

127.0.0.1	POST	"/users/login"
127.0.0.1	POST	"/users/login"
127.0.0.1	POST	"/users/register"
127.0.0.1	GET	"/sahalar/"

Go ile Backend ve Java ile Test Otomasyonu

Bu sunumda, Go dilinde geliştirilmiş bir halı saha rezervasyon sisteminin Java tabanlı Rest-Assured kütüphanesiyle nasıl otomatik test edildiğini ve yapay zekanın test süreçlerindeki dönüştürücü rolünü inceleyeceğiz.

```
String requestBody = "{" +  
    "\"name\": \"mehmet\", " +  
    "\"email\": \"mehmet_v2\" + System.currentTimeMillis() + \"@test.com\", " +  
    "\"password\": \"123456\" " +  
    "}";  
  
given()  
    .header(headerName: "Content-Type", headerValue: "application/json")  
    .body(requestBody)  
    .when()
```

Vizyonumuz: Yapay Zeka Destekli Test Otomasyonu



Hata Teşhisini Minimize Etme

Test süreçlerinde yapay zeka kullanımıyla hataların teşhis sürelerini önemli ölçüde azaltmayı hedefliyoruz.



Sistem Güvenilirliğini Artırma

AI entegrasyonu sayesinde sistemin güvenilirliğini ve kararlılığını en üst düzeye çıkarmayı amaçlıyoruz.



Test Verimliliğini Yükseltme

Otomasyon ve AI ile manuel çabayı minimize ederek test döngülerini hızlandırıyoruz.

Kritik Test Senaryoları

Sistemimizin temel işlevsellliğini doğrulamak için çeşitli test senaryoları uyguladık.



1

Kullanıcı Kayıt (POST /users/register)

Yeni kullanıcı kaydının başarılı olup olmadığını ve yanıt olarak beklenen **HTTP 201 Created** status kodu ile doğrulama mesajının gelip gelmediğini kontrol ettik.



2

Kullanıcı Giriş (POST /users/login)

Sisteme başarıyla giriş yapılmış olup olmadığını ve kullanıcının oturum açma (authentication) token'ının doğru şekilde alındığını test ettik.



3

Halı Saha İşlemleri (GET /sahalar/)

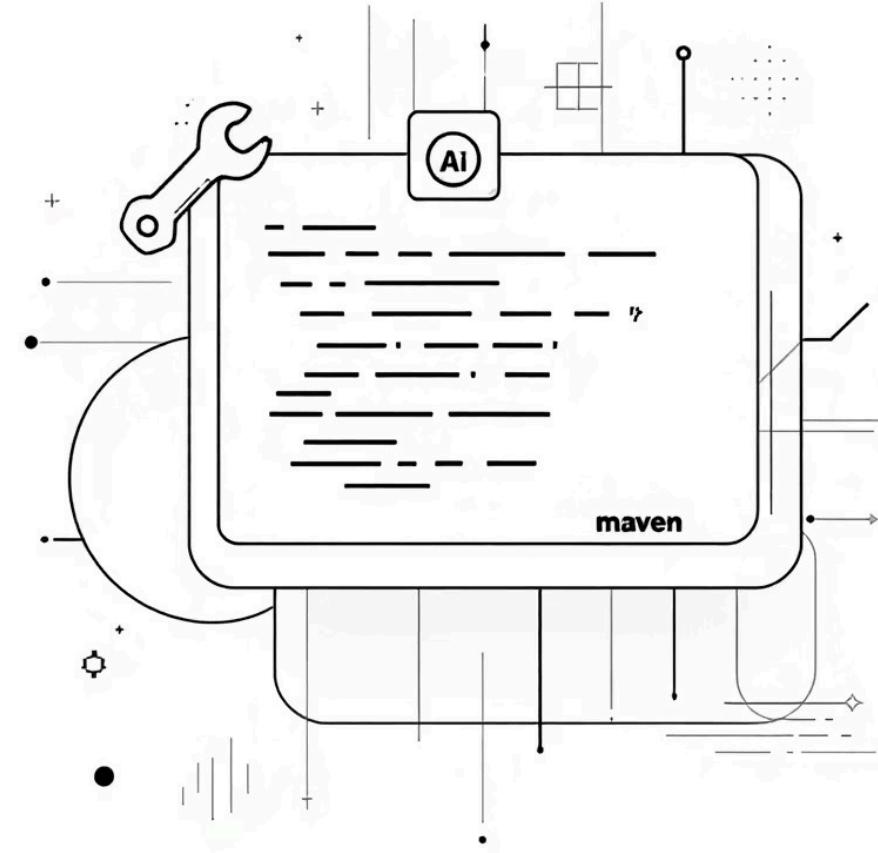
Mevcut halı sahaların listesinin doğru ve eksiksiz olarak dönüp dönmediğini, veri bütünlüğünü ve formatını doğruladık.

The screenshot shows a test results interface with a sidebar containing a tree view of test cases and a main area displaying a list of test runs with their details. The sidebar includes icons for 'Test run at 12/18/2025, 10:53:29 PM' (with 29 older results), 'Test run at 12/18/2025, 10:53:25 PM', 'Test run at 12/18/2025, 10:53:18 PM', 'Test run at 12/18/2025, 10:41:22 PM', and 'Test run at 12/18/2025, 10:40:45 PM'. Each run is expanded to show individual test cases like 'AppTest', 'kayitTesti()', 'loginBasariliTesti()', 'rezervasyonCakismaTesti()', 'rezervasyonOlusturmaTesti()', and 'rezervasyonCakismaTesti()' with their respective status icons (green checkmark or yellow warning). The bottom right corner of the screenshot has a watermark that says 'Made with GAMMA'.

AI ile Çözülen Temel Sorunlar: Maven ve Yollar

Maven Path ve Kütüphane Çakışmaları

Yapay zeka, Maven bağımlılıkları ve build path sorunlarını anında teşhis edip, **mvnw (Maven Wrapper)** yapılandırmasını optimize etmemize yardımcı oldu. Bu, kurulum ve geliştirme süremizi önemli ölçüde kısalttı.





Elde Edilen Kazanımlar: Verimlilik ve Hız

Bu projedeki otomasyon ve AI entegrasyonu sayesinde önemli verimlilik artışları sağladık.

Regresyon Testlerinde Hız

Manuel testlerle saatler sürecek regresyon testleri, otomatik testler sayesinde **saniyelere** indirildi.

Geliştirme Döngüsü Hızlanması

Test süreçlerinin kısalması, yeni özelliklerin daha hızlı devreye alınmasına olanak tanıdı.

Hata Yakalama Oranında Artış

Otomatik testler, manuel testlerde gözden kaçabilecek birçok hatayı erken aşamada yakaladı.

Gelecek Vizyonu: Test Otomasyonunda Yeni Nesil

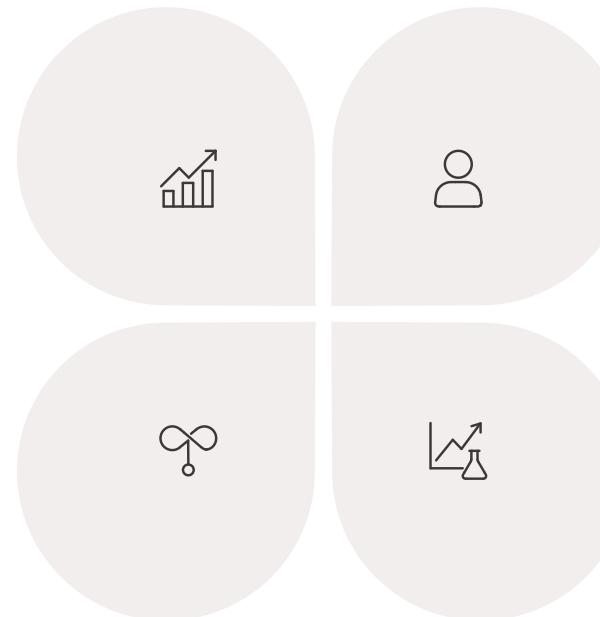
Yapay zekanın test süreçlerindeki potansiyeli henüz başlangıç aşamasında.

Sentetik Veri Üretimi

AI'nın test verilerini otomatik olarak üretmesi (Synthetic Data Generation) ile test kapsamını genişletme.

Sürekli Öğrenme

Test sisteminin, her çalıştırıldan sonra öğrenerek daha akıllı ve verimli hale gelmesi.



Kendi Kendini Düzeltten Testler

Test senaryolarının, küçük değişikliklerde otomatik olarak adapte olması ve kendini güncellemesi.

Tahmine Dayalı Hata Tespiti

AI ile potansiyel hataların ortaya çıkmadan önce tahmin edilmesi ve önleyici tedbirler alınması.