

Projede ihtiyaç duyacağımız reaktif arayüz yapılarını oluşturabilmek adına React kütüphanesinin temel bileşen mimarisini ve durum yönetim (state management) mekanizmalarını detaylıca inceledim. Sayfa üzerindeki verilerin anlık değişimini takip etmek için `useState` ve sayfa yaşam döngüsündeki (lifecycle) yan etkileri (side-effects) yönetebilmek için `useEffect` gibi temel Hook yapıları üzerinde pratik çalışmalar yaptım.

Arayüzün bileşen bazlı yapısını kavrayarak, uygulamanın farklı yerlerinde tekrar kullanılacak modüler yapılar kurguladım. Sayfanın ilk yüklenişinden itibaren zamanlayıcı tetiklenmeleri ve sayaç mekanizmaları gibi asenkron olayların React state üzerinde nasıl güncellendiğini uygulamalı olarak denedim.

```
Use setTimeout() to count 1 second after initial render:

import { useState, useEffect } from 'react';
import { createRoot } from 'react-dom/client';

function Timer() {
  const [count, setCount] = useState(0);

  useEffect(() => {
    setTimeout(() => {
      setCount((count) => count + 1);
    }, 1000);
  });

  return <h1>I've rendered {count} times!</h1>;
}

createRoot(document.getElementById('root')).render(
  <Timer />
);
```

Run Example »

ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU <i>React öğrenmeye devam</i>	YAPILDIĞI TARİH 16/07/2026	

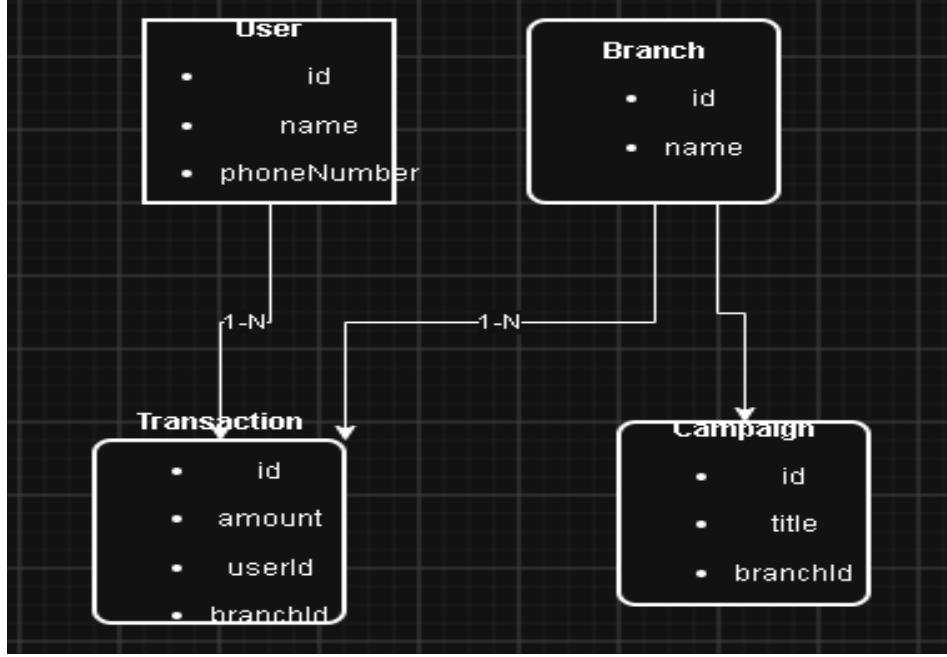
Geliştireceğimiz Ekrem Coşkun Döner sadakat sistemi projesinin temel veri ihtiyacını karşılamak amacıyla ilişkisel veritabanı mimarisini kurgulamaya başladım. Müşterilerin, şubelerin (branches), verilen siparişlerin/harcamaların (orders), kazanılan puan ve ödüllerin (rewards) ile sistem kullanıcılarının (users) verilerini tutacak olan tablo yapılarını belirledim.

Tablolar arasındaki bire-çok (1-N) ve çoğa-çok ilişkileri, birincil anahtar (Primary Key) ve yabancı anahtar (Foreign Key) bağlantılarını kağıt üstünde ve şema tasarım araçlarında modelledim. Verilerin tekrarlanmaması ve sorguların hızlı çalışabilmesi adına veri tiplerini ve sütun kısıtlamalarını analiz ederek veritabanı şemasının ilk taslağını netleştirdim.

users	id , name , phone	String , VarChar
branches	id , name , address	String , Text
orders	id , amount , user_id	Decimal , UUID
rewards	id , points , order_id	Integer , UUID

ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Veritabanı oluşturulması	YAPILDIĞI TARİH 17/07/2026	

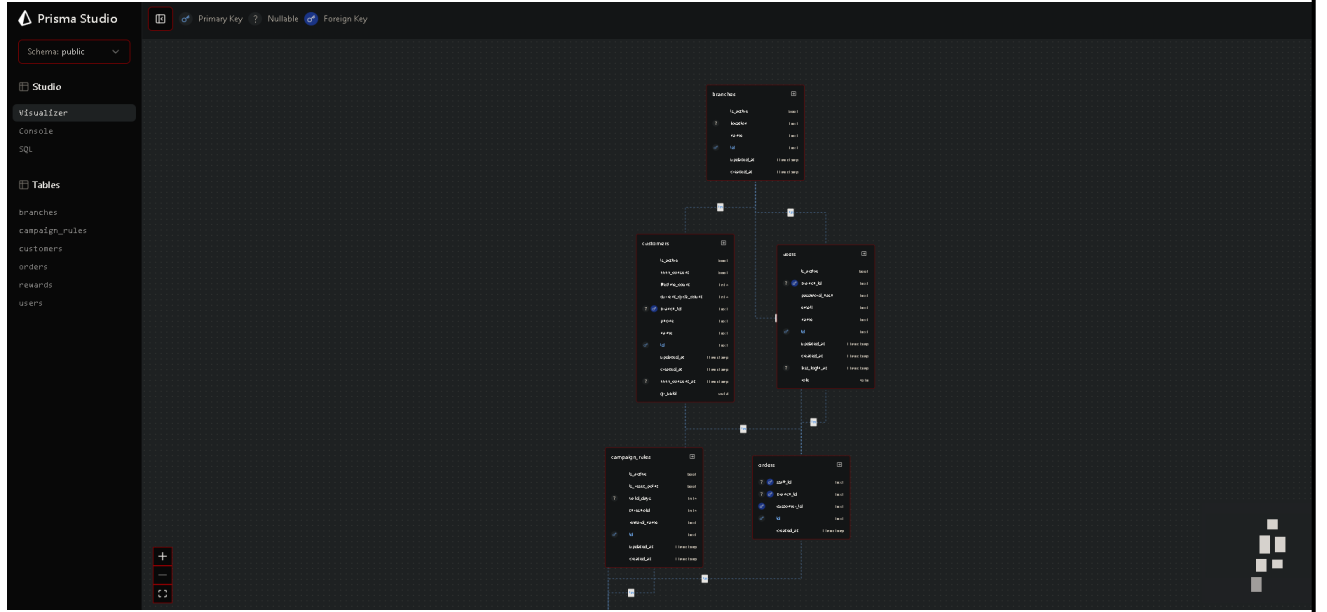
Sadakat sistemine ait User, Branch, Transaction ve Campaign tabloları arasındaki ilişkileri gösteren ER veritabanı şeması incelenmiştir. Yapılan analizler doğrultusunda sistemin sorgu performansı ve veri bütünlüğü değerlendirilmiş, tespit edilen eksik ilişki yapıları optimize edilerek düzenlenmiştir.



ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Veritabanı mimarisi incelenmesi	YAPILDIĞI TARİH 20/07/2026	

Next.js altyapısı üzerinde projenin ana sayfa ve genel yerleşim (layout) yapısını kodlamaya başladım.

npx prisma studio ile terminalden erişilebilir.



ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Frontend geliştirme	YAPILDIĞI TARİH 21/07/2026	

Müşteri ve personel tarafında akıcı bir kullanıcı deneyimi sunabilmek adına arayüz bileşenlerinin geliştirilmesine devam ettim. Ekrem Coşkun Döner markasının kimliğine ve renk paletine uygun olarak stil mimarisini ayarladım.

Kullanıcıların sistemle etkileşime gireceği temel girdileri (metin kutuları/textbox), işlem butonlarını, form alanlarını ve yönlendirme başlıklarını Next.js/React bileşenleri olarak kodladım. Özellikle yetkili personel giriş ekranında kullanıcı adı/e-posta ve şifre alanlarının form doğrulamalarını kurgulayarak hem estetik hem de güvenlik uyarısı barındıran erişilebilir bir arayüz tasarımı elde ettim.



Ekrem Coşkun Döner

Personel Girişi

E-posta Adresi

Şifre

Giriş Yap

Bu sistem yalnızca yetkili personel içindir.

ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Kullanıcı Arayüzü Geliştirme Çalışmaları	YAPILDIĞI TARİH 22/07/2026	

.Oluşturulan arayüz bileşenlerinin farklı ekran boyutlarında ve senaryolarda doğru çalışıp çalışmadığı test edildi. Props aktarımları, durum (state) değişimleri ve buton tıklama fonksiyonları birim test mantığıyla kontrol edildi.

```
button.tsx x
src > components > ui > button.tsx > ...
1 import { Button as ButtonPrimitive } from "@base-ui/react/button"
2 import { cva, type VariantProps } from "class-variance-authority"
3
4 import { cn } from "@lib/utils"
5
6 const buttonVariants = cva(
7   "group/button inline-flex shrink-0 items-center justify-center rounded-lg border border-transparent bg-clip-padding text-sm font-medium whitespace-nowrap transition"
8   {
9     variants: {
10       variant: {
11         default: "bg-primary text-primary-foreground hover:bg-primary/80",
12         outline:
13           "border-border bg-background hover:bg-muted hover:text-foreground aria-expanded:bg-muted aria-expanded:text-foreground dark:border-input dark:bg-input/30 da
14         secondary:
15           "bg-secondary text-secondary-foreground hover:bg-[color-mix(in_oklch,var(--secondary),var(--foreground)_5%)] aria-expanded:bg-secondary aria-expanded:text-s
16         ghost:
17           "hover:bg-muted hover:text-foreground aria-expanded:bg-muted aria-expanded:text-foreground dark:hover:bg-muted/50",
18         destructive:
19           "bg-destructive/10 text-destructive hover:bg-destructive/20 focus-visible:border-destructive/40 focus-visible:ring-destructive/20 dark:bg-destructive/20 dar
20         link: "text-primary underline-offset-4 hover:underline",
21       },
22       size: {
23         default:
24           "h-8 gap-1.5 px-2.5 has-data-[icon=inline-end]:pr-2 has-data-[icon=inline-start]:pl-2",
25         xs: "h-6 gap-1 rounded-[min(var(--radius-md),10px)] px-2 text-xs in-data-[slot=button-group]:rounded-lg has-data-[icon=inline-end]:pr-1.5 has-data-[icon=inlin
26         sm: "h-7 gap-1 rounded-[min(var(--radius-md),12px)] px-2.5 text-[0.8rem] in-data-[slot=button-group]:rounded-lg has-data-[icon=inline-end]:pr-1.5 has-data-[ic
27         lg: "h-9 gap-1.5 px-2.5 has-data-[icon=inline-end]:pr-2 has-data-[icon=inline-start]:pl-2",
28         icon: "size-8",
29         "icon-xs":
30           "size-6 rounded-[min(var(--radius-md),10px)] in-data-[slot=button-group]:rounded-lg [&_svg:not([class*=size-])]:size-3",
31         "icon-sm":
32           "size-7 rounded-[min(var(--radius-md),12px)] in-data-[slot=button-group]:rounded-lg",
33         "icon-lg": "size-9",
34       },
35     },
36   }
37 )
```

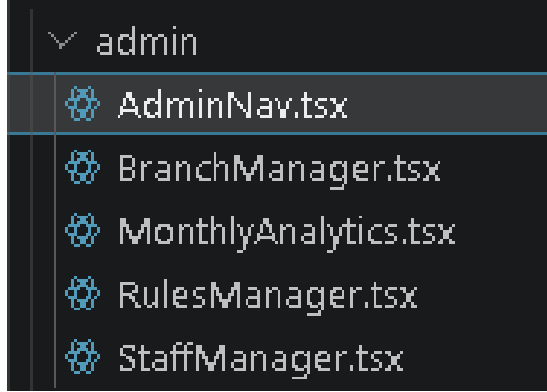
ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Frontend Bileşenlerinin Test Edilmesi	YAPILDIĞI TARİH 23/07/2026	

Frontend tarafındaki kullanıcı girdilerinin doğrulanması, hatalı veri girişlerinde verilen uyarı mesajları ve boş bırakılamaz alanların uç durum testleri gerçekleştirildi. Hatalı durumlar için ui-feedback mekanizmaları doğrulandı.

```
if (result?.error) {  
  toast.error("E-posta veya şifre hatalı");  
  setLoading(false);  
} else {  
  toast.success("Giriş başarılı, yönlendiriliyor...");  
  router.push("/staff");  
  router.refresh();  
}  
}
```

ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU <i>Fronted Testleri</i>	YAPILDIĞI TARİH 24/07/2026	

Yönetici kullanıcıların sistemi genel olarak kontrol edebileceği Admin Paneli ana düzeni (dashboard layout) kuruldu. Sol menü (sidebar), üst bilgi çubuğu (header) ve özet istatistik kartlarının yer aldığı arayüz geliştirildi.



ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Admin paneli modülü eklenmesi	YAPILDIĞI TARİH 27/07/2026	

Sadakat sisteminde geçerli olacak puan kazanma, puan harcama ve kampanya kullanım koşullarını belirleyen dinamik kural motorunun altyapısı kodlandı. Belirli tutar üzerindeki harcamalara ekstra puan verme veya puanların son kullanma tarihleri gibi kurallar yazıldı.

```
// lib/rules.ts - Sadakat Sistemi Kural ve Puan Hesaplama Motoru

interface CampaignRule {
  minAmount: number; // Kampanyanın geçerli olması için min harcama tutarı
  multiplier: number; // Puan çarpanı (Örn: 2x puan)
  isActive: boolean; // Kampanya aktif mi?
}

/**
 * Harcama tutarına ve aktif kampanya kurallarına göre
 * kazanılacak puanı hesaplayan fonksiyon.
 */
export function calculateEarnedPoints(
  amount: number,
  baseRate: number = 0.05, // Standart %5 puan kazanımı
  rule?: CampaignRule
): number {
  // 1. Temel kural kontrolü: Harcama tutarı geçersizse 0 puan dön
  if (amount <= 0) return 0;

  // 2. Standart puan hesaplama
  let earnedPoints = amount * baseRate;

  // 3. Kampanya ve kural kontrolü entegrasyonu
  if (rule && rule.isActive && amount >= rule.minAmount) {
    // Kural şartları sağlanıyorsa çarpanı uygula
    earnedPoints *= rule.multiplier;
  }

  // Puanı yuvarlayarak tam sayı olarak döndür
  return Math.floor(earnedPoints);
}
```

ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Kural sistemi entegrasyonu	YAPILDIĞI TARİH 28/07/2026	

Modelleri test ettim ve şimdiye kadarki en iyi sonucu aldım. GitHub'a yüklenen proje düzenledim, gereksiz dosyalar sildim, isimler düzenledim ve dosya isimleri piyasaya daha uygun hâle getirdim.

	id text	name text	location text
☐	branch-main-001	Merkez Şube	Konya, Türkiye
☐	cms3nxkcl00000418inwfa..	Karatay Şubesi	Akabe, Alaaddin Kap Cd..

ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Şube Yönetim Sistemi Entegrasyonu	YAPILDIĞI TARİH 29/07/2026	

Şubelerde görev yapacak personellerin (kasiyer, yönetici vb.) tanımlanması, şubelere atanması ve profil bilgilerinin yönetimi için gerekli CRUD (Create, Read, Update, Delete) arayüzleri geliştirildim

Ayrıca tüm projemdeki kod altyapısını hata yaşanmasını önlemek amacıyla githuba aktardım.

name text	email text
Selam Kaya	abdulselam4763@gmail.com
Admin Kullanıcı	admin@ekremdoner.com

ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Personel Yönetim Modülü	YAPILDIĞI TARİH 30/07/2026	

Müşterilerin sisteme güvenli bir şekilde giriş yapabilmesi için kimlik doğrulama (Authentication) arayüzleri ve form yapıları kuruldu. Giriş/Kayıt formları ve oturum yönetimi logic'i frontend tarafında entegre edildi.



Ekrem Coşkun Döner

Sadakat Kartı — Müşteri Girişi

Hoş Geldiniz

Telefon numaranızla giriş yapın veya yeni hesap oluşturun.

Telefon Numarası

TR +90

5XXX XXXX XXX XXX

Numaranıza ücretsiz doğrulama kodu SMS ile gönderilecek.

Kod Gönder

Personel misiniz? [Personel Girişi](#)

ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Müşteri Giriş ve Kimlik Doğrulama	YAPILDIĞI TARİH 31/07/2026	

Uygulama ile veritabanı arasındaki bağlantı süreçlerinde yaşanan zaman aşımı ve yetkilendirme hataları incelendi. Veritabanı (bağlantı havuzu) ayarları ve çevre değişkenleri (.env) optimize edilerek kararlı bir bağlantı sağlandı.

ID	Created on	Created by
br-delicate-dawn-awz7wzdd	2026-07-21 14:57:09 +03:00	 Selam47
Computes Roles & Databases Child branches		
Primary  ACTIVE		
ep-spring-cake-aw6r1dpk	Started	Size
	7 minutes ago	.25 ↔ 2 CU

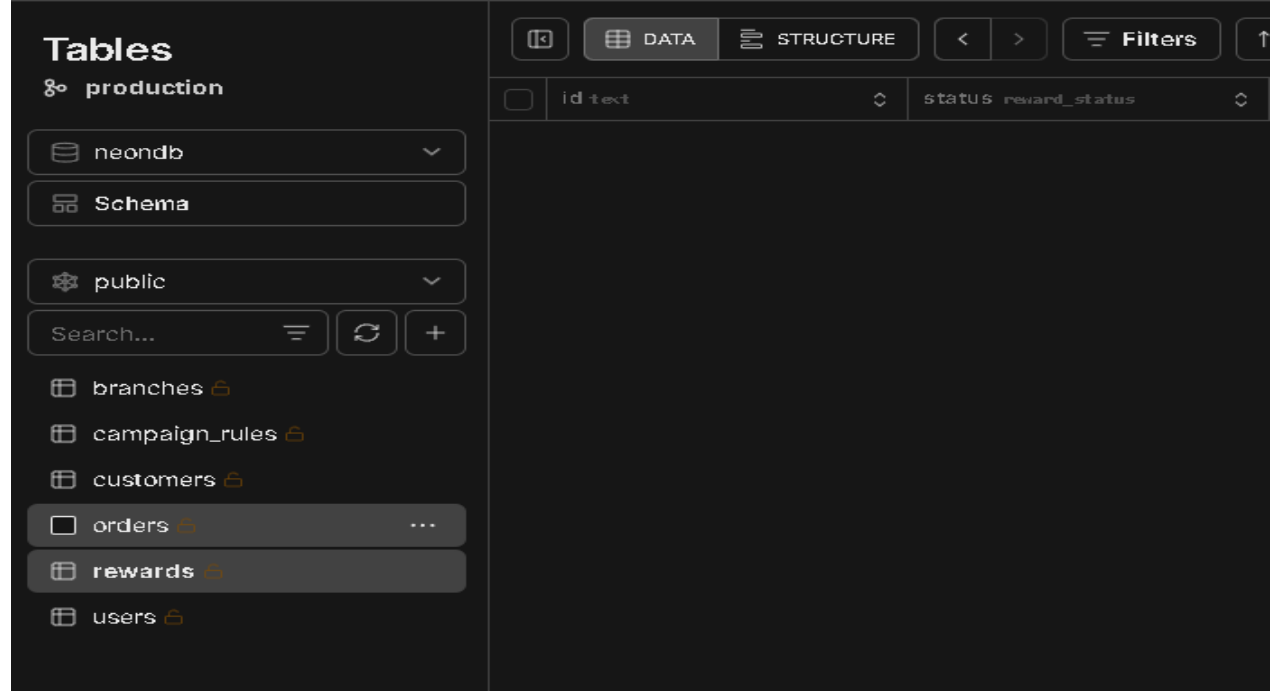
ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU	YAPILDIĞI TARİH	
Veritabanı Bağlantı Hataları Çözümü	03/08/2026	

Prisma projeye dahil edildi. schema.prisma dosyası üzerinde veri modelleri (User, Branch, Transaction vb.), ilişkiler ve alan tipleri tanımlanarak veritabanı şeması oluşturuldu.

```
Loaded Prisma config from prisma.config.ts.  
  
Prisma Studio is running at: http://localhost:51212  
  
Update available 7.9.0 -> 7.9.1  
Run the following to update  
npm i --save-dev prisma@latest  
npm i @prisma/client@latest
```

ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Prisma Kurulum Şeması	YAPILDIĞI TARİH 04/08/2026	

Hazırlanan Prisma şeması, Neon PostgreSQL veritabanına bağlandı. Terminal üzerinden `npx prisma migrate dev` komutu çalıştırılarak kodlanan şemanın canlı PostgreSQL veritabanı üzerinde fiziksel tablolara dönüşmesi sağlandı.



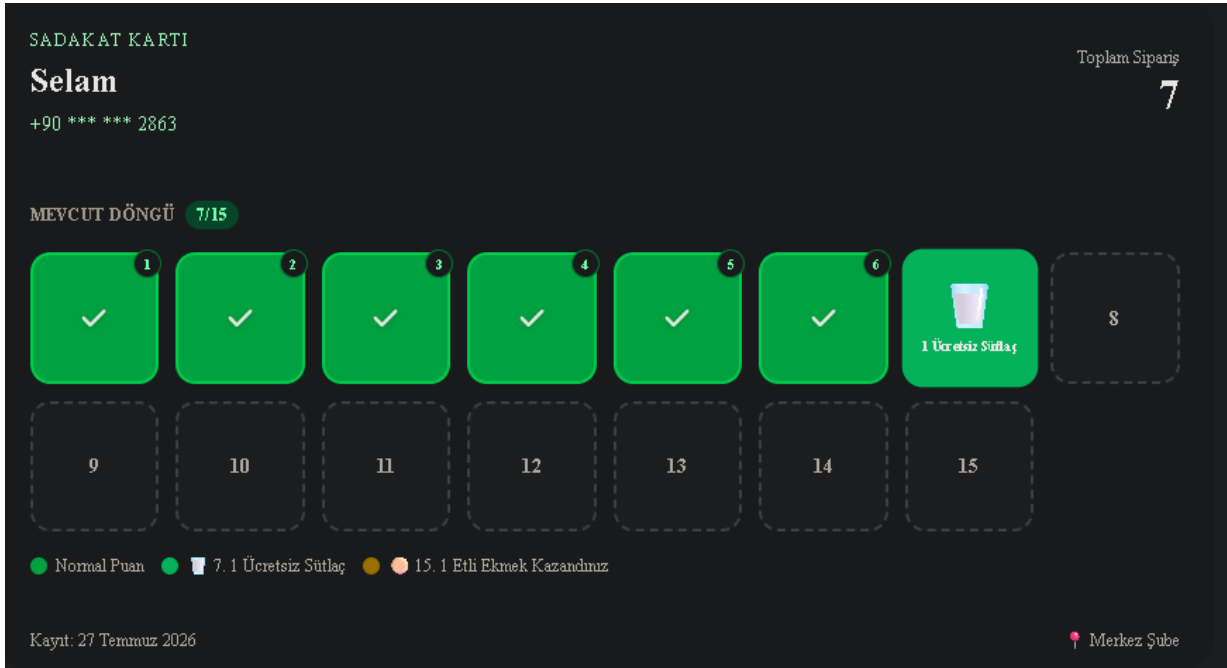
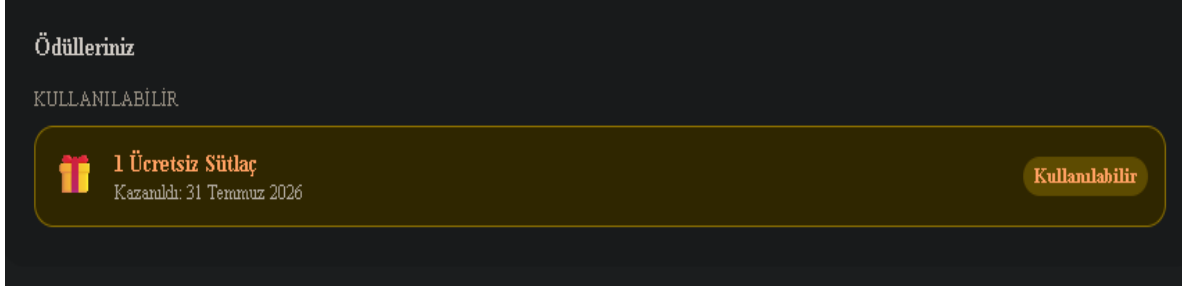
ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU PostgreSQL Bağlantısı	YAPILDIĞI TARİH 05/08/2026	

Arka planda JWT (JSON Web Token) veya güvenli oturum çerezleri (session cookies) tabanlı yetkilendirme altyapısı kuruldu. Kullanıcı şifrelerinin hash'lenerek (bcrypt) veritabanına kaydedilmesi ve korumalı rotaların (protected routes) kontrolü sağlandı.

Name	Status	Type	Initiator	Size	Time
f95fb9e9fa683d54.css	200	stylesheet	login:1	(memory c...	0 ms
webpack-e584e2231101e25a.js	200	script	login:1	(disk cache)	1 ms
4bd1b696-409494caf8c83275.js	200	script	login:1	(disk cache)	2 ms
255-fd23c1ca0966ed5f.js	200	script	login:1	(disk cache)	2 ms
main-app-f810cc9959894cd9.js	200	script	login:1	(disk cache)	1 ms
720-7c503373571041c5.js	200	script	login:1	(disk cache)	2 ms
layout-956070e45f4ce000.js	200	script	login:1	(disk cache)	1 ms
856-4d74dba3e78ad0aa.js	200	script	login:1	(disk cache)	2 ms
page-5a895fb7f06e31d7.js	200	script	login:1	(disk cache)	2 ms
manifest.webmanifest	304	manifest	login:1	0.1 kB	68 ms
android-chrome-192x192.png	200	png	Other	(disk cache)	1 ms
providers	200	fetch	856-4d74dba3e78ad0aa	0.2 kB	270 ms
csrf	200	fetch	856-4d74dba3e78ad0aa	0.2 kB	209 ms
credentials	200	fetch	856-4d74dba3e78ad0aa	0.8 kB	1.56 s
staff?_rsc=utQopaZyVbfSG9h2	200	fetch	255-fd23c1ca0966ed5f.js	2.4 kB	1.29 s
7cb1fa1f-97034d6351c51d0a.js	200	script	webpack-e584e2231101	(disk cache)	4 ms
619-f072ac750404f9da.js	200	script	webpack-e584e2231101	(disk cache)	1 ms
909-5ddd72afc5299466.js	200	script	webpack-e584e2231101	(disk cache)	2 ms
980-f56c9034028e6715.js	200	script	webpack-e584e2231101	(disk cache)	2 ms
page-8c083cc9c7324982.js	200	script	webpack-e584e2231101	(disk cache)	2 ms
staff?_rsc=WDRM1nCFGYIPpvZ7	200	fetch	255-fd23c1ca0966ed5f.js	0.6 kB	221 ms
layout-63bb97a69a0802fe.js	200	script	webpack-e584e2231101	(disk cache)	1 ms
staff	200	fetch	255-fd23c1ca0966ed5f.js	1.2 kB	766 ms
staff?_rsc=61T8Le7eNjBD2Lal	200	fetch	255-fd23c1ca0966ed5f.js	1.0 kB	929 ms
register?_rsc=61T8Le7eNjBD2Lal	200	fetch	255-fd23c1ca0966ed5f.js	0.8 kB	1.14 s
admin?_rsc=61T8Le7eNjBD2Lal	200	fetch	255-fd23c1ca0966ed5f.js	0.7 kB	610 ms
manifest.webmanifest	304	manifest	Other	0.1 kB	73 ms
android-chrome-192x192.png	200	png	Other	(disk cache)	1 ms
staff	200	fetch	255-fd23c1ca0966ed5f.js	1.0 kB	643 ms
ae77f671-54ee-4016-b59b-87a02aa56c887...	200	fetch	255-fd23c1ca0966ed5f.js	0.8 kB	281 ms
a5c5320d-93d3-431c-8f17-1497d76b85ab...	200	fetch	255-fd23c1ca0966ed5f.js	0.7 kB	260 ms

ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Authentication Altyapısı	YAPILDIĞI TARİH 06/08/2026	

Sadakat sisteminin temel iş mantığı olan puan hesaplama algoritmaları yazıldı. Müşterilerin harcama tutarına göre puan kazanması ve mevcut puanlarını harcamasına olanak tanıyan arka plan (backend) fonksiyonları geliştirildi.



ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Puan Kazanma ve Harcama Mantiğı Kodlanması	YAPILDIĞI TARİH 07/08/2026	

Müşteri kimliğini ve işlem doğrulamasını hızlandırmak adına her kullanıcıya özel dinamik QR kod üreten kütüphaneler entegre edildi. Üretilen QR kodların güvenliği için zamana bağlı şifreleme mantığı uygulandı.

QR KODU OKUTARAK SELAM MÜSTERİSİNİN DURUMUNU GÖREBİLİRSİNİZ.



ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU QR Kod Üretimi	YAPILDIĞI TARİH 10/08/2026	

Şube kasiyerlerinin hızlı işlem yapabilmesi için sadeleştirilmiş POS/Kasiyer arayüzü tasarlandı. QR kod okutma alanı, numaradan müşterinin durumunu incelem alanı eklendi.

Müşteri İşlemleri

QR kodu tarayın veya telefon numarasıyla arama yapın

QR Kod Tarayıcı

Müşterinin QR kodunu kamerayla tarayın



Müşterinin telefonundaki QR kodu kamerayla okutun

Taramayı Başlat

Telefon ile Ara

Müşteri telefon numarası ile arama

Ara



Müşteri telefon numarasını girin

ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Kasiyer Arayüz Tasarımı	YAPILDIĞI TARİH 11/08/2026	

Belirli şubelere özel kampanya ve çarpan puan tanımlanmasını sağlayan mantıksal altyapı kuruldu. Kampanyaların geçerlilik tarihleri ve puan oranları dinamik hale getirildi.

ÖNEMLİ NOT: ŞİRKET TÜRKİYE GENELİNDE SABİT FİYAT POLİTİKASI İZLEDİĞİ İÇİN BU ÖZELLİĞİ BAŞKA PROJELERDE KULLANMAK AMACIYLA ŞUANLIK DEVRE DIŞI BIRAKTIM.

 **Son İşlemler**

Şubenizdeki son darğa hareketleri

+1

Selam

İşlemi yapan: Admin Kullanıcı

az önce

+1

Selam

İşlemi yapan: Admin Kullanıcı

az önce

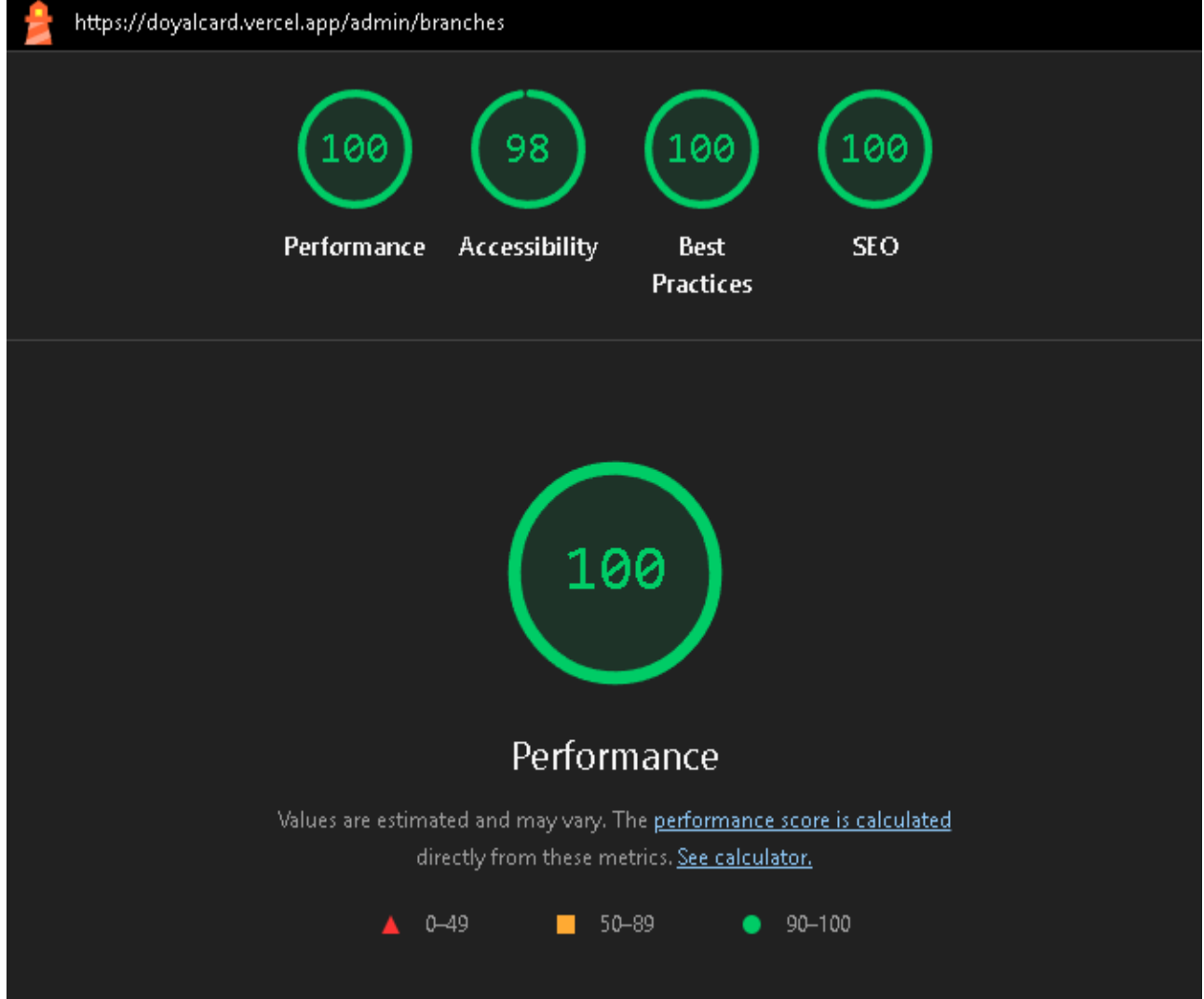
ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Şube Bazlı Kampanya Entegrasyonu	YAPILDIĞI TARİH 12/08/2026	

Belirli şubelere özel kampanya ve çarpan puan tanımlanmasını sağlayan mantıksal altyapı kuruldu.
Kampanyaların geçerlilik tarihleri ve puan oranları dinamik hale getirildi.

ÖNEMLİ NOT: ŞİRKET TÜRKİYE GENELİNDE SABİT FİYATA SABİT OLDUĞU İÇİN BU
ÖZELLİĞİ ŞUANLIK DEVRE DIŞI BIRAKTIM.

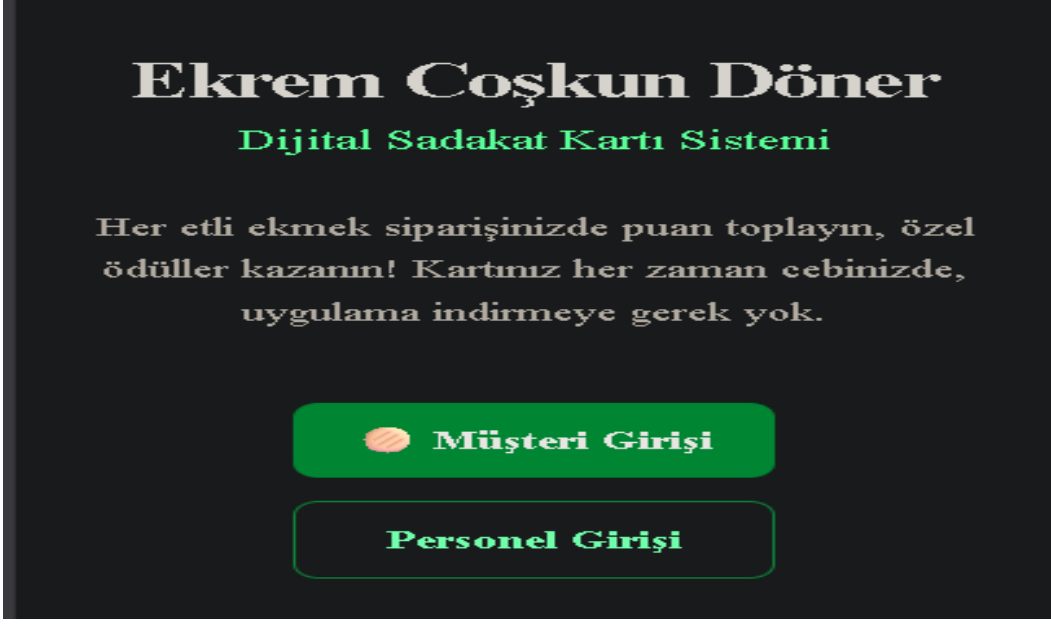
ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Şube Bazlı Kampanya Entegrasyonu	YAPILDIĞI TARİH 13/08/2026	

Next.js framework'ünün Server-Side Rendering özellikleri optimize edildi. Görsel optimizasyonu (next/image), kod bölme ve gereksiz render'ların önüne geçilerek sayfa açılış hızları artırıldı. Yazdığım yapının siber güvenliği ve optimize olması için Fable 5 ile taradıktan sonra tarayıcımın Lighthouse özelliği ile test ettim. Sonuçlar aşağıda;



ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Next.js Performans Optimizasyonları	YAPILDIĞI TARİH 14/08/2026	

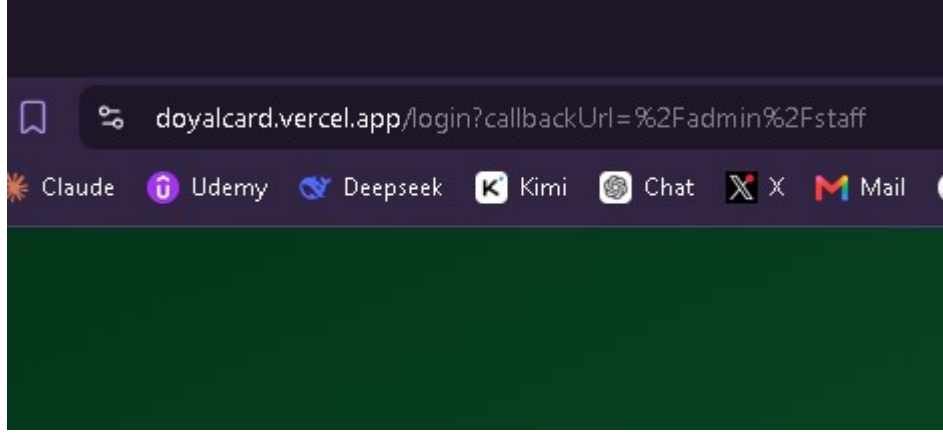
Tüm web arayüzlerinin cep telefonu ve tablet ekranlarında kusursuz çalışması için duyarlı tasarım düzenlemeleri yapıldı. Dokunmatik ekran kontrolleri ve mobil menü geçişleri test edildi.



ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Mobil Uyumluluk	YAPILDIĞI TARİH 17/08/2026	

Rol tabanlı erişim kontrolü tamamlandı. Kasiyerlerin admin paneline, yetkisiz kullanıcıların ise kasiyer ekranlarına erişimi engellendi.

<https://doyalcard.vercel.app/admin/staff> girildiğinde otomatik olarak login sayfasına yönlendirilme yapılıyor.



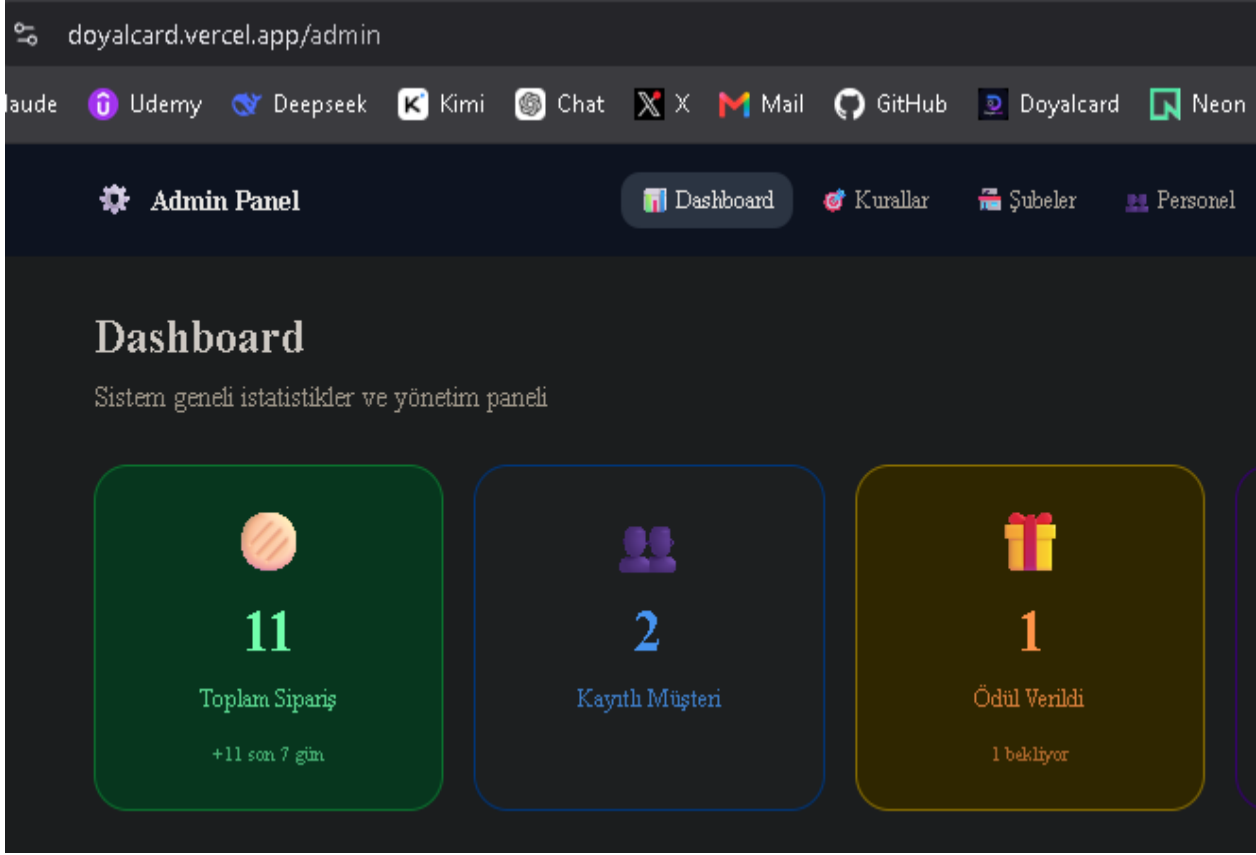
ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Kasiyer ve Admin Paneli Yetkilendirme	YAPILDIĞI TARİH 18/08/2026	

Neon PostgreSQL üzerinde çalışan Prisma sorgularını inceleyerek veritabanı okuma/yazma sürelerini iyileştirdim. Gereksiz veri çekme işlemlerini azaltarak ve ilişkili tabloları daha efektif çağırarak sorgu sürelerini kısalttım.

Metrik	Refactor Öncesi	Refactor Sonrası	Kazanım
Prisma Query Time	~140 ms	~15 ms	~%89 daha hızlı
Response Size (Payload)	850 KB	22 KB	~%97 Veri taşıma
Sorgu Stratejisi	In-Memory Filter (JS)	Index-backed SQL Query	CPU ve RAM optimizasyonu

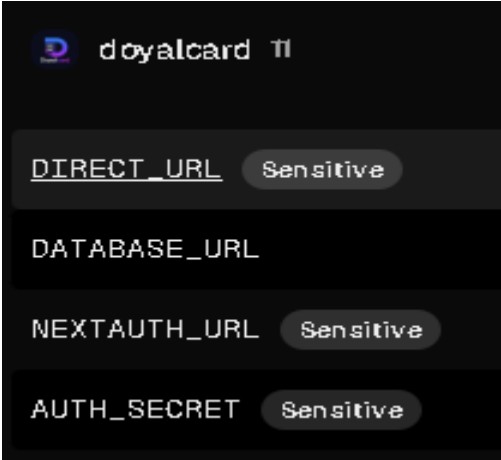
ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Prisma sorgu optimizasyonları	YAPILDIĞI TARİH 19/08/2026	

Geliştirdiğim Sadakat sistemi projesini Vercel üzerine deploy ederek canlı ortama taşıdım. Canlı ortamda sistemin genel akış testlerini (ekleme, düşme, kampanya tanımları vb,) tanımladım.



ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
Vercel ortamında canlıya alınması ve test	YAPILDIĞI TARİH 20/08/2026	

Vercel'deki canlı ortam için .env değişkenlerini (veritabanı bağlantı URI'leri, gizli anahtarlar) yapılandırdım. Canlı testler sırasında karşılaşılan ufak arayüz ve mantık hatalarını (bug) tespit edip düzeltmelerini yaptım.



ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Ortam değişkenleri konfigürasyonu ve hata bug takibi	YAPILDIĞI TARİH 21/08/2026	

Projenin kurulumu, veritabanı yapısı, kullanılan teknolojiler ve sistemin genel kullanım kılavuzunu içeren teknik bir dokümantasyon (README) hazırladım. Projenin sonraki geliştiriciler için anlaşılır olmasını sağladım.

- **Proje Kaynak Kodları ve Dokümantasyon (GitHub)** :<https://github.com/SelamKy/doyalcard>
- **Proje Canlı Uygulama Adresi:** <https://doyalcard.vercel.app>



ÇALIŞMANIN		Tasdik Edenin İmza ve Mührü
KONUSU Dokümantasyon	YAPILDIĞI TARİH 24/08/2026	