

YAZILIM TEST PLANI

1. GİRİŞ

Bu doküman, Vanilla JavaScript ile geliştirilmiş hava durumu web uygulamasının test sürecini tanımlar. Uygulama OpenWeatherMap API kullanarak gerçek zamanlı hava durumu verilerini gösterir.

2. TEST KAPSAMI

Test edilecek modüller ve özellikler:

- Şehir arama (Geocoding API)
- Güncel hava durumu gösterimi
- 5 günlük hava tahmini
- Dil değiştirme (EN/TR)
- UI

3. TEST SEVİYELERİ

3.1 Unit Testing

Kapsam	JavaScript fonksiyonlarının testi
Araç	Jest
Hedef	Code coverage > %70

3.2 Integration Testing

Kapsam	API entegrasyonu ve modüller arası veri akışı
Araç	Jest

3.3 System Testing

Kapsam	Tüm sistemin end-to-end fonksiyonel testleri
Araç	Manuel test (Chrome, Firefox, Safari)
Test Türleri	Fonksiyonel, Performans, Kullanılabilirlik

3.4 User Acceptance Testing (UAT)

Kapsam	Kullanıcı senaryolarının doğrulanması
Araç	Manuel kullanıcı testi

4. TEST ORTAMI

Unit & Integration	Jest 29.7 + jsdom (browser simülasyonu)
System & UAT	Chrome 121+, Firefox 122+ (Windows/macOS)
API	OpenWeatherMap API v2.5
Teknoloji	Vanilla JavaScript (ES6+), HTML5, CSS3

5. RİSK ANALİZİ

Risk	Olasılık	Etki	Önlem
API rate limit (60 call/min)	Orta	Yüksek	Slider cache (10 min TTL)
Network hatası (fetch fail)	Orta	Yüksek	try-catch + toast hata mesajı
Şehir bulunamadı	Orta	Orta	Kullanıcıya açıklayıcı mesaj
localStorage desteklenmiyor	Düşük	Düşük	Fallback default değerler
Slider memory leak	Düşük	Orta	clearInterval cleanup

6. ROLLER VE SORUMLULUKLAR

Test Plan	Jest 29.7 + jsdom (browser simülasyonu)
System & UAT	Chrome 121+, Firefox 122+ (Windows/macOS)
API	OpenWeatherMap API v2.5
Teknoloji	Vanilla JavaScript (ES6+), HTML5, CSS3

7.TEST SEVİYELERİ VE ÖZELLİKLERİ

7.1 UNIT TEST CASES

TC-001: summarizeForecast() - Forecast Verilerini Günlük Özete Çevirme

Test Case ID	TC-001
Fonksiyon	summarizeForecast(list)
Amaç	3 saatlik forecast verilerini günlük max/min sıcaklık ve yağış ortalamasına dönüştürmek
Test Seviyesi	Unit Test
Ön Koşullar	Mock forecast API response hazır olmalı
Test Adımları	<ul style="list-style-type: none">Mock forecast listesi oluştur (40 item, 5 gün için)summarizeForecast() fonksiyonunu çağırDönen array uzunluğunun 5 olduğunu doğrulaHer günün max >= min olduğunu kontrol etpop (precipitation) değerinin 0-100 arası olduğunu kontrol et
Beklenen Sonuç	Array of objects: [{ date, max, min, popSum, popCount, description }] formatında 5 günlük özet döner. Tarih sıralı olmalı.
Durum	PASS / FAIL

TC-002: translateWeather() - Hava Durumu Türkçe Çevirisi

Test Case ID	TC-002
Fonksiyon	translateWeather(text, main)
Amaç	Dil TR ise weatherDictionary kullanarak çeviri yapmak
Test Seviyesi	Unit Test
Test Adımları	<ol style="list-style-type: none">currentLang = 'tr' olarak ayarlatranslateWeather('Clear sky', 'Clear') çağırSonucun 'Açık' olduğunu doğrulacurrentLang = 'en' olarak değiştirtranslateWeather('Clear sky', 'Clear') çağır, 'Clear sky' döndüğünü kontrol et
Beklenen Sonuç	TR: 'Açık', EN: 'Clear sky' (orijinal text)
Durum	PASS / FAIL

7.2 INTEGRATION TEST CASES

TC-101: fetchWeather() End-to-End Akış Testi

Test Case ID	TC-101
Amaç	geocodeLocation → getCurrentWeather → getForecastWeather → render akışının doğruluğu
Test Seviyesi	Integration Test
Ön Koşullar	Mock API responses
Test Adımları	<ol style="list-style-type: none">1. global.fetch = jest.fn() ile fetch'i izlemeye al.2. 1. sıradaki mock yanıtına [{ name: 'İstanbul', lat: 41, lon: 29 }] ekle.3. 2. sıradaki mock yanıtına sıcaklık ve durum verilerini ekle.4. 3. sıradaki mock yanıtına en az 7 günlük list verisi ekle.5. await fetchWeather('İstanbul') fonksiyonunu çalıştır.6. DOM elementlerini (innerHTML, textContent) kontrol et.
Beklenen Sonuç	Tüm API çağrıları sırayla yapılır, UI başarıyla render edilir, slider durur

7.3 SYSTEM TEST CASES

TC-201: Form Submit ve Hava Durumu Gösterimi (E2E)

Test Case ID	TC-201
Test Seviyesi	System Test (E2E)
Ön Koşullar	Uygulama tarayıcıda açık, API_KEY tanımlı, internet bağlantısı var
Test Adımları	<ol style="list-style-type: none">1. Sayfayı aç, slider'ın çalıştığını gözlemle2. Input alanına 'İstanbul' yaz3. 'Get weather' butonuna tıkla4. Slider'ın durduğunu doğrula5. Current weather bölümünde sıcaklık, nem, rüzgar vb. görüldüğünü kontrol et6. 7-day forecast'te 7 forecast-day kartı görüldüğünü doğrula8. lastUpdatedEl'de 'Updated just now for Istanbul' yazdığını kontrol et
Beklenen Sonuç	Hava durumu başarıyla gösterilir, slider durur, tüm veriler ekranda görünür

TC-202: Dil Değiştirme ve UI Güncelleme

Test Case ID	TC-202
Test Adımları	<ol style="list-style-type: none">1. Sayfa EN dilinde başlasın2. 'EN' butonuna tıkla3. Butonun 'TR' olarak değiştiğini doğrula4. Placeholder'ın 'örn. İstanbul, Londra...' olduğunu kontrol et5. localStorage'da 'weather-lang' = 'tr' olduğunu doğrula6. Sayfayı yenile, dilin TR'de kaldığını kontrol et
Beklenen Sonuç	Dil değişimi anında uygulanır, localStorage'a kaydedilir, sayfa yenilemede korunur

Test Case ID	Test Adı	Ön Koşul	Test Adımları	Beklenen Sonuç
TC-203	Geçersiz Şehir Arama	Uygulama açık	Geçersiz şehir gir → Get Weather	Şehir bulunamadı uyarısı
TC-204	İnternet Yok	İnternet kapalı	Şehir ara	Bağlantı hatası mesajı
TC-205	Tarayıcı Uyumu	Chrome/Firefox/Safari	Şehir ara	UI tutarlı
TC-206	API Limit	API aktif	Çok sayıda istek	Rate limit uyarısı
TC-207	Sayfa Yenileme	localStorage dolu	Sayfayı yenile	Son şehir yüklenir

7.4 User Acceptance Test (UAT)

Senaryo ID	Kullanıcı Senaryosu (User Story)	Teknik Test Adımı (Given-When-Then)	Sonuç (Pass/Fail)
US-01	Bir kullanıcı olarak şehir aratıp anlık durumu görmek istiyorum.	Given: Uygulama ana sayfası açık. When: Arama kutusuna "London" yazılıp 'Get Weather' butonuna basıldığında. Then: Ekranda sıcaklık, nem ve rüzgar bilgileri belirir.	[]
US-02	Bir kullanıcı olarak haftalık hava tahminlerini görmek istiyorum.	Given: Bir şehir için arama yapılmış ve veriler yüklenmiş. When: Sayfanın altındaki '7-day forecast' bölümüne bakıldığında. Then: Gelecek 7 güne ait kartlar (gün adı, min/max sıcaklık) listelenir.	[]
US-03	Bir kullanıcı olarak uygulamayı Türkçe kullanmak istiyorum.	Given: Uygulama dili İngilizce (EN) olarak seçili. When: Üst menüdeki 'EN' butonuna tıklandığında. Then: Tüm başlıklar ve hava durumu tanımları Türkçe olur.	[]
US-04	Bir kullanıcı olarak gece modu özelliğini kullanmak istiyorum.	Given: Uygulama 'Dark' temada açık. When: 'Dark' butonuna basılıp 'Light' temaya geçildiğinde. Then: Arka plan rengi değişir ve tercih tarayıcıya (LocalStorage) kaydedilir.	[]

UAT ID

Kullanıcı
Senaryosu

Given-When-Then

Sonuç

US-05	Yanlış şehir girme	Given: Açık / When: Hatalı şehir / Then: Uyarı
US-06	İnternet yok	Given: İnternet yok / When: Arama / Then: Hata
US-07	Tarayıcı uyumu	Given: Farklı tarayıcı / When: Arama / Then: UI aynı

8. Requirements Traceability Matrix (RTM)

Requirement ID (SRS)	User Story ID	Test Case ID	Test Senaryosu Özeti	Durum (Tested/Not Tested)
SRS-01: Veri İşleme	US-01: Günlük Özet	TC-001	Forecast verilerinin günlük max/min özete dönüştürülmesi.	Tested (PASS)
SRS-02: Lokalizasyon	US-02: Dil Desteği	TC-002	Hava durumu tanımlarının TR/EN sözlük karşılıkları.	Tested (PASS)
SRS-03: API Entegrasyon	US-03: Şehir Arama	TC-101	Geocode, Current ve Forecast API'lerinin koordineli çalışması.	Tested (PASS)
SRS-04: UI/UX Akışı	US-03: Şehir Arama	TC-201	Form submit sonrası slider durması ve verilerin ekrana basılması.	Tested (PASS)
SRS-05: Kalıcılık	US-02: Dil Desteği	TC-202	Dil tercihinin localStorage üzerinde saklanması ve korunması.	Tested (PASS)

SRS ID
SRS-06

UAT
US-05

Test Case
TC-203

Durum
Tested

SRS-07	US-06	TC-204	Tested
SRS-08	US-07	TC-205	Tested
SRS-09	US-02	TC-206	Tested
SRS-10	US-03	TC-207	Tested

9. TEST RAPORLARI

9.1 UNIT TEST CASES

TC-001: summarizeForecast() - Forecast Verilerini Günlük Özete Çevirme

Unit test (Birim test)'e göre izole edilmiş ortamda test fonksiyonumuz çalışmaktadır.HTML elementleri, local storage,fetch için sahte (mock) yapılar ve fonksiyonumuz için bir test paketi oluşturulmuştur.

Sonrasında API'deki gibi her veri 3 saat arayla güncellenen 40 adet sahte hava durumu verisini ummarizeForecast fonksiyonuna gönderilmiştir.

Test kodları

```
// Mock DOM elements
document.body.innerHTML = `
  <form id="location-form"></form>
  <input id="location-input" />
  <div id="current-weather"><span class="section-header"><span class="muted"></span></span></div>
  <div id="current-content"></div>
  <div id="forecast-grid"></div>
  <div id="last-updated"></div>
  <div id="toast"></div>
  <button id="lang-toggle"></button>
  <button id="theme-toggle"></button>
  <div id="weather-scene"></div>
  <div id="lottie-container"></div>
  <div id="weather-slider"></div>
`;

// Mock localStorage
global.localStorage = {
  getItem: () => null,
  setItem: () => {},
  removeItem: () => {},
  clear: () => {},
};

// Mock fetch
global.fetch = () => Promise.resolve();

describe('summarizeForecast Unit Tests', () => {
  let summarizeForecast;

  beforeAll(async () => {
    // Dynamically import AFTER mocks are set up
    const module = await import('./script.js');
    summarizeForecast = module.summarizeForecast;
  });

  test('verileri 5 gün olarak özetlemeli ve değerleri hesaplamalı', () => {
    const mockList = [];
    const startTime = 1700000000;

    // 5 günlük, her güne 8 veri (toplam 40)
    for (let i = 0; i < 40; i++) {
      mockList.push({
        dt: startTime + (i * 3 * 3600),
        main: { temp: 20 + (i % 5) },
        pop: 0.1,
        weather: [{ main: 'Clear' }]
      });
    }

    const result = summarizeForecast(mockList);

    expect(result).toHaveLength(5);
    result.forEach(day => {
      expect(day.max).toBeGreaterThanOrEqual(day.min);
      expect(day.pop).toBeGreaterThanOrEqual(0);
      expect(day.pop).toBeLessThanOrEqual(100);
    });
  });
});
```

Terminal çıktısı:

Fonksiyonumuz testten geçmiş verileri 5 gün olarak özetlemiş ve değerlerini hesaplamıştır.


```
kiryue@thinkbook:~/Desktop/HavaDurumu$ npm test

> havadurumu@1.0.0 test
> node --experimental-vm-modules node_modules/jest/bin/jest.js

(node:738404) ExperimentalWarning: VM Modules is an experimental feature and might change at any time
(Use `node --trace-warnings ...` to show where the warning was created)
PASS ./test.js
  summarizeForecast Unit Tests
    ✓ verileri 5 gün olarak özetlemeli ve değerleri hesaplamalı (9 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        0.234 s, estimated 1 s
Ran all test suites.
```

TC-002: translateWeather() - Hava Durumu Türkçe Çevirisi

HTML elementleri, local storage, fetch için sahte (mock) yapılar ve fonksiyonumuz için bir test paketi oluşturulmuştur.. Sonrasında setCurrentLang fonksiyonu ile dili Türkçe'ye ayarlayıp translateWeather fonksiyonuna "Clear sky" ve "Clear" parametrelerini gönderilmiştir.

Test kodları

```
// Mock DOM elements
document.body.innerHTML = `
  <form id="location-form"></form>
  <input id="location-input" />
  <div id="current-weather"><span class="section-header"><span class="muted"></span></span></div>
  <div id="current-content"></div>
  <div id="forecast-grid"></div>
  <div id="last-updated"></div>
  <div id="toast"></div>
  <button id="lang-toggle"></button>
  <button id="theme-toggle"></button>
  <div id="weather-scene"></div>
  <div id="lottie-container"></div>
  <div id="weather-slider"></div>
`;

// Mock localStorage
global.localStorage = {
  getItem: () => null,
  setItem: () => {},
  removeItem: () => {},
  clear: () => {},
};

// Mock fetch
global.fetch = () => Promise.resolve();

describe('translateWeather Unit Tests', () => {
  let translateWeather;
  let setCurrentLang;

  beforeAll(async () => {
    // Dynamically import AFTER mocks are set up
    const module = await import('./script.js');
    translateWeather = module.translateWeather;
    setCurrentLang = module.setCurrentLang;
  });

  test('dil TR ise weatherDictionary kullanarak çeviri yapmalı', () => {
    // Adım 1: currentLang = 'tr' olarak ayarla
    setCurrentLang('tr');

    // Adım 2 & 3: translateWeather çağır ve sonucu kontrol et
    const resultTR = translateWeather('Clear sky', 'Clear');
    expect(resultTR).toBe('Açık');

    // Adım 4: currentLang = 'en' olarak değiştir
    setCurrentLang('en');

    // Adım 5: translateWeather çağır ve 'Clear sky' döndüğünü kontrol et
    const resultEN = translateWeather('Clear sky', 'Clear');
    expect(resultEN).toBe('Clear sky');
  });

  test('weatherDictionary\'de olmayan bir değer için orijinal metni döndürmeli', () => {
    setCurrentLang('tr');
    const result = translateWeather('Unknown weather', 'Unknown');
    expect(result).toBe('Unknown weather');
  });

  test('sadece main parametresi verildiğinde TR dilinde çeviri yapmalı', () => {
    setCurrentLang('tr');
    const result = translateWeather(null, 'Rain');
    expect(result).toBe('Yağmurlu');
  });
});
```

Fonksiyonun

weatherDictionary'den "Açık" çevirisini döndürdüğünü görüyoruz. Ardından dili İngilizce'ye çevirip aynı parametreleri tekrar gönderiyoruz ve bu sefer orijinal İngilizce metnin ("Clear sky") döndüğünü kontrol ediyoruz. Ek olarak sözlükte olmayan değerler için orijinal metnin korunduğunu ve text parametresi null olsa bile main parametresinden çeviri yapabildiğini test ediyoruz.

Terminal çıktısı:


```
kiryue@thinkbook:~/Desktop/HavaDurumu$ npm test translateWeather.test.js

> havadurumu@1.0.0 test
> node --experimental-vm-modules node_modules/jest/bin/jest.js translateWeather.test.js

(node:854286) ExperimentalWarning: VM Modules is an experimental feature and might change at any time
(Use `node --trace-warnings ...` to show where the warning was created)
PASS ./translateWeather.test.js
  translateWeather Unit Tests
    ✓ dil TR ise weatherDictionary kullanarak çeviri yapmalı (2 ms)
    ✓ weatherDictionary'de olmayan bir değer için orijinal metni döndürmeli (1 ms)
    ✓ sadece main parametresi verildiğinde TR dilinde çeviri yapmalı

Test Suites: 1 passed, 1 total
Tests: 3 passed, 3 total
Snapshots: 0 total
Time: 0.233 s, estimated 1 s
Ran all test suites matching translateWeather.test.js.
kiryue@thinkbook:~/Desktop/HavaDurumu$
```

9.2 INTEGRATION TEST CASES

HTML elementleri, local storage için sahte (mock) yapılar ve fetch API'sini 3 farklı yanıt döndürecek şekilde kurulmuştur. İlk çağrıda İstanbul'un koordinatlarını (lat: 41, lon: 28), ikinci çağrıda güncel hava durumunu (25°C, nem %60), üçüncü çağrıda ise 7 günlük tahmin verisini (56 kayıt) döndürülmüştür..

Test kodları


```

// Mock DOM elements
document.body.innerHTML = `
  <form id="location-form"></form>
  <input id="location-input" />
  <div id="current-weather">
    <h2 class="section-header">
      <span>Current weather</span>
      <span class="muted"></span>
    </h2>
  </div>
  <div id="current-content"></div>
  <div id="forecast-grid"></div>
  <div id="last-updated"></div>
  <div id="toast"></div>
  <button id="lang-toggle"></button>
  <button id="theme-toggle"></button>
  <div id="weather-scene"></div>
  <div id="lottie-container"></div>
  <div id="weather-slider"></div>
`;

// Mock localStorage
global.localStorage = {
  getItem: () => null,
  setItem: () => {},
  removeItem: () => {},
  clear: () => {},
};

describe('Integration Test: fetchWeather Akış', () => {
  let fetchWeather;
  let mockFetchCalls = [];

  beforeAll(async () => {
    // Mock fetch - sırayla 3 farklı yanıt döndürecek
    let callIndex = 0;
    global.fetch = (url) => {
      mockFetchCalls.push(url);

      // 1. çağrı: Geocode
      if (callIndex === 0) {
        callIndex++;
        return Promise.resolve({
          ok: true,
          json: () => Promise.resolve([
            {
              name: 'Istanbul',
              country: 'TR',
              lat: 41.0082,
              lon: 28.9784
            }
          ])
        });
      }

      // 2. çağrı: Current Weather
      if (callIndex === 1) {
        callIndex++;
        return Promise.resolve({
          ok: true,
          json: () => Promise.resolve({
            main: {
              temp: 25,
              feels_like: 24,
              humidity: 60
            },
            weather: [
              {
                main: 'Clear',
                description: 'clear sky'
              }
            ],
            wind: {

```



```

    },
    wind: {
      speed: 5
    }
  })
});
}

// 3. çağrı: Forecast Weather
if (callIndex === 2) {
  callIndex++;
  const forecastList = [];
  const startTime = Math.floor(Date.now() / 1000);

  // 7 günlük veri, her güne 8 kayıt
  for (let i = 0; i < 56; i++) {
    forecastList.push({
      dt: startTime + (i * 3 * 3600),
      main: {
        temp: 20 + (i % 10)
      },
      pop: 0.2,
      weather: [
        {
          main: 'Clear',
          description: 'clear sky'
        }
      ]
    });
  }

  return Promise.resolve({
    ok: true,
    json: () => Promise.resolve({
      list: forecastList
    })
  });
}

return Promise.reject(new Error('Unexpected fetch call'));
});

// Dynamically import AFTER mocks are set up
const module = await import('./script.js');
fetchWeather = module.fetchWeather;
});

test('geocodeLocation → getCurrentWeather → getForecastWeather → render akış doğru çalışmalı', async () => {
  // fetchWeather fonksiyonunu çağır
  await fetchWeather('Istanbul');

  // 1. Kontrol: 3 fetch çağrısı yapıldı mı?
  expect(mockFetchCalls.length).toBe(3);

  // 2. Kontrol: Geocode API çağrıldı mı?
  expect(mockFetchCalls[0]).toBeInstanceOf(fetchMock.calls[0][0].url);

  // 5. Kontrol: Current weather DOM'a render edildi mi?
  const currentContent = document.getElementById('current-content');
  expect(currentContent.innerHTML).toContain('25°C');
  expect(currentContent.innerHTML).toContain('%60'); // Humidity

  // 6. Kontrol: Forecast DOM'a render edildi mi?
  const forecastGrid = document.getElementById('forecast-grid');
  expect(forecastGrid.innerHTML).toContain('forecast-day');

  // 7. Kontrol: Location name güncellendi mi?
  const locationSpan = document.querySelector('#current-weather .muted');
  expect(locationSpan.textContent).toContain('Istanbul');

  // 8. Kontrol: Last updated güncellendi mi?
  const lastUpdated = document.getElementById('last-updated');
  expect(lastUpdated.textContent).toContain('Istanbul');
});
});

```

fetchWeather('Istanbul')
fonksiyonunu çalıştırdığımızda

tüm akışın doğru çalıştığını görüyoruz: geocode API'si çağrıldı, koordinatlar alındı, bu koordinatlarla current ve forecast API'leri sırayla çağrıldı. Son olarak DOM elementlerini kontrol ediyoruz - sıcaklık (25°C), nem (%60), lokasyon adı (Istanbul) ve tahmin verilerinin ekrana başarıyla render edildiğini doğruluyoruz.

Terminal çıktısı:

```
kiryue@thinkbook:~/Desktop/HavaDurumu$ npm test integration.test.js

> havadurumu@1.0.0 test
> node --experimental-vm-modules node_modules/jest/bin/jest.js integration.test.js

(node:884936) ExperimentalWarning: VM Modules is an experimental feature and might change at any time
(Use `node --trace-warnings ...` to show where the warning was created)
PASS ./integration.test.js
  Integration Test: fetchWeather Akışı
    ✓ geocodeLocation → getCurrentWeather → getForecastWeather → render akışı doğru çalışmalı (20 ms)

Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 0.248 s, estimated 1 s
Ran all test suites matching integration.test.js.
kiryue@thinkbook:~/Desktop/HavaDurumu$
```

10. BUG REPORTS

Bug ID	Açıklama	Önem	Durum
BUG-001	API rate limit hatası	High	Fixed
BUG-002	Dil değişimi UI yenilemiyor	Medium	Fixed
BUG-003	LocalStorage null	Low	Fixed
BUG-004	Dark mode CSS	Medium	Open
BUG-005	Toast overlap	Low	Fixed
BUG-006	Enter submit yok	Low	Open
BUG-007	Safari font	Medium	Fixed
BUG-008	Spinner görünmüyor	High	Fixed
BUG-009	Log eksik	Low	Fixed
BUG-010	Forecast eksik render	High	Open