests › Missing name and e-mail -----

Error: Timed out 5000ms waiting for expect(locator).toHaveText(expected)

Locator: locator('.alert-message')
Expected string: "Tüm alanları doldurmanız gerekmektedir."
Received: <element(s) not found>
Call log:
  - expect.toHaveText with timeout 5000ms
  - waiting for locator('.alert-message')

22 |         // Expect to see the alert message
23 |         const alert = page.locator('.alert-message');
> 24 |         await expect(alert).toHaveText('Tüm alanları doldurmanız gerekmektedir.');
```

at /home/esra/Downloads/testt/slu/tests/loginpagetests.spec.js:24:29

attachment #1: screenshot (image/png) -----
test-results/loginpagetests-Login-Page-Tests-Missing-name-and-e-mail-chromium/test-failed-1.png

```
3   test.describe("Login Page Tests", () => {
10     test("Missing name and e-mail", async ({ page }) => {
21
22         // Expect to see the alert message
23         const alert = page.locator('.alert-message');
24         await expect(alert).toHaveText('Tüm alanları doldurmanız gerekmektedir.');
```

Error: Timed out 5000ms waiting for expect(locator).toHaveText(expected) Missing name and e-mail

Error: Timed out 5000ms waiting for expect(locator).toHaveText(expected)

Locator: locator('.alert-message')

Expected string: "Tüm alanları doldurmanız gerekmektedir."

Received: <element(s) not found>

Call log:

- expect.toHaveText with timeout 5000ms

TEST RAPORU

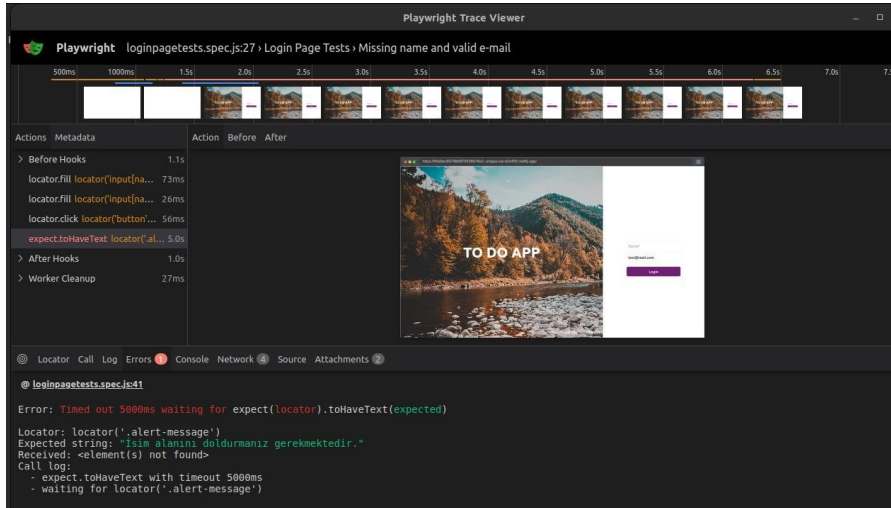
5.1.2. Eksik İsim ve Geçerli E-posta : Amaç isim girdisinin eksik olduğu durumda hata mesajının alınmasını test etmektir. İsim alanı boş bırakılır, e-posta alanı geçerli bir e-posta ile doldurulur, "Login" butonuna tıklanır ve hata mesajının görünür olması beklenir. **Sonuç:** Başarısız.

```
2) [chromium] › loginpagetests.spec.js:27:9 › Login Page Tests › Missing name and valid e-mail —

Error: Timed out 5000ms waiting for expect(locator).toHaveText(expected)

Locator: locator('.alert-message')
Expected string: "İsim alanını doldurmanız gerekmektedir."
Received: <element(s) not found>
Call log:
  - expect.toHaveText with timeout 5000ms
  - waiting for locator('.alert-message')

39 |         // Expect to see the alert message
40 |         const alert = page.locator('.alert-message');
> 41 |         await expect(alert).toHaveText('İsim alanını doldurmanız gerekmektedir.');
```



TEST RAPORU

5.1.3. Geçerli İsim ve Eksik E-posta : Amaç geçerli isim ve eksik e-posta girdileriyle hata mesajının görüntülenmesini test etmektir. İsim alanına geçerli bir isim girilir, e-posta alanı boş bırakılır, "Login" butonuna tıklanır ve hata mesajının görünür olması beklenir.

Sonuç: Başarısız.

```

3) [chromium] › loginpagetests.spec.js:44:9 › Login Page Tests › Valid name and missing e-mail —

Error: Timed out 5000ms waiting for expect(locator).toHaveText(expected)

Locator: locator('.alert-message')
Expected string: "E-mail alanını doldurmanız gerekmektedir."
Received: <element(s) not found>
Call log:
  - expect.toHaveText with timeout 5000ms
  - waiting for locator('.alert-message')

56 |         // Expect to see the alert message
57 |         const alert = page.locator('.alert-message');
> 58 |         await expect(alert).toHaveText('E-mail alanını doldurmanız gerekmektedir.');    |                               ^
59 |     });
60 |
61 |     test("Image Detection", async ({ page }) => {

at /home/esra/Downloads/testt/siu/tests/loginpagetests.spec.js:58:29

```

attachment #1: screenshot (image/png)

test-results/loginpagetests-Login-Page--12043-lid-name-and-missing-e-mail-chromium/test-failed-1.png

```

56 // Expect to see the alert message
57 const alert = page.locator('alert-message');
58 await expect(alert).toHaveText('E-mail alanın doldurmanız gerekmektedir.');
```

Error: Timed out 5000ms waiting for expect(locator).toHaveText(expected) Valid name and missing e-mail

Error: Timed out 5000ms waiting for expect(locator).toHaveText(expected)

Locator: locator('.alert-message')

Expected string: "E-mail alanını doldurmanız gerekmektedir."

Received: <element(s) not found>

Call log:

- expect toHaveText with timeout 5000ms
- waiting for locator('.alert-message')

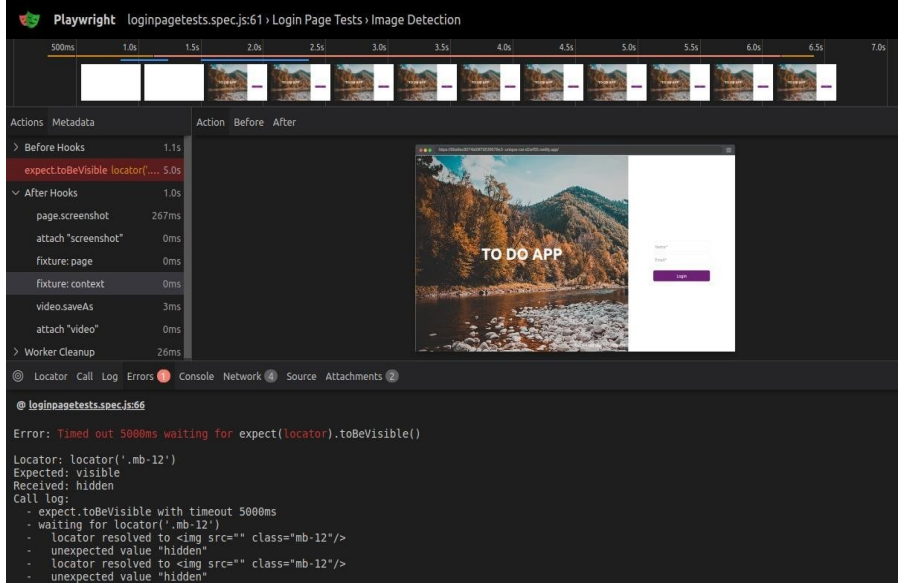
at tests/loginpagetests.spec.js:58:29

5.1.4. Görselin Görünürlüğü : Amaç form inputlarının üstünde bulunan resmin doğru yüklendiğini ve görüldüğünü test etmektir. Görselin sayfada görünür olup olmadığı ve boyutunun sıfırdan büyük olması kontrol edilir.

Sonuç: Başarısız.

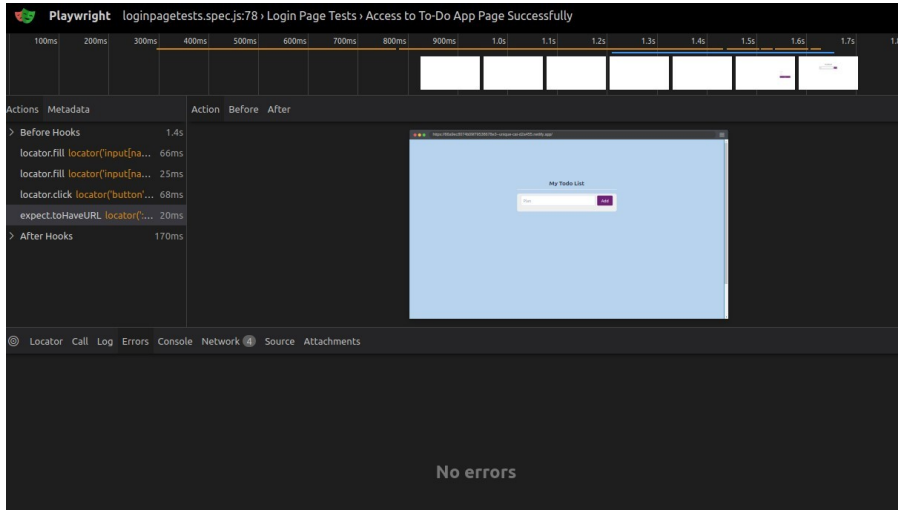
[illegible]

TEST RAPORU



5.1.5. Geçerli İsim ve E-posta ile To-Do App Sayfasına Giriş : Amaç geçerli isim ve e-posta ile to-do uygulama sayfasına başarılı bir şekilde erişim sağlandığını test etmektir. Geçerli isim ve e-posta girilir, "Login" butonuna tıklanır ve to-do app sayfasının yüklenip yüklenmediği kontrol edilir.

Sonuç: Başarılı.



5.1.6. Farklı Ekran Boyutlarında Görünürlük : Amaç giriş sayfasının farklı ekran boyutlarında doğru şekilde görünüp görünmediğini test etmektir. Tarayıcı için farklı boyutlar belirlenir, sayfa yenilenir ve sayfa elemanlarının erişilebilir olması beklenir.

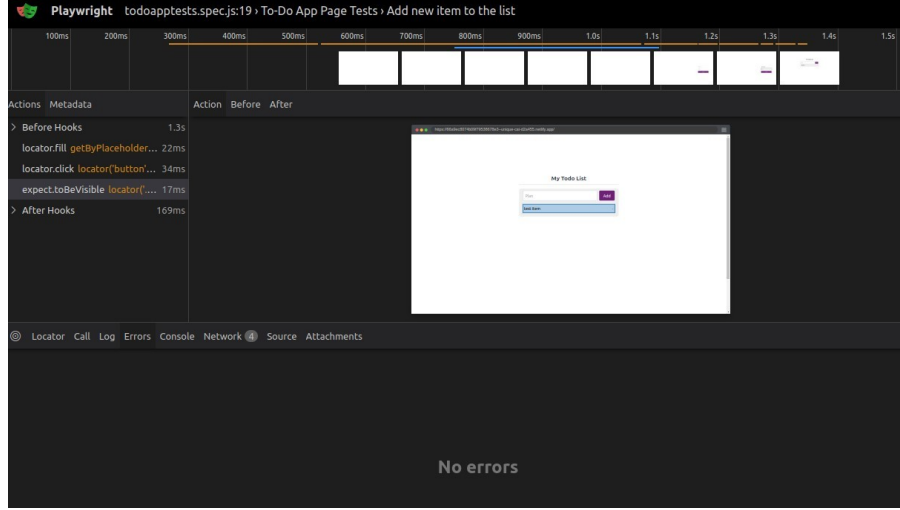
Sonuç: Başarılı.

5.2. To-Do App Page : Bu kısım, to-do app sayfasının çeşitli yönlerini test etmek için tasarlanmıştır. Her bir testten önce, giriş sayfasında geçerli verilerle to-do app sayfasına girilmiş olması sağlanmıştır. İşte bu sayfada test edilen senaryolar:

TEST RAPORU

5.2.1. Listeye Eleman Ekleme : Amaç kullanıcının listeye bir eleman ekleyebildiğini test etmektir. İntput alanına bir girdi yazılır, "Add" butonuna tıklanır ve öğenin listeye eklenip eklenmediği kontrol edilir.

Sonuç: Başarılı.



TEST RAPORU

5.2.2. Listedeki Eleman Çıkarma : Amaç kullanıcının listeden bir eleman çıkarabildiğini test etmektir. Input alanına tekrarlı olarak girdiler yazılır ve "Add" butonuna tıklanır, listenin bir öğesi çıkarılır ve diğer öğelerin listedeki yerini koruyup korumadığı kontrol edilir.

Sonuç: Başarısız.

```
5) [chromium] › todoapptests.spec.js:32:9 › To-Do App Page Tests › Remove item from list

Error: Timed out 5000ms waiting for expect(locator).toHaveText(expected)

Locator: locator('mb-2').first()
Expected string: "item1"
Received: <element(s) not found>
Call log:
  - expect.toHaveText with timeout 5000ms
  - waiting for locator('mb-2').first()

48 |
49 | // Expect the 3rd item to be removed from the list
> 50 | await expect(page.locator("mb-2").nth(0)).toHaveText("item1");
    |                                         ^
51 |
52 | // Verify the correct items remain
53 | await expect(page.locator("mb-2").nth(2)).toHaveText("item4");

at /home/esra/Downloads/testt/siu/tests/todoapptests.spec.js:50:51

attachment #1: screenshot (image/png)
test-results/todoapptests-To-Do-App-Page-Tests-Remove-item-from-list-chromium/test-failed-1.png
```

```
48 |
49 | // Expect the 3rd item to be removed from the list
50 | await expect(page.locator("mb-2").nth(0)).toHaveText("item1"); - 2ms - 5021ms

Error: Timed out 5000ms waiting for expect(locator).toHaveText(expected) Remove item from list
Error: Timed out 5000ms waiting for expect(locator).toHaveText(expected)

Locator: locator('mb-2').first()
Expected string: "item1"
Received: <element(s) not found>
Call log:
  - expect.toHaveText with timeout 5000ms
  - waiting for locator('mb-2').first()

at tests/todoapptests.spec.js:50:51
```

TEST RAPORU

5.2.3. HTML Etiket Ekleme : Amaç kullanıcının listeye HTML etiketi ekleme girişiminde uyarı mesajı alıp almadığını test etmektir. Input alanına bir HTML etiketi eklenir ve uyarı mesajının görüntülenip görüntülenmediği kontrol edilir. **Sonuç:** Başarısız.

```
6) [chromium] › todoapptests.spec.js:58:9 › To-Do App Page Tests › Add a HTML tag to the list —
Error: Timed out 5000ms waiting for expect(locator).toHaveText(expected)

Locator: locator('.alert-message')
Expected string: "HTML tagleri girilmesi uygun değildir."
Received: <element(s) not found>
Call log:
  - expect.toHaveText with timeout 5000ms
  - waiting for locator('.alert-message')

70 |         // Expect an alert message
71 |         const alert = page.locator(".alert-message");
> 72 |         await expect(alert).toHaveText("HTML tagleri girilmesi uygun değildir.");
    |                               ^
73 |     });
74 |
75 |

at /home/esra/Downloads/testt/siu/tests/todoapptests.spec.js:72:29

attachment #1: screenshot (image/png)
test-results/todoapptests-To-Do-App-Page-Tests-Add-a-HTML-tag-to-the-list-chromium/test-failed-1.png
```

```
70 |         // Expect an alert message
71 |         const alert = page.locator(".alert-message");
72 |         await expect(alert).toHaveText("HTML tagleri girilmesi uygun değildir."); - 3ms - 5019
Error: Timed out 5000ms waiting for expect(locator).toHaveText(expected) Add a HTML tag to the list
Error: Timed out 5000ms waiting for expect(locator).toHaveText(expected)

Locator: locator('.alert-message')
Expected string: "HTML tagleri girilmesi uygun değildir."
Received: <element(s) not found>
Call log:
  - expect.toHaveText with timeout 5000ms
  - waiting for locator('.alert-message')

at tests/todoapptests.spec.js:72:29
```

TEST RAPORU

5.2.4. Sayfa Yenilendiğinde Giriş Durumunun Korunması : Amaç sayfanın yenilenmesi durumunda kullanıcının giriş yapmış olma durumunun korunup korunmadığını test etmektir. To-do app sayfası yenilenir ve sayfa elemanlarının erişilebilir olup olmadığı kontrol edilir. **Sonuç:** Başarısız.

```
7) [chromium] › todoapptests.spec.js:76:9 › To-Do App Page Tests › Keep logged in after reload the page

Error: Timed out 5000ms waiting for expect(locator).toBeVisible()

Locator: getByPlaceholder('Plan')
Expected: visible
Received: <element(s) not found>
Call log:
  - expect.toBeVisible with timeout 5000ms
  - waiting for getByPlaceholder('Plan')

83 |
84 | // Expect to be in the to-do app page
> 85 | await expect(itemInput).toBeVisible();
    |                                     ^
86 |
87 | await expect(addButton).toBeVisible();
88 | });

at /home/esra/Downloads/testt/slu/tests/todoapptests.spec.js:85:33

attachment #1: screenshot (image/png) -----
test-results/todoapptests-To-Do-App-Pag-6f47d-ed-in-after-reload-the-page-chromium/test-failed-1.png

83 |
84 | // Expect to be in the to-do app page
85 | await expect(itemInput).toBeVisible(); - 4ms - 5025ms
Error: Timed out 5000ms waiting for expect(locator).toBeVisible() Keep logged in after reload the page
Error: Timed out 5000ms waiting for expect(locator).toBeVisible()

Locator: getByPlaceholder('Plan')
Expected: visible
Received: <element(s) not found>
Call log:
  - expect.toBeVisible with timeout 5000ms
  - waiting for getByPlaceholder('Plan')

at tests/todoapptests.spec.js:85:33
```

5.2.5. Boşluk Karakterleriyle Girdi Ekleme : Amaç kullanıcının listeye ardışık boşluk karakterleri içeren bir input girdiğinde bu boşlukların otomatik olarak kırılıp kırılmadığını test etmektir. Input alanına boşluk karakterleri içeren bir girdi girilir, "Add" butonuna tıklanır ve öğenin listeye boşluklar atılmış olarak eklenip eklenmediği kontrol edilir.

Sonuç: Başarılı.

5.2.6. Enter Tuşuyla Eleman Ekleme : Amaç kullanıcının listeye eleman ekleyip "Enter" tuşu ile ekleme yapmadığını test etmektir. Input alanına bir girdi girilir, "Enter" tuşuna tıklanır ve öğenin listeye eklenmediği kontrol edilir.

Sonuç: Başarılı.

6. Geliştirme Önerileri

6.1. Login Page

6.1.1. Kullanıcı parola alanın eklenmesi gerektiği ve kullanıcıların şahsi yapılacaklar listesine kullanıcı adı ve e-mailine sahip yetkisiz kişilerin giriş yapılmasının önüne geçilmesi gerektiği,

TEST RAPORU

6.1.2. Kullanıcının yaptığı başarılı ya da başarısız giriş denemelerinde kullanıcının bilgilendirilmesi için alert ya da benzeri bildirim ifadelerinin gösteriminin yapılabildiği alanların olması gerektiği düşünülmektedir.

6.2. To-Do App Page

6.2.1. Hangi kullanıcının giriş yaptığının dinamik bir şekilde gösterilmesi,

6.2.2. Güncel tarih ve zaman bilgisinin ekranda gösterilmesi,

6.2.3. Liste elemanlarına sıra numarası verilmesi,

6.2.4. Listeye eleman eklenirken liste elemanına;

6.2.4.a. Bitiş tarihi,

6.2.4.b. Kategori seçimi,

6.2.4.c. Öncelik durumu ataması,

6.2.4.ç. Status seçimi (devam ediyor, tamamlandı, iptal edildi vb.),

eklenmesi,

6.2.5. Eklenecek tüm ifadelerin sıralabilir ve filtrelenebilir olması,

6.2.6. Liste elemanlarının düzenlenebilir olması için “-” butonu yanına düzenleme butonu konulması,

6.2.7. Liste elemanlarının pagination özelliğine sahip olması bir sayfada belli sayıda gösterim yapılması ve sayfalar arası geçiş yapılması,

6.2.8. Tüm işlemlerin kaydedilmesi için bir veritabanına sahip olması,

6.2.9. Sayfadan çıkış yapılması için bir logout butonunun bulunması gerektiği,

6.2.10. Kullanıcının yaklaşan etkinliklerinin kullanıcıya bildiriminin yapılabilmesi için bir bildirim mekanizmasının (e-mail, sms veya mobil uygulama ise mobil bildirimleri) kurulması hususlarının mevcut projenin geliştirilmesine olumlu yönde katkı sağlayacağı değerlendirilmektedir.