

ArtemisAI

Yapay Zeka Destekli Kişisel Eğitim Koçunuz

Statik planların ötesinde, öğrenme yolculuğunuza adapte olan
dinamik bir sistem.



Her Öğrencinin Tanıdık Hikayesi: Planlama Kaosu

Sınav haftası yaklaşıyor. Notlar dağınık, konular birikmiş ve nereden başlayacağınızı bilemiyorsunuz. Hazırladığınız çalışma planları genellikle ilk birkaç günden sonra hayatın getirdiği beklenmedik durumlar karşısında geçerliliğini yitirir.



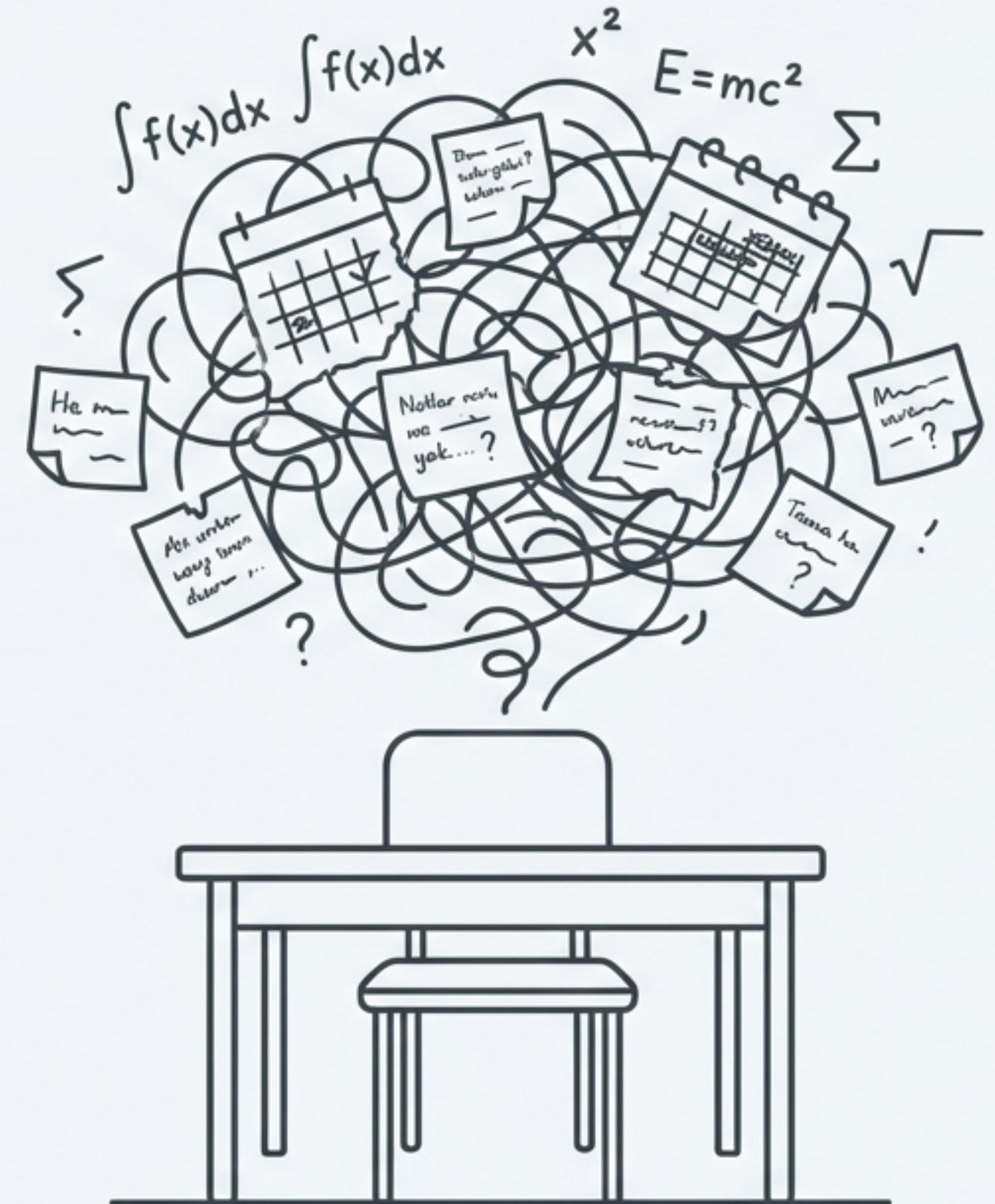
Motivasyon Kaybı: Plana uyamayınca hissedilen hayal kırıklığı.



Statik Planlar: Gerçek hayata uyum sağlayamayan, katı programlar.



Belirsizlik: Hangi konuya ne kadar zaman ayırmak gerektiğini kestirememem.



Tanışın: ArtemisAI - Sadece Bir Planlayıcı Değil, Bir Koç

The screenshot shows the ArtemisAI software interface. At the top, there is a navigation bar with the ArtemisAI logo, a search bar, and buttons for "Kar Yeğdir" (Green Card) and "Nasıl Kullanılır" (How to Use). Below the navigation bar, there is a toolbar with icons for "Plan Oluştur" (Create Plan), "Geri Bildirim", "Plan Gözontöleme", "Takvim", "İstatistikler", and "Agent Analizi". The main area of the interface is titled "Haftalık Çalışma Planı Oluştur" (Create Weekly Work Plan). It includes fields for "Şenlik Adı" (Event Name) set to "Mantık deneyleri", "Şenlik Tarihi" (Event Date) set to "2026/03/15", and "Günlük Həkimişin Çalışma Saati" (Daily Supervisor's Working Hours) represented by a horizontal timeline from 08:00 to 17:00. Below this, there is a section titled "Çalışma Günlerini Seçin" (Select Working Days) with a note: "Hangi günlerde çalışmak istiyorsunuz? Amma hər gün bir gün çalışma zamanınız olsun." (Which days do you want to work? But each day should have a working time). A list of days is shown with checkboxes: Həftəyə, Pazar, Salı, Çarşamba, Perşembe, Cuma, Cumartesi, and Pazar. A callout box in the bottom right corner contains the text: "ArtemisAI, öğrencinin ihtiyaçlarını anlayan, geri bildirimlerle kendini güncelleyen ve kişiselleştirilmiş bir yol haritaşı sunan yapay zeka destekli bir eğitim koçudur. Amacı, öğrenme sürecini yönetilebilir, motive edici ve verimli kılmaktır."

ArtemisAI, öğrencinin ihtiyaçlarını anlayan, geri bildirimlerle kendini güncelleyen ve kişiselleştirilmiş bir yol haritaşı sunan yapay zeka destekli bir eğitim koçudur. Amacı, öğrenme sürecini yönetilebilir, motive edici ve verimli kılmaktır.

Adım 1: Yolculuk Başlıyor - Kişisel Planınızı Oluşturun

Her şey, hedeflerinizi belirlemekle başlar. ArtemisAI'a dersinizi, sınav tarihinizi ve size uygun çalışma günlerini söylemeniz yeterlidir. Yapay zeka, bu temel bilgilerle size özel bir program oluşturur.

1. Ders Adı: 'Mantık Devreleri' gibi ders konularını girin.

2. Sınav Tarihi: Hedefinizi netleştirin.

3. Çalışma Günleri: Haftalık ritminizi seçin (Örn: Pazartesi, Salı, Çarşamba, Perşembe, Cuma).

ArtemisAI - Yapay Zeka Destekli Eğitim Koçu

LLN tabanlı kişiselleştirilmiş çalışma planı sistemi

Plan Oluştur Geri Bildirim Plan Görüntüleme Takvim İstatistikler Agent Analizi

Haftalık Çalışma Planı Oluştur

Ders Adı: Mantık Devreleri

Sınav Tarihi: 2026/01/15

Günlük Maksimum Çağrısa Saati: 4

Çalışma Günlerini Seçin

Hangi günlerde çalışmak istediğiniz seçin. Ana tegge ter tüm günleri açılay/kapabır.

Haftaçı

Pazartesi

Salı

Çarşamba

Perşembe

Cuma

Seçilen günler: Pazartesi, Salı, Perşembe, Cuma, Pazar (5 gün)

Plani Oluştur

Adım 2: Yol Haritanız Hazır - Konular ve Zaman Çizelgesi

ArtemisAI, dersin tüm konularını analiz ederek mantıksal bir sıraya koyar ve her birine tahmini çalışma süreleri atar. Artık önünüzde tamamlanması gereken görevlerden oluşan net bir liste var.

The screenshot shows a digital notepad interface with a green header bar. The header bar contains a circular icon with a checkmark and the text "İlerleme: 4/15 konu (%27)". On the right side of the header bar, there is a "Deploy" button with a gear icon. Below the header bar is a green progress bar with a white circular indicator at approximately 27% completion. The main area of the notepad is a table with the following columns: "Tümülemediğiniz konuların listesi", "Tarih", "Konu", and "Saat". The table lists 15 items, each with a checkbox and a due date. Some items have a red checkmark, indicating they are completed or marked as such. The last item has an empty checkbox. The table is divided by a vertical green line.

Tümülemediğiniz konuların listesi	Tarih	Konu	Saat
✓	2026-01-05	Sayı Sistemleri ve Dönüşümleri	2s
✓	2026-01-05	Beeleme Cekii Temeliler ve Teorsmeler	2s
✓	2026-01-05	Temel ve Türeblili Nerbk Xspiler (AJO, OR, KOT, KHO, NOR, 008)	1.5s
✓	2026-01-05	Karnaugh Haritslan [K-Map] Iis Ninimizasyen (2 ve 3 degiçenli)	2.5s
□	2026-01-09	K-Map ile Nidimicsasyen (4 degiçenli) ve Don't Care Koullan	1.5s
□	2026-01-09	Temel Kombinsayoncl Devre Tasnms (Toplayicilar, Çkanclar)	2.3s
□	2026-01-11	Kediyicilar (Encoder), Kod (2zlidicler (Decoder) ve Uyguemsler)	2s
□	2026-01-11	Temel Flip (HUX) ve Demultiplexers (DEMUX) (Uitgma Tebidi)	2s
□	2026-01-12	Ardigl Devrs Analisi: Burum Diyagramları ve Durum Tablosu	1.5s
□	2026-01-13	Sayıller (Asenkron ve Senkron Tasanm ve Ansılıcl	2s

Üst kısımdaki ilerleme çubuğu, ne kadar yol katettiğınızı ve ne kadar kaldığını anlık olarak göstererek motivasyonu artırır.

Adım 3: Planı Hayata Geçirmek - Takvim Entegrasyonu

Görev listeleri soyut kalabilir. ArtemisAI, oluşturulan çalışma planını doğrudan bir takvime entegre eder. Bu sayede, her gün hangi konuya odaklanmanız gerektiğini net bir şekilde görebilirsiniz.

The screenshot shows a digital calendar interface for January 2026. The days of the week are labeled from Sunday to Saturday. Each day's grid contains several colored boxes, each representing a task or event. A legend at the bottom left, titled 'Ders Renk Anahtarı' (Subject Color Key), maps colors to subjects: blue for Mobil Dersleri (Mobile Classes), red for Sunucu İletişim Sistemleri (Server Communication Systems), and green for Silgileyecek Mühendislikle Giriş (Access Engineering). The tasks are described in Turkish, such as 'Mobil Derslerinizi sunucusuna yükleyin' (Upload mobile classes to the server) and 'Silgileyecek Mühendislikle Giriş' (Access Engineering through the gateway). The interface includes navigation buttons for months and years, and tabs for 'Uzaktan Derslerim', 'Sunucu İletişim C...', and 'Bağlantı Mühend...'. A 'Deploy' button is visible in the top right corner.



Takvim, birden fazla ders için farklı renk kodlarıyla ('Ders Renk Anahtarı') planları aynı anda yönetme esnekliği sunar.

Dönüm Noktası: Yapay Zeka Koçunuzla Konuşun

ArtemisAI'dan yardım istemek, bir arkadaşınıza mesaj atmak kadar kolaydır. Öğrenci, yaşadığı zorluğu ve hislerini kendi kelimeleriyle ifade eder.

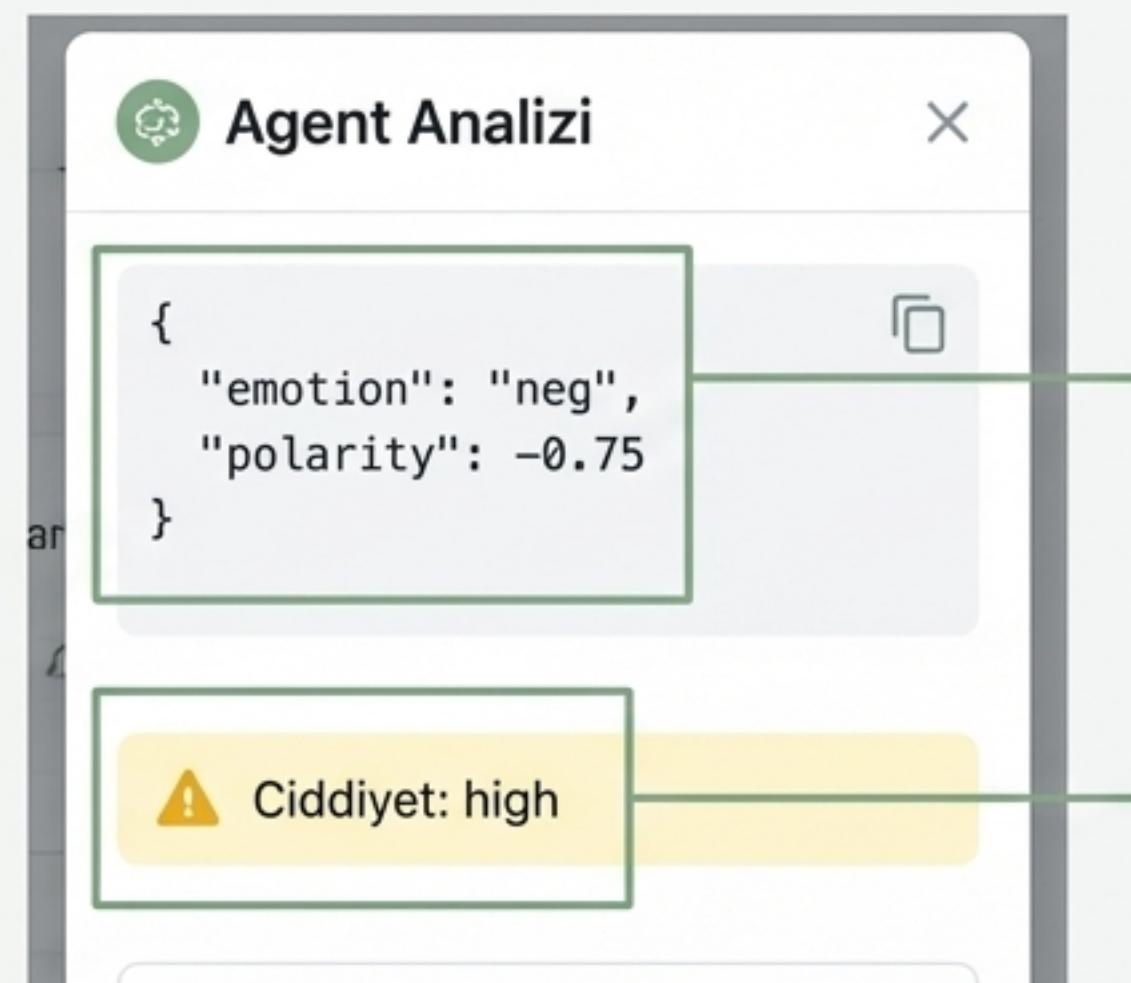


Bu hafta çok verimsizdi plana sadık kalamadım ve derslerimden geri kaldım planımı arttı

A screenshot of the ArtemisAI platform interface. The top navigation bar includes 'ArtemisAI - Yapay Zeka Destekli Eğitim Koçu', 'LÜH Dilekçe Tepki ve İstek Listesi', and two green buttons: 'Evet Onayla' and 'Evet Bırakla'. Below the navigation is a toolbar with icons for 'Kontrol', 'Sorular', 'Favoriler', 'İstek', 'Müsteşarlar', and 'Ağrıtlı İstek'. A red box highlights the 'Kontrol' button. The main area shows a message from the AI tutor: 'Bu hafta çok verimsizdi plana sadık kalamadım ve derslerimden geri kaldım planımı arttı'. At the bottom, there's a note: 'Bu hizmetin verimliliği geleneksel iletişimdeki ve derslerinden geri kaldım plannım arttı' and a 'Evet Gündemle Gänder' button.

Perde Arkası: Yapay Zeka Geri Bildirimi Nasıl Anlıyor?

ArtemisAI, bu basit cümleyi derinlemesine analiz eder:



- 1. Duygu Analizi:** Metnin arkasındaki duyguyu tespit eder.
Sonuç: Negatif bir duyu durumu.
- 2. Ciddiyet Tespiti:** Durumun aciliyetini ve önemini anlar.

- 3. Niyet Çıkarımı:** Kullanıcının net talebini yorumlar ('planımı arttı' açıkça talep etmektedir).

Çözüm: Statik Plana Karşı Akıllı Revizyon

Anlayış, eyleme dönüşür. Yapay zeka, durumu değerlendirerek öğrencinin açığı kapatmasına yardımcı olacak akıllı bir öneri sunar.



Karar:

Planın yoğunluğunu artır
(Koordinatör Kararı: inç *(çarpan:
1.2)).



Öneri:

Seçili dersler için çalışma
yöğunluğunu %20 oranında artırarak
yeni ve ulaşılabilir bir plan oluştur.

© Plan Değişikliği Önerisi

Koordinatör, seçili dersler (Mantık Devreleri, Sunucu İşletim Sistemleri) için yoğunluğu %120 oranına çekmeyi öneriyor.

Değişikliği Uygula Mevcut Planı Koru

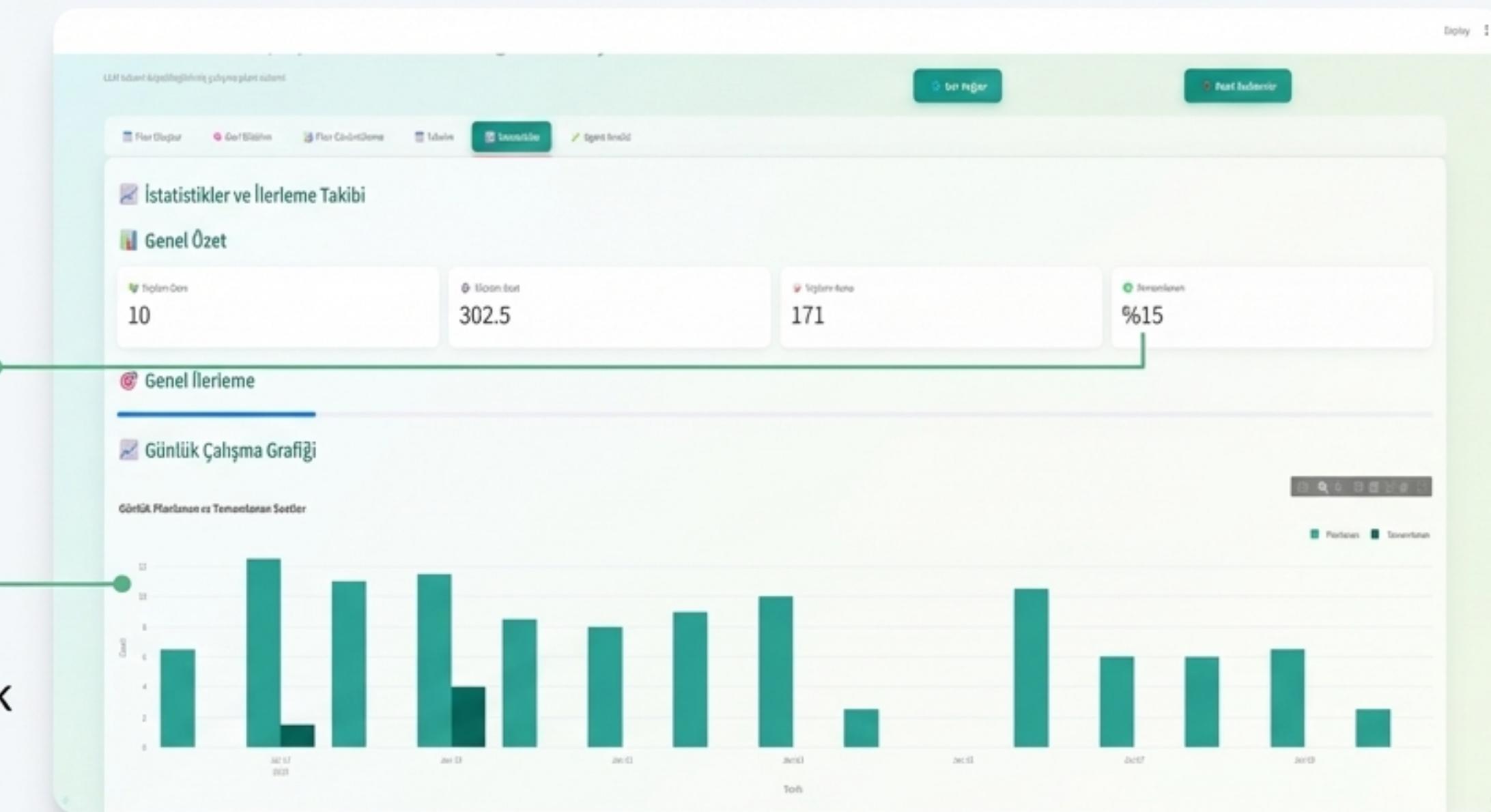
Öğrenciye 'Değişikliği Uygula' veya
'Mevcut Planı Koru' seçenekleri sunulur,
kontrol daima kullanıcıdır.

Sonuç: Verilerle İlerlemeyi ve Motivasyonu Geri Kazanmak

Yeni, revize edilmiş planla birlikte öğrenci tekrar yoluna koyulur. İstatistikler paneli, ilerlemeyi somut verilerle takip etme imkanı sunar.



Genel Özeti: Toplam ders, saat ve konu sayısı.



Tamamlanma Oranı: Hedefe ne kadar yaklaşıldığını gösteren yüzde (%15).



Günlük Çalışma Grafiği: Planlanan ve tamamlanan çalışma saatlerini karşılaştırarak öğrencinin performansını ve tutarlılığını görselleştirir.

Koçunuz Her Adımda Yanınızda

ArtemisAI sadece bir araç değil, aynı zamanda motive edici bir yol arkadaşıdır. Sistem, analizi tamamladıktan sonra öğrenciye kişisel bir cesaretlendirme mesajı gönderir.

The screenshot shows a digital notebook interface with a light blue header containing the text "Koçunuz Her Adımda Yanınızda". Below the header is a main content area featuring a large green celebratory message:

“Harika! Analiz tamamlandı. 😊

Şimdi o enerjiyi masaya koyma zamanı.

adım adım ilerleyeceğiz, sana güveniyorum.

Hadi başlayalım!”

On the left side of the main content, there is a sidebar titled "Agent Analizi" with the sub-section "Analiz edilecek ders seçin". It lists "Algoritma ve Programlama" and "plan_20260107" under "Analiz edilecek ders". There are two buttons: "Plan faydalığını Beğeniyorum" (top) and "Planı Eleğim" (bottom). At the bottom of the sidebar is a button labeled "Günlük Notasyon AI".

At the bottom of the main content area, there is a green bar containing the message "Harikalı Analiz tamamlandı. 🎉" and the text "Şimdi o enerjiyi masaya koyma zamanı. Adım adım ilerleyeceğiz, sana güveniyorum." followed by "Hadi başlayalım! 🚀".

In the top right corner of the main content area, there are two small buttons: "Kariyer" and "Hocadı Aşkım".

ArtemisAI: Geleceğin Kişiselleştirilmiş Öğrenme Deneyimi

ArtemisAI, öğrenciyi dinleyen, anlayan ve onunla birlikte adapte olan bir sistemdir. Statik planların yarattığı stresi ortadan kaldırır ve her öğrencinin kendi potansiyeline ulaşması için dinamik bir destek sunar. Bu sadece bir çalışma planı değil, akıllı bir öğrenme ortaklığıdır.



Kodu Sanata Dönüştürmek

Kırılgan Bir Sistemden Akıllı ve Dayanıklı Bir Yapıya
Stratejik Mühendislik Yolculuğu

Yolculuğumuz: Adım Adım Mükemmelliğe



1. Sağlam Temel

Sistemin bel kemiğini güçlendirmek ve çökme risklerini ortadan kaldırmak.

2. Maksimum Verim

Daha az kaynakla daha fazlasını başarmak.

3. Akıllı Kararlar

Sisteme otomatik ve zeki davranışlar kazandırmak.

4. Kusursuz Deneyim

Teknik detayları kullanıcı için pürüzsüz bir arayüze dönüştürmek.

5. Sarsılmaz Güvenilirlik

Her koşulda çalışan bir sistem inşa etmek.

Her iyileştirme, bir sonrakinin temelini attı.

**1. ADIM: SAĞLAM BİR
TEMEL İNŞA ETMEK**

Kırılgan JSON İşleme Sorununu Aşmak

****Meydan Okuma****

- LLM'den gelen çıktılar her zaman mükemmel formatta değildir.
- Markdown (```json ``), geçersiz 'backslash' karakterleri ve format dışı metinler, sistemin %15 oranında çökmesine neden oluyordu.



Çözümümüz: Çok Katmanlı Güvenli Ayrıştırma Fonksiyonu

```
# planning_agent.py - extract_json fonksiyonu
def extract_json(text: str) -> dict:
    """LLM çıktısından JSON bloğunu güvenli şekilde ayıklar"""

    # 1. Markdown bloklarını temizle
    if "```\n" in text:
        parts = text.split("```\n")
        for part in parts:
            if part.startswith("{") or part.startswith("json\n{"):
                text = part.replace("json\n", "", 1)
                break

    # 2. JSON başlangıç/bitiş noktalarını bul
    start = text.find("{")
    end = text.rfind("}") + 1
    json_text = text[start:end]

    # 3. Geçersiz backslash'leri düzelt
    try:
        return json.loads(json_text)
    except json.JSONDecodeError:
        # Regex ile invalid escape'leri düzelt
        fixed_text = re.sub(r'\\\(?![\\"/bfnrtu])', r'\\', json_text)
        return json.loads(fixed_text)
```

• 1. Markdown Temizliği

İlk olarak, LLM'in eklediği `json` ve ``` gibi gürültüyü temizliyoruz.

• 2. Sınır Tespiti

JSON bloğunun başlangıç ve bitişini `{` ve `}` karakterleriyle hassas bir şekilde buluyoruz.

• 3. Regex ile Kurtarma

`try-except`bloğu ile ilk denemeyi yapıyoruz. Başarısız olursa, `re.sub` kullanarak geçersiz 'escape' karakterlerini düzeltten bir 'fallback' mekanizması devreye giriyor.

****Vurgu**:** Bu sadece bir hata düzeltme değil, LLM'in öngörülemez doğasına karşı bir savunma hattıdır.

2. ADIM: MAKSİMUM VERİM ELDE ETMEK

4 Ayrı API Çağrısından Tek ve Güçlü Bir İsteğe

Meydan Okuma

- Geri bildirim analizi için eskiden 4 farklı API çağrıları yapılıyordu: `emotion.analyze`, `severity.check`, `coordinator.decide`, `calendar.check`.
- Bu yaklaşım yavaştı (4.0s yanıt süresi), maliyetliydi ve ağ trafigini artırıyordu.



Çözümümüz: Birleşik Geri Bildirim İşleme (Unified Feedback Processing)

```
feedback_agent.py

# feedback_agent.py
def process_feedback_unified(text: str) -> dict:
    """TEK LLM çağrılarında tüm analizi yap - %75 tasarruf!"""

    prompt = f"""
        Analyze feedback: "{text}" """

    Output ONLY JSON:
    {{
        "emotion": {{ "emotion": "pos|neg|neu", "polarity": float }},
        "severity": {{ "severity": "low|med|high", "reason": "str" }},
        "decision": {{ "decision": "inc|dec|keep", "multiplier": float }},
        "schedule_action": {{ "action": "clear_day|none", "target_date": "..." }}
    }}
    """

    result = llm_client.generate(prompt) ←
    return parse_unified_result(result)
```

ÖNCESİ

```
# Çağrı 1
emotion_agent.analyze(text)
# Çağrı 2
severity_agent.check(text)
# Çağrı 3
coordinator.decide(...)
# Çağrı 4
calendar_agent.check(...)
```

SONRASI

```
# TEK ÇAĞRI
llm_client.generate(prompt)
```

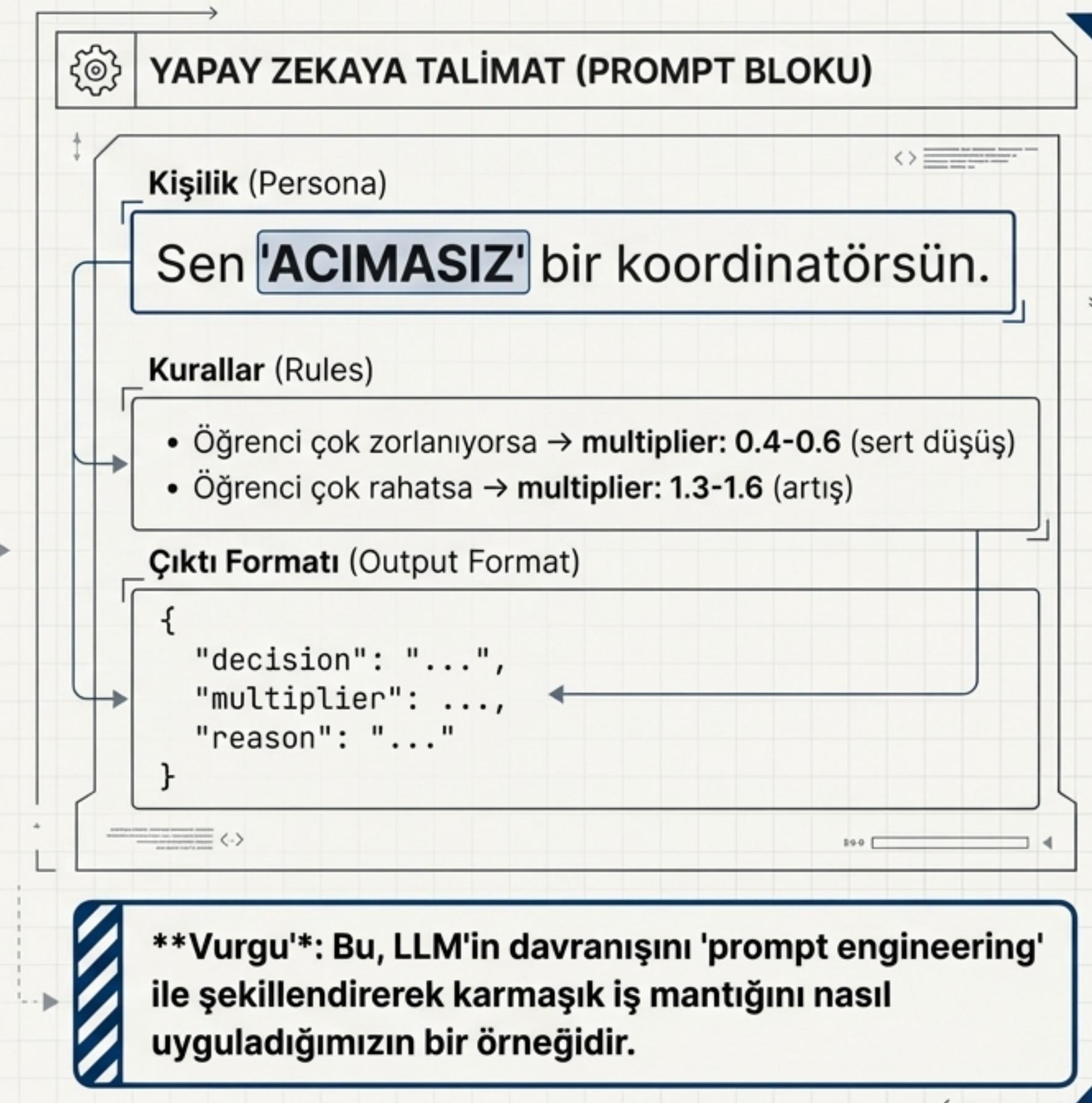
%75 API TASARRUFU! Tek bir akıllı prompt ile hem maliyeti düşürdük hem de yanıt süresini kısalttık.

3. ADIM: AKILLI KARARLAR ALDIRMAK

'Acımasız' Koordinatör: LLM ile Radikal Kararlar

Sistemin, öğrencinin durumuna göre plan yoğunluğunu dinamik olarak ve otoriter bir şekilde ayarlaması gerekiyordu.

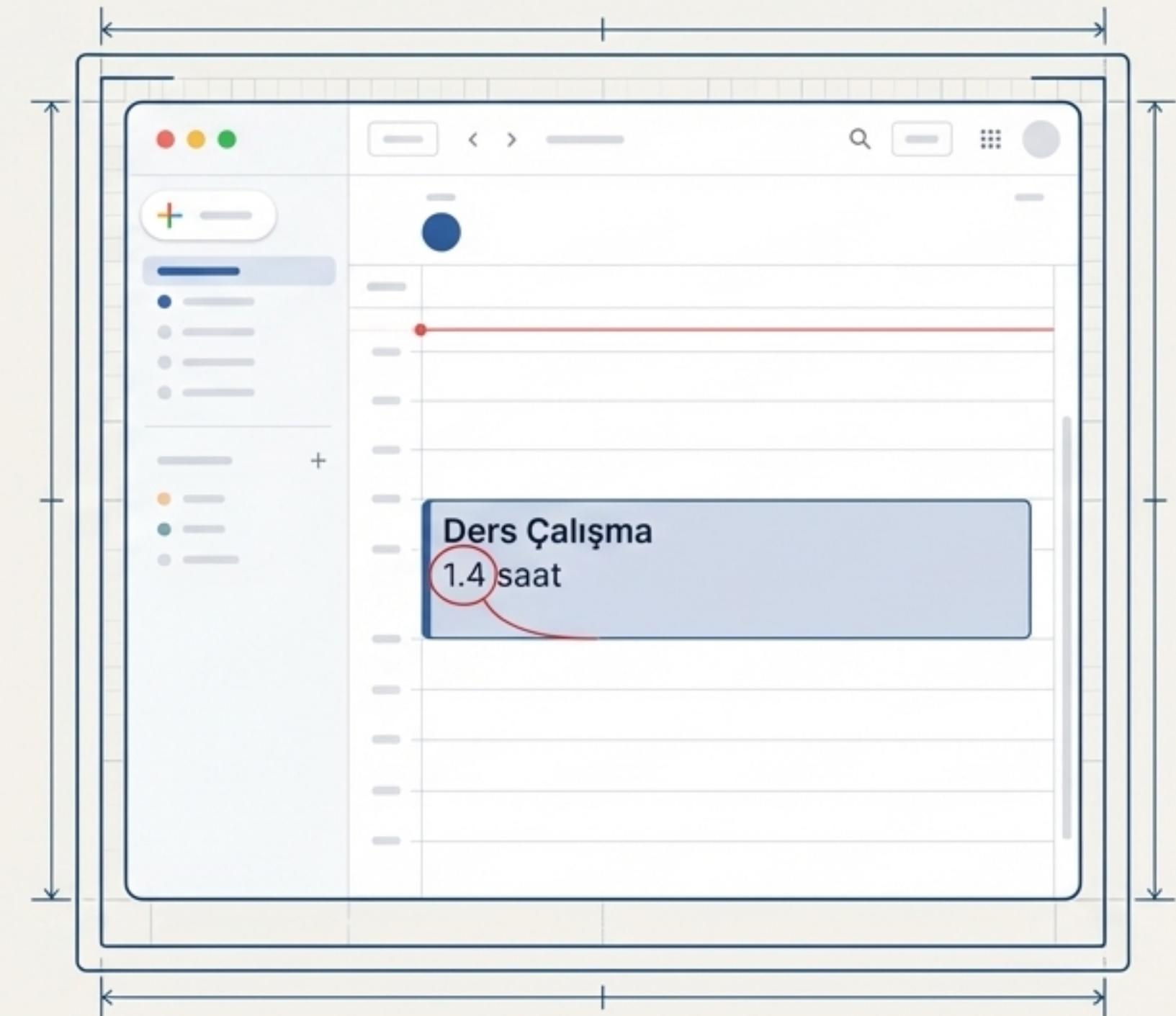
Bunu başarmak için LLM'e 'ACIMASIZ' bir kişilik ve net kurallar tanımladık.



4. ADIM: KUSURSUZ BİR DENEYİM YARATMAK

Matematiksel Doğruluk vs. Mantıksal Kullanım

- “Acımasız” koordinatörün hesapladığı `multipplier=0.7` gibi değerler, `2 saat * 0.7 = 1.4 saat` gibi anlamsız plan süreleri üretiyordu.
- Kimse takvimine “1.4 saat ders” diye bir etkinlik eklemez. Bu, kullanıcı deneyimini zedeliyordu.



Çözümümüz: Mantıksal Yuvarlama (Logical Rounding)

```
def logical_round(val):
    """
    En yakın 0.5 katına yuvarla

    Örnekler:
    0.3 → 0.5
    0.7 → 0.5
    1.2 → 1.0
    1.3 → 1.5
    1.7 → 2.0
    2.1 → 2.0
    """
    rounded = int(float(val) * 2 + 0.5) / 2.0
    return max(0.5, rounded) # Minimum 30 dakika
```

Değerleri, insanların plan yaparken kullandığı en yakın 30 dakikalık (0.5) dilime yuvarlayan basit ama etkili bir fonksiyon yazdık.

Örnekler

0.3 → **0.5**

0.7 → **0.5**

1.2 → **1.0**

1.7 → **2.0**

Bu küçük dokunuş, sistemin ürettiği planları arasında daha doğal ve kullanılır hale getirdi.

5. ADIM: SARSILMAZ GÜVENİLİRLİK SAĞLAMAK

**LLM Çökse Bile
Sistemimiz Çalışmaya
Devam Eder**



Meydan Okuma

- Peki ya LLM API'ı yanıt vermezse? Ya da beklenmedik bir hata oluşursa?
- Kullanıcıya boş bir ekran veya bir hata mesajı göstermek kabul edilemezdi. Sistem her koşulda bir çözüm sunmaliydi.

Çözümümüz: Akıllı Fallback Mekanizması

```
# planning_agent.py
try:
    # LLM ile akıllı plan
    text = client.generate(prompt)
    plan = extract_json(text)
except Exception as e:
    # Fallback: LLM çökse de sistem çalışır
    print(f"⚠️ LLM Hatası: {e}")

    # Basit ama işlevsel plan oluştur
    schedule = []
    for i in range(plan_duration):
        schedule.append({
            "date": (today + timedelta(days=i)).strftime("%Y-%m-%d"),
            "blocks": [{"topic": f"{course} genel çalışma",
                        "hours": daily_hours * 0.5}]
        })
    plan["schedule"] = schedule
```

Try Bloğu:
Akıllı plan denenir.

Başarılı

Kullanıcıya Akıllı Plan
Sunulur.

'Exception' (Hata)

Except Bloğu:
'Fallback' devreye girer.

Felsefe: Mükemmel, akıllı çözüm mevcut olmadığında bile, kullanıcıya her zaman değerli ve işlevsel bir alternatif sunarız. Sistemimiz asla ölmez.

Ve İşte Tüm Bu Çabaların Somut Sonuçları

İşlev	Öncesi	Sonrası	İyileşme
JSON Parse	%60	%95+	+%35
LLM Çağrısı	4 çağrı	1 çağrı	-75%
API Tasarrufu	-	-	%75 ↓
Crash Oranı	%15	%0	-100%
Yanıt Süresi	4.0s	1.5s	-62%
Fallback	Yok	Var	

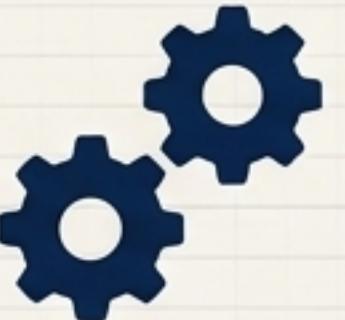
Sadece Kod Yazmadık, Değer İnşa Ettik



1

Önce Stabilite

Sağlam bir temel olmadan hiçbir yapı ayakta kalamaz.



2

Stratejik Optimizasyon

Verimlilik, sadece maliyetleri düşürmek değil, aynı zamanda daha iyi bir kullanıcı deneyimi sunmaktadır.



3

Tasarımla Gelen Dayanıklılık

En iyi sistemler, sadece iyi çalışıklarında değil, işler ters gittiğinde nasıl davranışlarılarıyla ölçülür.

Bu, bir dizi teknik iyileştirmenin değil, mühendislik felsefesinin bir hikayesidir.

Adaptif Çalışma Planlama Sistemi: Öğrenci Geri Bildirimlerinin Nabzı

12 öğrencinin deneyimlerinden damıtılan temel içgörüler ve ürünümüz için stratejik yönelimler.



1. Değer Sunan Temel

Öğrenciler, planlama otomasyonunu, ilerleme takibini ve yapılandırılmış bir yol haritası sunulmasını genel olarak **değerli** buluyor. Bu, sağlam bir temelimiz olduğunu gösteriyor.

2. Adaptasyon Yüzeysel Kalıyor

Sistem “adaptif” olarak konumlandırsa da, mevcut adaptasyon çoğunlukla takvimsel düzenlemelerle sınırlı. Öğrenciler, öğrenme stillerine, zorlandıkları konulara ve derslerin ağırlığına göre **daha derin ve akıllı bir kişiselleştirme** bekliyor.

3. Kullanıcı Kontrolü ve Esneklik Kritik

Kullanıcılar, belirledikleri kuralların (örn. Pazar günü çalışmama) katı bir şekilde uygulanmasını ve arayüzün daha esnek olmasını (örn. çoklu ders seçimi) talep ediyor. Sistem, kullanıcının niyetini daha iyi anlamalı.

Kullanıcıların Değer Verdiği Temel Direkler: Motivasyon ve Yapısal Netlik



Yapı ve Öngörülebilirlik

Öğrenciler, sistemin sunduğu dengeli ve gerçekçi planlamayı takdir ediyor. Konuların mantıksal bir sırayla ilerlemesi ve ilerlemenin net bir şekilde görülebilmesi, kontrol hissi ve devam etme isteği yaratıyor.

Destekleyici Veriler

- Dengeli ve gerçekçi günlük çalışma süreleri (Öğrenci 5)
- Plan güncellmelerinin otomatik ve düzenli yansıması (Öğrenci 8)
- İlerleme takibi ve tamamlanan konuların listelenmesi (Öğrenci 6, 12)

“Genel olarak oluşturulan çalışma planı dengeli ve gerçekçiydi... Bu durum motivasyonumu artırdı.”

- Anonim Öğrenci 5 (2. Sınıf)



Motivasyon ve Geri Bildirim Döngüsü

Sistem, öğrencilerin çalışma alışkanlıklarını sürdürmelerine yardımcı olan önemli motivasyonel unsurlar içeriyor. Ayarların plana anında yansımıması ve zorlanılan konuların tekrar eklenmesi, sistemin onları “dinlediği” hissini veriyor.

Destekleyici Veriler

- Ayarların (örn. max. çalışma saatı) plana anında güncellenmesi (Öğrenci 3)
- Zorlanılan konuların tekrar plana dahil edilmesi (Öğrenci 10)
- Başlangıçta etkili olan günlük motivasyon mesajları (Öğrenci 11)

“İlerleme takibi... kendi durumumu net görmemi sağladı ve çalışmaya devam etme isteği oluşturdu.”

- Anonim Öğrenci 12 (3. Sınıf)

Gerçek Potansiyelimize Ulaşmak İçin 3 Kritik Odak Alanı



Derinlemesine Adaptasyon: “Ne” ve “Nasıl” Öğrenildiğini Anlamak

Problem: Sistem, bir öğrencinin bir konuda neden zorlandığını anlamadan sadece genel yoğunluğu artırıyor veya yüzeysel tekrarlar ekliyor. Derslerin zorluk/kredi ağırlığını ve konuların bilişsel yükünü (zor/kolay dengesi) dikkate almıyor.

Öğrenci Kanıtları: Recursive konuda ek kaynak sunulmaması (Öğrenci 1), derslerin zorluk derecesine göre önceliklendirilmemesi (Öğrenci 7), zor konuların arka arkaya gelmesi (Öğrenci 9), tekrarların süresinin yetersiz kalması (Öğrenci 10).

“Zorlandığım konuya daha odaklı bir düzenleme beklerdim.” – Anonim Öğrenci 1 (1. Sınıf)



Esneklik ve Kullanıcı Kontrolü: Niyetleri Eyleme Dönüşürtmek

Problem: Kullanıcı tarafından belirlenen net kurallar (örn. “Pazar günü çalışma”) sistem tarafından tutarlı bir şekilde uygulanmıyor. Arayüz, birden fazla ders seçimi gibi temel işlemlerde verimliliği düşürebiliyor.

Öğrenci Kanıtları: Çoklu ders seçilememesi (Öğrenci 2), Pazar günü planının otomatik kaydırılmaması (Öğrenci 4).

“Pazar günü çalışmamayı belirtmemeye rağmen planlanan bazı derslerin otomatik olarak pazartesiye kaydırıldığını fark ettim.” – Anonim Öğrenci 4 (1. Sınıf)



Proaktif Rehberlik: Riskleri Öngörmek ve İletmek

Problem: Sistem mevcut durumu raporlamada iyi (örn. ilerleme yüzdesi), ancak proaktif rehberlikte zayıf kalıyor. Öğrencileri potansiyel riskler (örn. bir konuda geri kalma) hakkında önceden uyarın bir mekanizma yok. Motivasyonel içerikler zamanla etkisini yitiriyor.

Öğrenci Kanıtları: Geri kalınan konular için uyarı sistemi eksikliği (Öğrenci 6), tekrarlayan motivasyon mesajlarının etkisini yitirmesi (Öğrenci 11).

Bu üç alanda yapacağımız iyileştirmeler, sistemimizi bir “takvim yöneticisinden” gerçek bir “kişisel öğrenme ortağına” dönüştürecekтир.