

**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**İNTERNET PROGRAMLAMA DERSİ FİNAL PROJESİ**

**SPLİTABLE**

23120205035, Kerem YILMAZ

231202025038, Muhammet Burak ASARCIKLI

**Ders Sorumlusu**

Dr. Öğr. Üyesi Muhammet Sinan BAŞARSLAN

Ocak, 2025

İstanbul Medeniyet Üniversitesi, İstanbul

**İÇİNDEKİLER**

[1. GİRİŞ 1](#_Toc219640584)

[1.1. Proje Özeti 1](#_Toc219640585)

[1.2. Amacı ve Hedefleri 1](#_Toc219640586)

[1.3. Proje Kapsamı 2](#_Toc219640587)

[1.4. En Güçlü Alternatifler ve Splitable’ın Avantajları 2](#_Toc219640588)

[1.4.1. En Güçlü Alternatifler 2](#_Toc219640589)

[1.4.1.1. Splitwise 2](#_Toc219640590)

[1.4.1.2. Tricount 3](#_Toc219640591)

[1.4.2. Splitable’ın Avantajları 3](#_Toc219640592)

[2. UYGULAMA TASARIMI VE ÖZELLİKLERİ 3](#_Toc219640593)

[2.1. Uygulama Özellikleri 3](#_Toc219640594)

[2.2. Kullanıcı Tipleri ve Roller 5](#_Toc219640595)

[2.3. UI Betimleme 6](#_Toc219640596)

[2.3.1. Web Tasarım 6](#_Toc219640597)

[3. SİSTEM TASARIMI VE TEKNOLOJİLER 12](#_Toc219640598)

[3.1. kullanılmıştır Teknolojiler 12](#_Toc219640599)

[3.1.1. Web Uygulama Frontend (React + Tailwind CSS) 12](#_Toc219640600)

[3.1.2. Backend (Node.js + Express.js) 13](#_Toc219640601)

[3.1.3. Veri Tabanı (PostgreSQL) 13](#_Toc219640602)

[3.3. Database Tasarımı 14](#_Toc219640603)

[3.3.1. ENUM’lar 14](#_Toc219640604)

[3.3.2. Tablolar 15](#_Toc219640605)

[3.4. Güvenlik 22](#_Toc219640606)

[3.4.1. Kapsam, Varsayımlar ve Olası Tehdit Modeli 22](#_Toc219640607)

[3.4.2. Veri Sınıflandırma 22](#_Toc219640608)

[3.4.3. Güvenlik Mimarisi 22](#_Toc219640609)

[3.4.4. Özetle Yapılacak Kontroller 22](#_Toc219640610)

[4. PROJE YÖNETİMİ VE GELİŞTİRME SÜRECİ 24](#_Toc219640611)

[4.1. Proje Planı ve Görev Dağılımı 24](#_Toc219640612)

[4.1.1. Proje Ekibi 24](#_Toc219640613)

[4.1.2. Görev Dağılımı 24](#_Toc219640614)

[4.2. Geliştirme Yöntemi 25](#_Toc219640615)

[5. GELECEK PLANLARI VE SONUÇ 26](#_Toc219640616)

[5.1. Gelecek Planları 26](#_Toc219640617)

[5.2. Sonuç ve Beklenen Katkılar 27](#_Toc219640618)

[6. KAYNAKÇA 28](#_Toc219640619)

# 1. GİRİŞ

## 1.1. Proje Özeti

Splitable, kullanıcıların kendi aralarındaki borç ve alacak ilişkilerini kolayca takip edebilmelerini sağlayan bir uygulamadır. Uygulama, özellikle arkadaş gruplarının ortak harcamalarını yönetmesini ve kimin kime ne kadar borçlu olduğunun takip edilebilmesini amaçlar.

Proje kaynak kodlarına GitHub üzerinden erişilebilir:  
 <https://github.com/keremy321/Splitable>

## 1.2. Amacı ve Hedefleri

Splitable uygulamasının temel amacı, bireylerin ve arkadaş gruplarının aralarındaki borç–alacak ilişkilerini düzenli ve kolay bir şekilde takip etmelerini sağlamaktır. Günlük yaşamda sıkça karşılaşılan “kim kime ne kadar borçlu?” problemini ortadan kaldırarak, kullanıcıların finansal ilişkilerini dijital ortamda güvenli bir biçimde yönetmesine yardımcı olmayı hedefler.

Ayrıca Splitable, ortak ödemelerin paylaşımını da kolaylaştırır. Örneğin, bir grup arkadaşın restoranda yaptığı toplam ödemenin kişi başına düşen tutarı otomatik olarak hesaplanır ve herkesin ne kadar ödemesi gerektiği açıkça görüntülenir. Bu sayede, ortak harcamaların yönetimi hem daha adil hem de daha şeffaf hale gelir.

Uygulama, sadece tek seferlik borç işlemleri için değil, aynı zamanda Netflix, Spotify veya YouTube Premium gibi ortak aboneliklerin takibini de hedeflemiştir. Kullanıcılar, bu tür aboneliklerin aylık ödemelerini paylaşabilir, ödeme tarihlerini takip edebilir ve paylarına düşen tutarı kolayca görüntüleyebilirler.

Splitable’ın hedef kitlesi şunlardır:

* **Arkadaş Grupları:** Birlikte seyahat eden, yemek yiyen veya etkinliklere katılan arkadaşların ortak harcamalarını kolayca bölüşmesini sağlar.
* **Ev Arkadaşları:** Kira, fatura ve ortak giderlerin aylık takibini dijital ortamda yapabilmek için uygundur.
* **Abonelik Paylaşan Kullanıcılar:** Netflix, Spotify, YouTube Premium gibi ortak abonelikleri paylaşan kullanıcılar için düzenli ödeme takibi sağlar.
* **Küçük Gruplar ve Takımlar:** Ortak projelerde veya etkinliklerde masrafları paylaşan ekiplerin hesap takibini kolaylaştırır.

Splitable projesinin hedefleri, kullanıcı deneyimini kolaylaştırmak ve finansal işlemlerin dijital ortamda daha pratik hale getirilmesini sağlamaktır. Bu kapsamda temel hedefler şunlardır:

* **Basit ve anlaşılır bir arayüz** ile kullanıcıların borç ekleme, görüntüleme ve ödeme işlemlerini kolayca yapabilmesini sağlamak.
* Ortak harcamaların **otomatik hesaplanmasını** sağlayarak kişi başına düşen borç miktarını adil bir şekilde göstermek.
* **Grup sistemi** aracılığıyla kullanıcıların arkadaş çevreleriyle veya ekipleriyle ortak harcamalarını yönetebilmesini mümkün kılmak.
* **Abonelik takibi sistemi** geliştirerek Netflix, Spotify, YouTube Premium gibi ortak aboneliklerin düzenli ödemelerini takip edebilmek.
* Harcamaların ve borç geçmişinin **görsel ve istatistiksel** olarak izlenebilmesi.

## 1.3. Proje Kapsamı

Splitable, kullanıcıların kişisel veya grup bazlı finansal ilişkilerini kolayca yönetmelerini sağlayan bir web uygulamadır. Proje kapsamında uygulamanın mobil sürümü öncelikli olarak geliştirilmiştir, ardından web arayüzü desteklenmiştir.

Kullanıcılar; uygulama üzerinden borç ekleyebilir, ödemelerini kaydedebilir, grup oluşturabilir ve tüm bu işlemleri güvenli bir şekilde takip edebilirler. Restoran, seyahat, abonelik işlemleri veya fatura gibi ortak harcamalarda toplam tutar otomatik olarak kişi sayısına bölünür; böylece herkesin payına düşen miktar açıkça gösterilir.

Splitable yalnızca tek seferlik borç işlemleriyle sınırlı değildir. Kullanıcılar Netflix, Spotify veya YouTube Premium gibi ortak aboneliklerini sisteme ekleyerek aylık ödemeleri paylaşabilir, ödeme tarihlerini takip edebilir ve geçmiş harcamalarını istatistiksel grafiklerle görüntüleyebilirler.

Uygulama aynı zamanda grup yönetimi, kullanıcı rolleri, fiş/dekont yükleme, ödeme onayı ve bildirim sistemleri gibi temel fonksiyonları da kapsar. Uzun vadede ise bazı ödeme sistemleriyle anlaşmalı bir şekilde sanal ödeme entegrasyonu ile kullanıcıların doğrudan ödeme yapabilmesi ve yalnızca abonelikler için sanal ortak bir kart oluşturulması hedeflenmektedir.

Bu kapsam doğrultusunda Splitable, bireylerin finansal ilişkilerini şeffaf, modern ve kullanıcı dostu bir ortamda yönetebilecekleri, çok yönlü bir finans paylaşım platformu olmayı hedeflemiştir.

## 1.4. En Güçlü Alternatifler ve Splitable’ın Avantajları

### 1.4.1. En Güçlü Alternatifler

### 1.4.1.1. Splitwise

|  |  |
| --- | --- |
| Temel Özellikler | Pro Özellikler |
| Grup ve arkadaş ekleme | Para birimi dönüştürme |
| Harcamaları bölme, borç kaydı tutma | Fiş tarama (kamera ile fiş ekleme) |
| Eşit veya eşit olmayan bölme seçenekleri | Harcama kalemlerine ayırma (itemization) |
| Yüzdeye veya paya göre bölme | Grafik ve istatistikler (charts and graphs) |
| Toplam bakiyeleri hesaplama | Harcama arama özelliği |
| Borçları basitleştirme (karşılıklı borçları otomatik düşürme) | Varsayılan bölme ayarlarını kaydetme (default splits) |
| Tekrarlayan harcamalar (örneğin aylık kira gibi) | Tamamen reklamsız deneyim |
| Çevrimdışı mod (internet olmadan kayıt tutma) | Yeni özelliklere erken erişim |
| Bulut senkronizasyonu (verilerin cihazlar arasında otomatik eşitlenmesi) | Sınırsız harcama kaydı |
| Toplam harcama özeti | İşlem dışa aktarımı / veri içe aktarma (transaction import) |
| Harcamaları kategorilere ayırma |  |
| 7+ dil desteği |  |
| 100’den fazla para birimi desteği |  |
| Ödeme entegrasyonları (örneğin Venmo bağlantısı) |  |

Splitwise, temel olması gereken birçok özellikliği yalnızca ücretli kullanıcılara sunarak erişilebilirliği azaltmaktadır. Splitable, bu ücretli özellikleri tüm kullanıcılara ücretsiz sunarak finansal paylaşım uygulamaları arasında daha adil ve kapsayıcı bir deneyim sağlamayı hedeflemiştir.

### 1.4.1.2. Tricount

|  |  |  |
| --- | --- | --- |
| Temel Özellikler | | |
| Başlıca Özellikler | **Seyahat Özellikleri** | **Diğer Özellikler** |
| Gider Takibi | Para Çevirici | Analizler ve İstatistikler |
| Tutar Özelleştirme | eSIM | Splitwise İçe Aktarıcı |
| Giderleri Hallet | Fotoğraf yükleme | Hesap Makinesi |
| Ödeme Talebi | Çevrim dışı mod |  |
| Ücretsiz Kart |  |  |

Tricount, Splitwise’a kıyasla daha az özelliklere sahip olmasına rağmen eSIM ve ücretsiz kart gibi bazı ek özelliklere sahip olduğu görünüyor.

### 1.4.2. Splitable’ın Avantajları

Splitable, mevcut pazar liderleri olan Splitwise ve Tricount uygulamalarının eksik kaldığı alanlara yenilikçi çözümler sunarak fark yaratmayı amaçlamaktadır. Mevcut uygulamalar genellikle temel borç paylaşımı özelliklerini barındırmakta ancak ödeme onayı, fiş/dekont yükleme, abonelik yönetimi ve kullanıcı yetkilendirme gibi ileri seviye özellikleri ya ücretli olarak sunmakta ya da tamamen dışarıda bırakmaktadır.

Splitable, rakiplerinden farklı olarak kullanıcıya hem bireysel hem grup bazlı finans yönetimi olanağı sunar. Uygulama, abonelik paylaşımı, ödemelerin belgelenmesi, rol tabanlı kontrol, görsel analizler ve otomatik hatırlatmalar gibi gelişmiş özellikleriyle:

* Splitwise’ın ücretli özelliklerini ücretsiz hale getirir,
* Tricount’un eksik olduğu yönetimsel kontrolleri ekler,
* Modern tasarım ve güvenli altyapı ile finans paylaşımını dijital ortamda daha verimli ve güvenilir hale getirir.

Sonuç olarak Splitable; hem kullanıcı deneyimi hem de teknik kapasite bakımından mevcut alternatiflerden daha bütünsel, erişilebilir ve yenilikçi bir finans paylaşım çözümü sunmayı hedeflemiştir.

# 

# 2. UYGULAMA TASARIMI VE ÖZELLİKLERİ

## 2.1. Uygulama Özellikleri

1. Kullanıcı Yönetimi
   1. Kayıt/Giriş: E-posta + şifre veya misafir girişi ile basit işlemler
   2. Profil: Ad-soyad, Avatar, IBAN’lar, Para birimi tercihi, bildirim tercihleri…
   3. Güvenlik: E-posta doğrulama, şifre bcrypt ile hash…
2. Borç Yönetimi
   1. Borç Oluşturma: Alacaklı, borçlu(lar), başlık, açıklama, tutar, para birimi, tarih ve son tarih, fiş.
   2. Bölme Seçenekleri: Eşit Bölme, eşit olmayan bölme, yüzde payına göre bölme…
   3. Para Birimi Değiştirme
   4. Düzenleme/Silme: Yetkiye göre
   5. Tekrarlayan borç: periyot (haftalık/aylık), bitiş tarihi/sonsuz, atlama.
   6. Sadeleştirme: İki kişinin karşılıklı olan borçlarında sadeleştirme yapılması.
   7. Arama/Filtreleme: kişi, tarih aralığı, kategori, durum (açık/ödenmiş/itiraz).
3. Ortak Harcama Hesaplama
   1. Girdi: toplam tutar, katılımcılar, bölme türü (eşit/eşit olmayan/%/hisse).
   2. Çıktı: kişi başı pay, kim kime ne kadar borçlu tablosu, toplam bakiye.
   3. Yuvarlama kuralları: 2 ondalık, kalan kuruş dağıtımı en çok borcu olana/ilk ekleyene.
   4. Kategori: Yemek, Ulaşım, Fatura, Abonelik, Diğer vb. (özelleştirilebilir).
4. Borç Ödeme
   1. Ödeme talebi: borçlu, “Ödeme Yaptım” → açıklama + dekont (görsel/PDF)
   2. Onay akışı: alacaklı Kabul/Reddet.
   3. Kısmi ödeme: tek borca birden çok ödeme; kalan tutar otomatik hesaplanır.
   4. Durumlar: Açık → Ödeme talebi → Onaylandı/Reddedildi → (tamamlandı).
   5. İade/geri alma: yanlış onay için yönetici/üye düzeyinde geri alma.
5. Fiş/Dekont Yönetimi
   1. Dosya tipleri: JPG/PNG/PDF, max boyut (örn. 5 MB)
   2. Depolama
   3. Görüntüleme: küçük önizleme + tam ekran.
   4. İndirebilme
6. Gruplar
   1. Grup: ad, açıklama, fotoğraf, varsayılan para birimi.
   2. Üyelik: davet linki/QR, daveti kabul/ret, üyeyi çıkarma.
   3. Aktivite akışı: “A, 300₺ harcama ekledi”, “B, ödeme talebi gönderdi” gibi log.
   4. Arşivleme: pasif grupları arşivle, veri saklama devam.
7. Roller ve Yetkilendirme
   1. Roller: Sahip (grup kurucusu), Yönetici, Üye, rol üretme.
   2. Yetkiler: harcama/borç ekle-sil, ödeme onay, üye yönetimi.
8. Abonelik Yönetimi
   1. Yapı: servis adı (Netflix), plan tutarı, periyot, fatura günü, katılımcılar, kişi payı.
   2. Tahsilat akışı: otomatik oluşturulan aylık “borç kayıtları”; hatırlatma + ödeme onayı.
   3. Durumlar: aktif, askıda, iptal; ödenmedi durumunda gecikme bildirimi.
   4. Rapor: kişi başı maliyet.
9. Bildirim Sistemi
   1. Anlık bildirim: yeni borç, ödeme talebi, onay/ret, davet.
   2. E-posta: hesap işlemleri, parola sıfırlama.
   3. Borç son tarih hatırlatma
10. Analitik ve Görselleştirme
    1. Grafikler: aylık toplam harcama, kategori dağılımı, kişi bazlı borç/alacak trendi.
    2. Filtre: grup, kişi, tarih, kategori, para birimi.
    3. Görüşler: “Bu ay geçen aya göre %X fazla harcadın”; en çok birlikte harcama yapılan kişi vb.
11. Arama, Filtre, Sıralama
    1. Metin arama (başlık/açıklama), tarih aralığı, durum, kategori, kişi.
    2. Sıralama: en yeni, tutar, kategori, onay durumu.
12. Davet & Paylaşım
    1. Davet linki/QR: tek kullanımlık veya süreli; iptal edilebilir.
    2. Rol atama: davet linkine varsayılan rol (üye/görüntüleyici).
13. Güvenlik
    1. Auth
    2. Veri doğrulama
    3. Dosya güvenliği
    4. Loglama
    5. Gizlilik
14. Erişilebilirlik ve Kullanılabilirlik
    1. Tema: açık/koyu.
    2. Dil: EN/TR +
15. Yönetim Paneli (Admin)
    1. Sistem istatistikleri: kullanıcı sayısı, aktif grup, depolama kullanımı.
    2. Bakım: yedekleme, migrasyon kontrol.

## 2.2. Kullanıcı Tipleri ve Roller

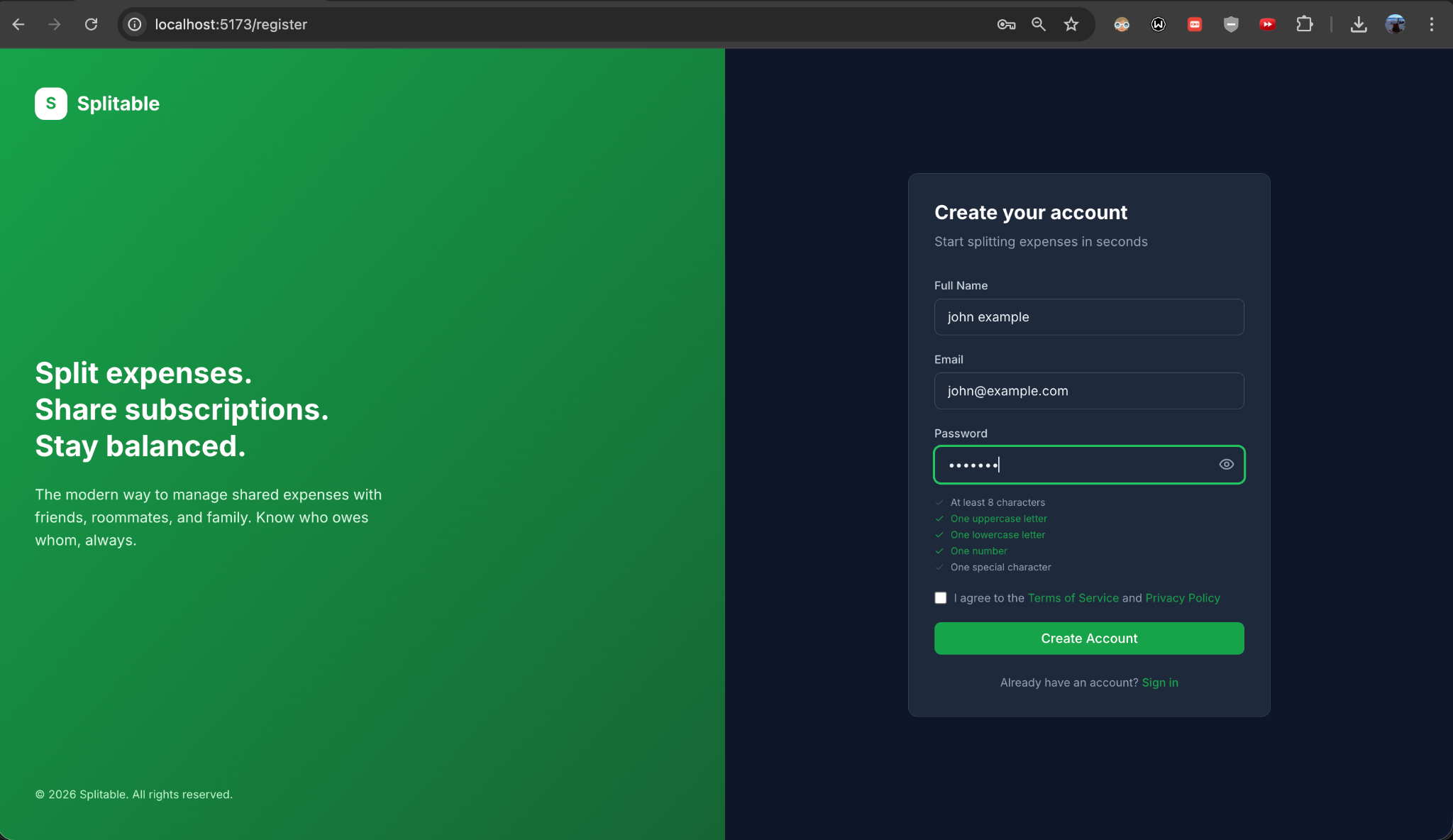
Splitable uygulamasında kullanıcılar farklı yetki düzeylerine sahiptir. Bu roller, uygulama içindeki işlemleri güvenli ve düzenli biçimde yönetmek için oluşturulmuştur. Roller hem bireysel kullanımı hem de grup yönetimini kapsar.

|  |  |  |
| --- | --- | --- |
| Kullanıcı Tipi | Açıklama | Örnek Senaryo |
| Standart Kullanıcı | Uygulamayı kullanan temel kullanıcı tipidir. Borç ekleme, ödeme gönderme, grup katılımı gibi işlemleri yapabilir. | Bir arkadaşına 100 TL borç gönderen veya ödeyen kişi. |
| Grup Üyesi | Bir grubun parçası olan kullanıcıdır. Grup içi borçları görebilir. Borç ekleme, ödeme, düzenleme talebi gönderebilir. Üye ekleme talebi gönderebilir. | “Tatil” grubuna dahil olan kişi. |
| Grup Yöneticisi | Üye ekleme/çıkarma (yalnızca grup üyesi rolünde olanlar) ve borç ekleme/silme/düzenleme yapabilir. | “Tatil” grubunun yöneticisi birini gruba davet edebilir ve borçları ekleyip, silip, düzenleyebilir. |
| Grup Kurucusu | Grup oluşturma ve silme, üye ekleme/çıkarma, yetkilendirme ve borç ekleme/silme/düzenleme işlemlerini yapabilir. | “Tatil” grubunun kurucusu. İstediği kişiye yeni rol verebilir. |
| Opsiyonel Roller | Grup Kurucusu tarafından oluşturulan yeni rollerdir. Rolün ismi ve yetkileri girilir. | “Tatil” grubunda kurucu tarafından oluşturulmuş sadece borç görme yetkisine sahip olan “Hayalet” rolü. |
| Sistem Yöneticisi (Admin) | Uygulamanın genel işleyişini denetleyen yönetici rolüdür. Kullanıcı hesapları, raporlar, sistem logları üzerinde yetkilidir. | Uygulamanın admin paneline erişimi olan teknik yönetici. |

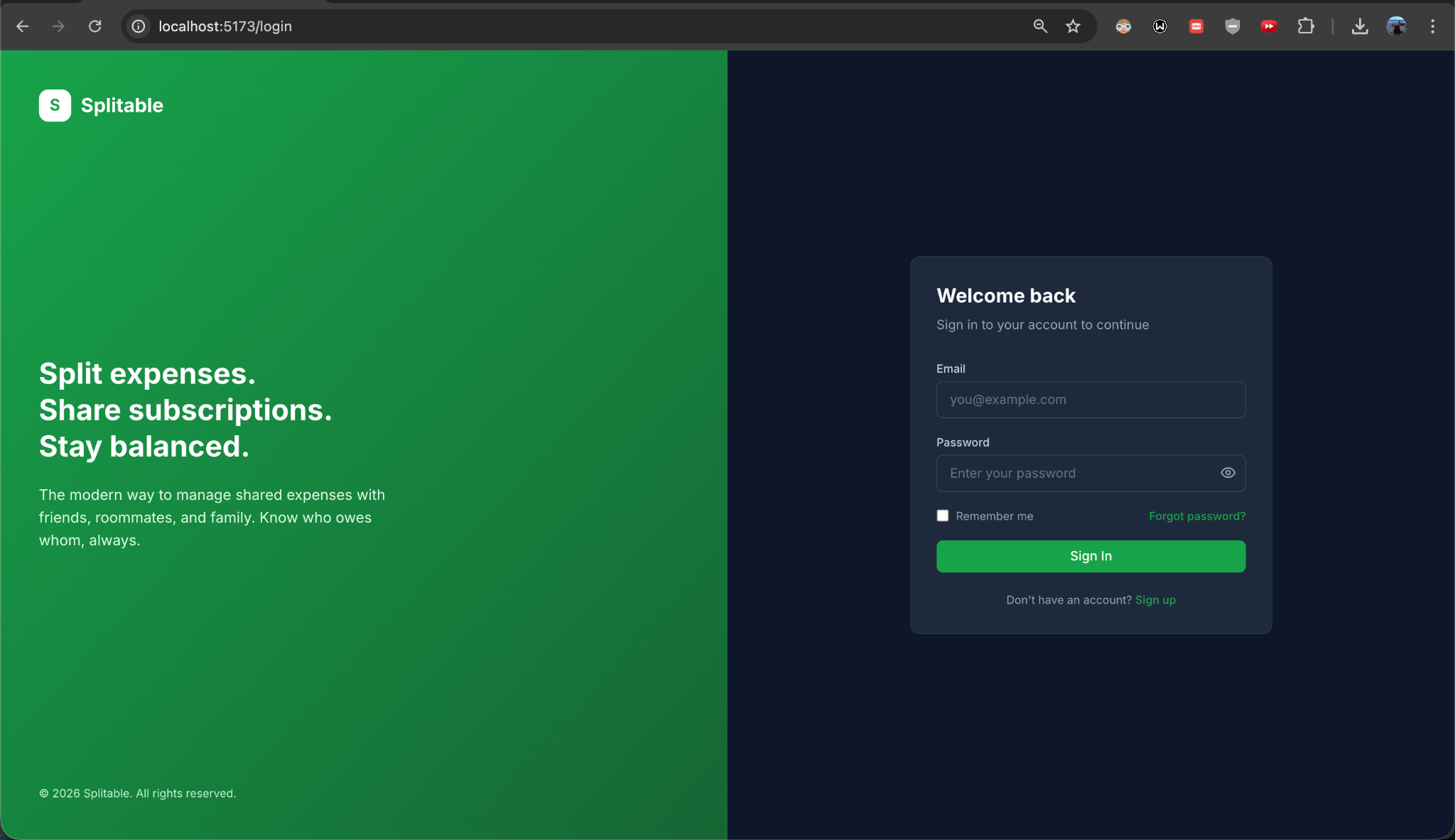
## 2.3. UI Betimleme

### 2.3.1. Web Tasarım

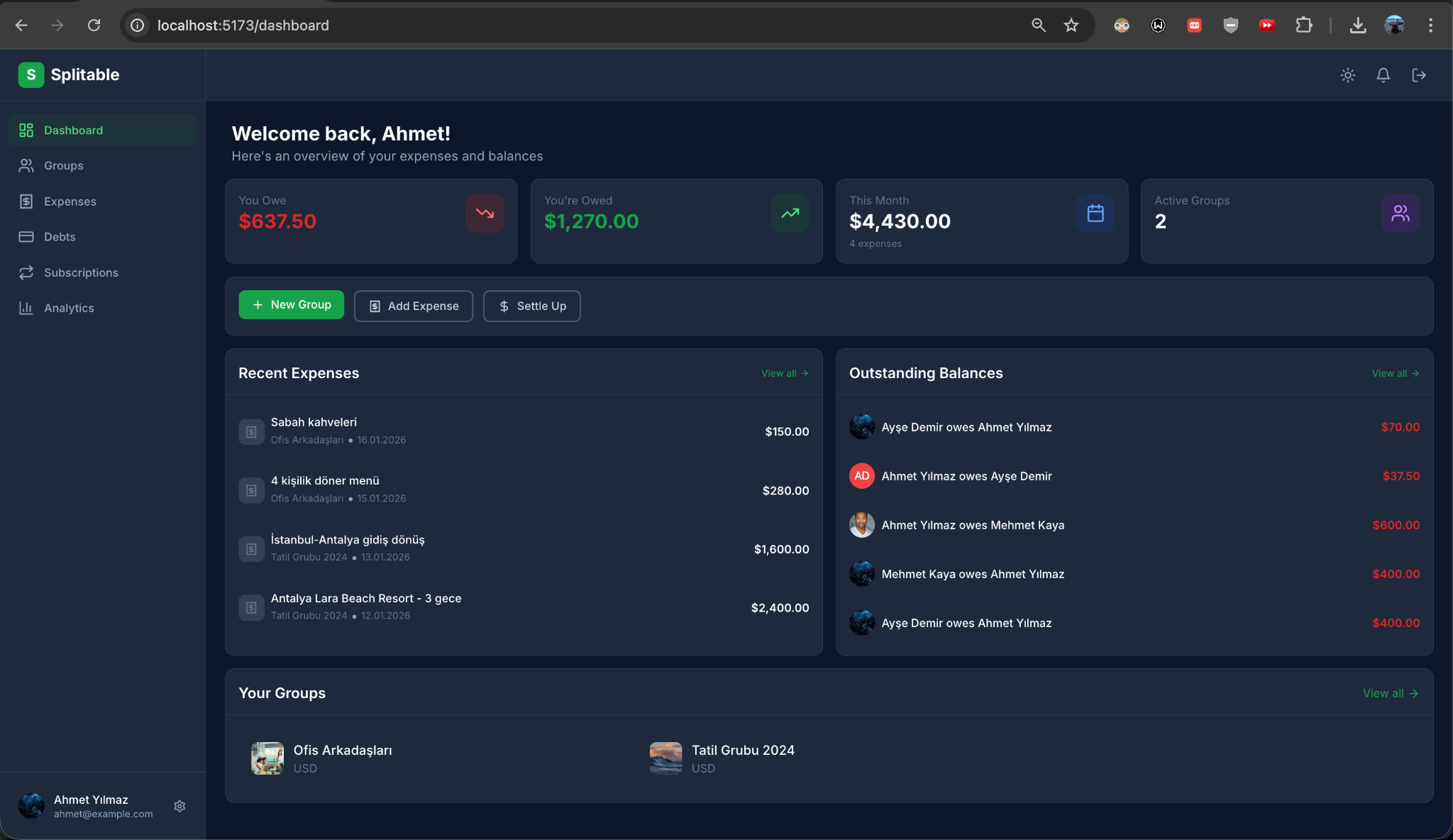
Aşağıda web arayüzüne ait örnek ekran görüntüleri verilmiştir.



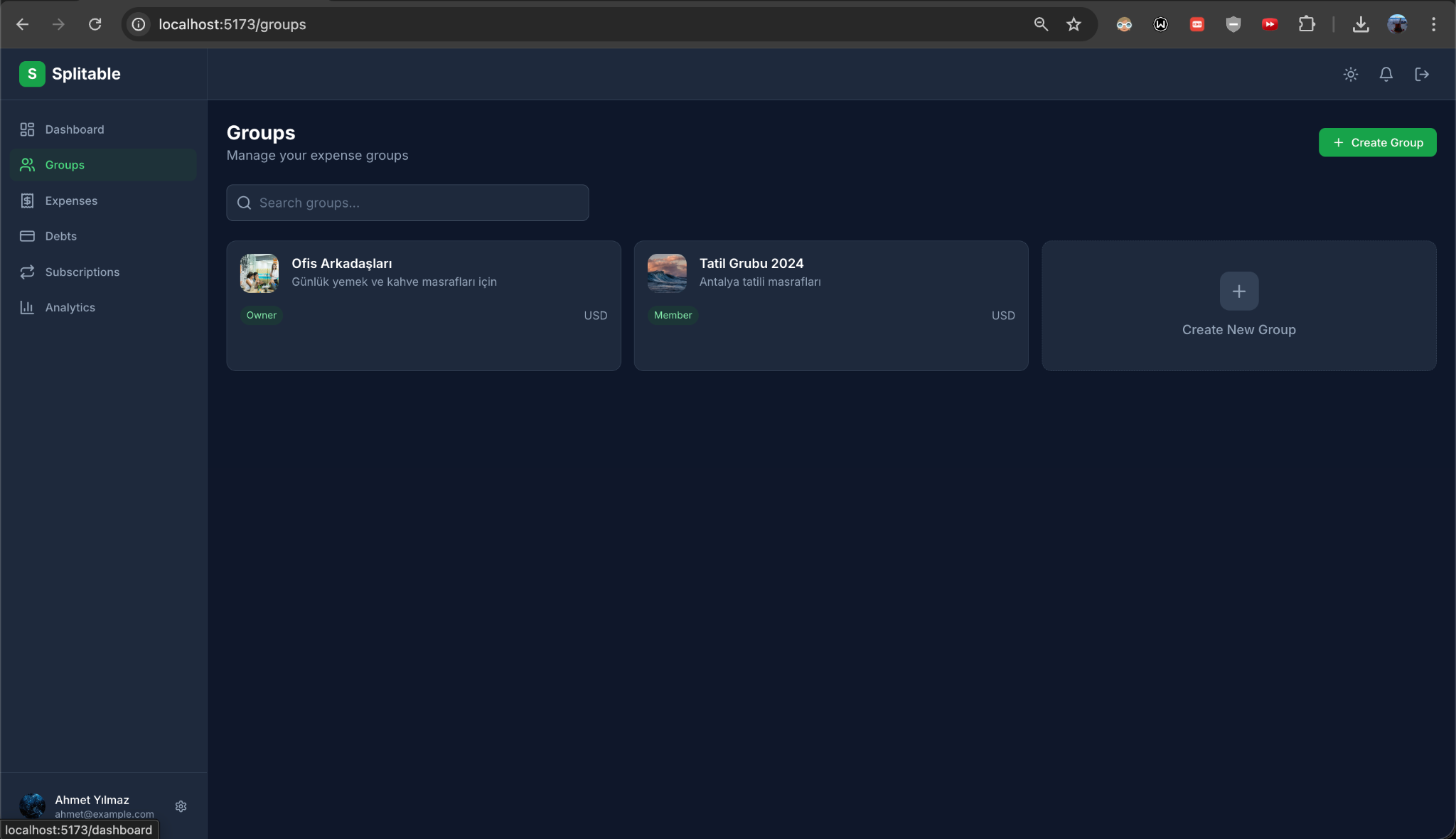
Şekil 1. Kayıt ol ekranı



Şekil 2. Giriş yap ekranı



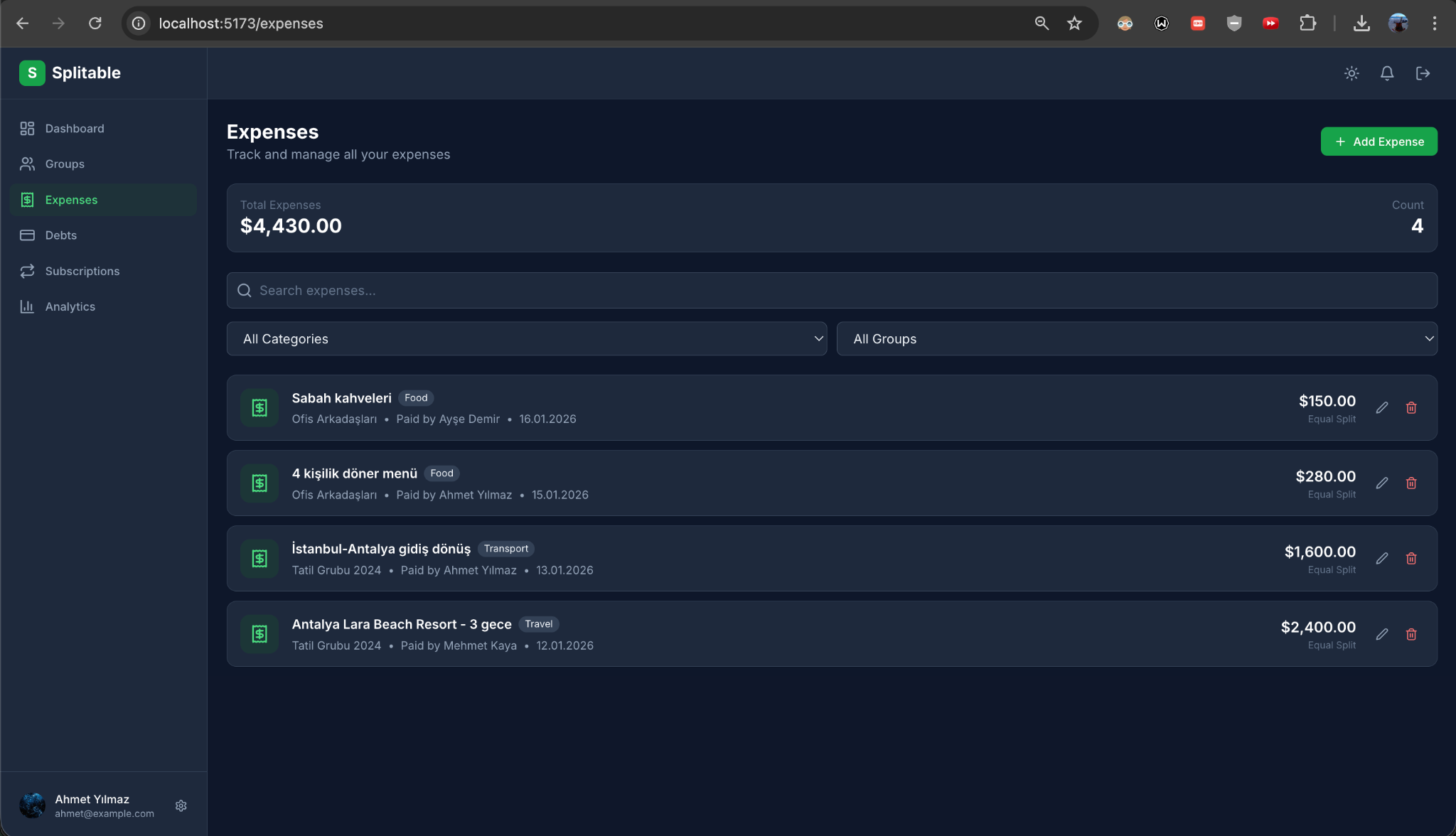
Şekil 3. Dashboard ekranı



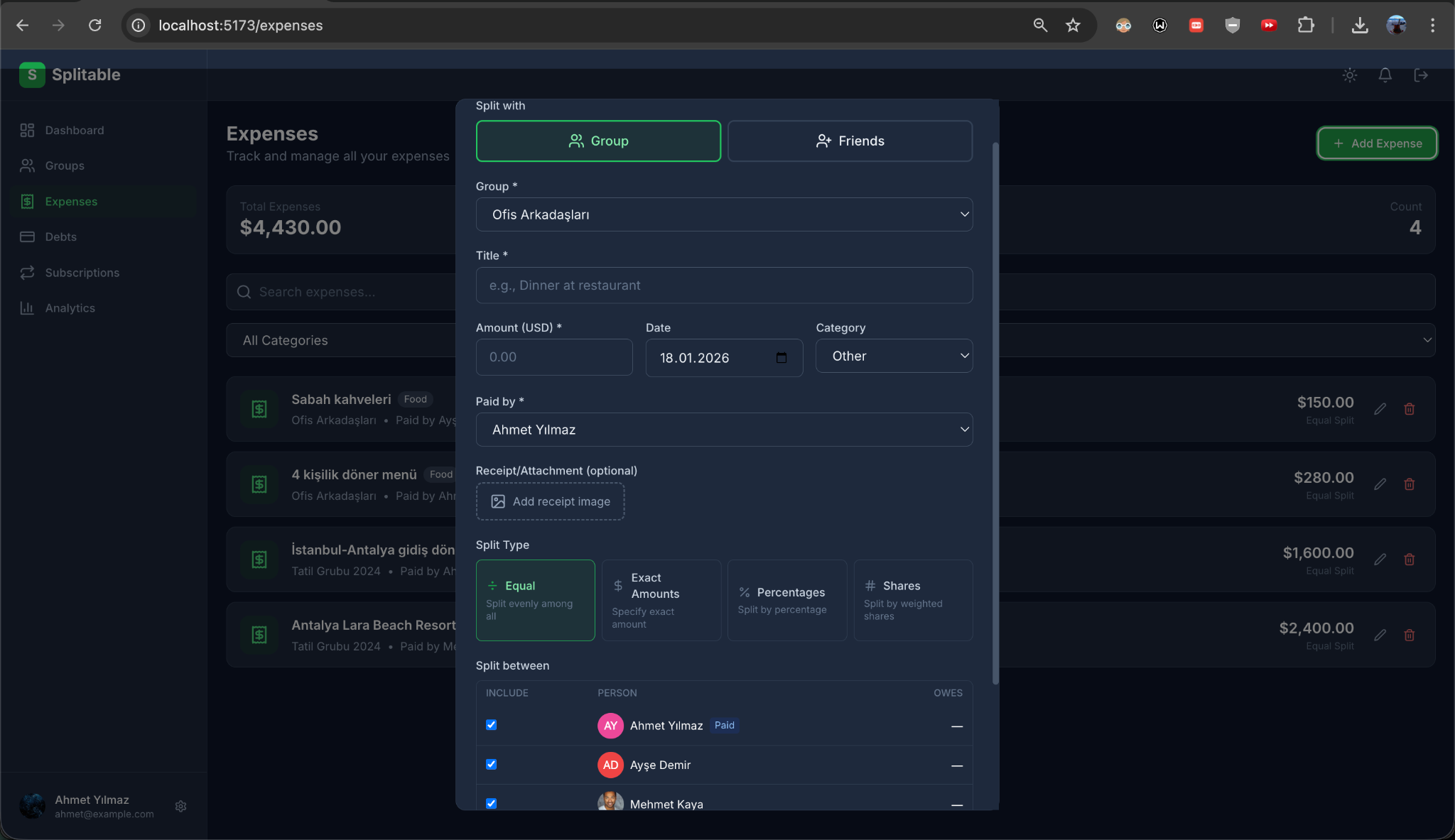
Şekil 4. Gruplar ekranı



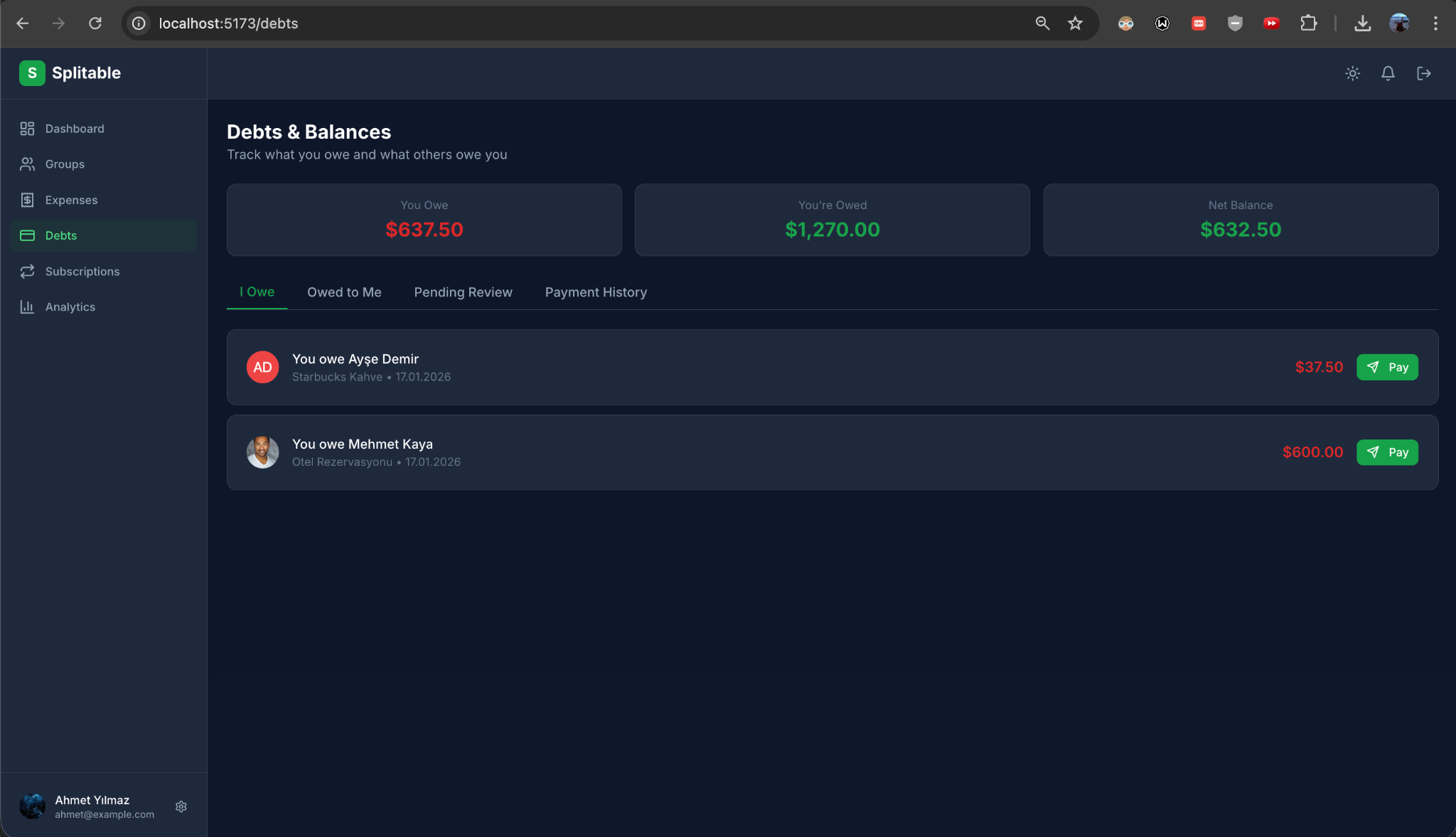
Şekil 5. Yeni grup oluşturma ekranı



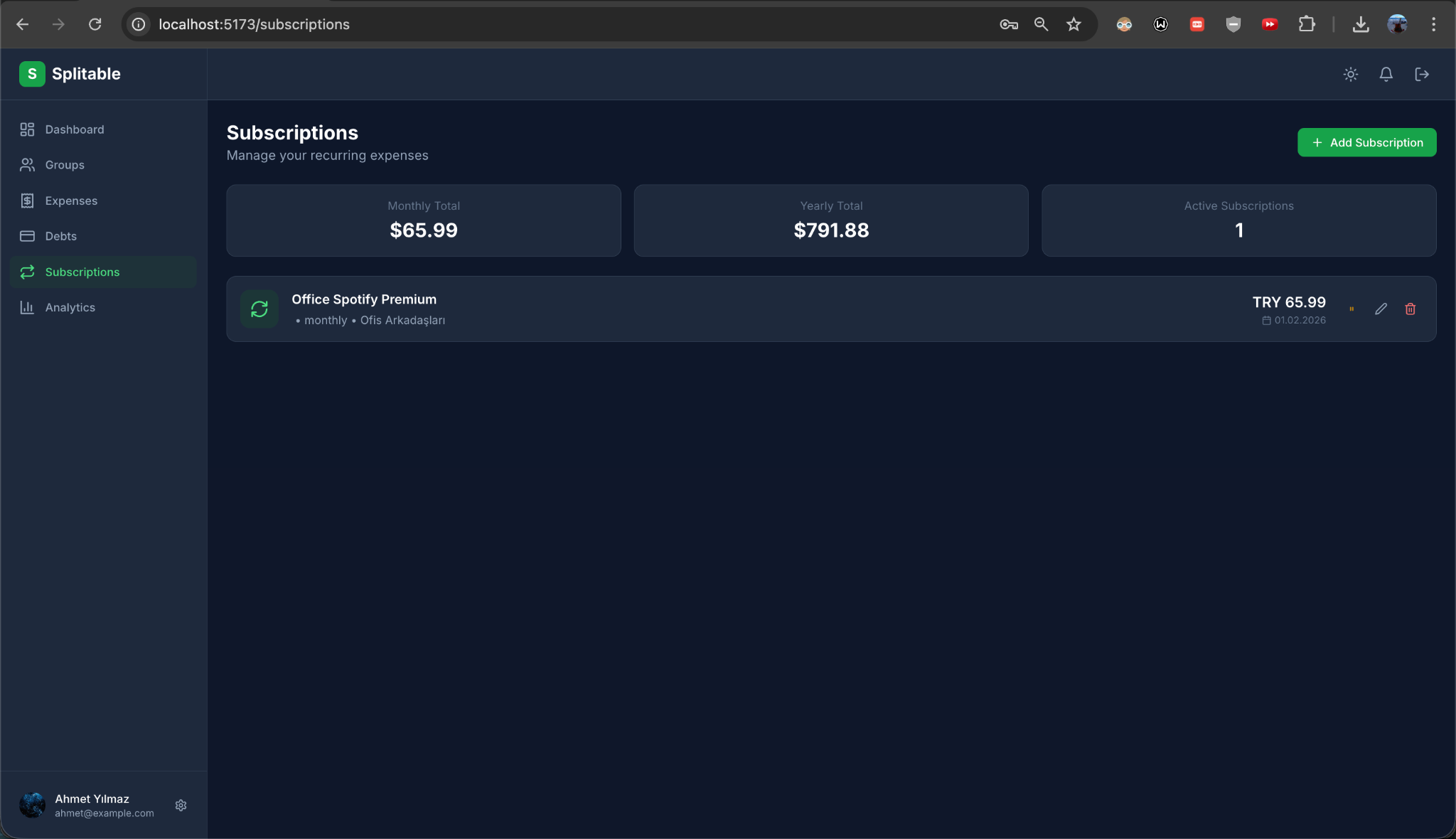
Şekil 6. Giderler ekranı



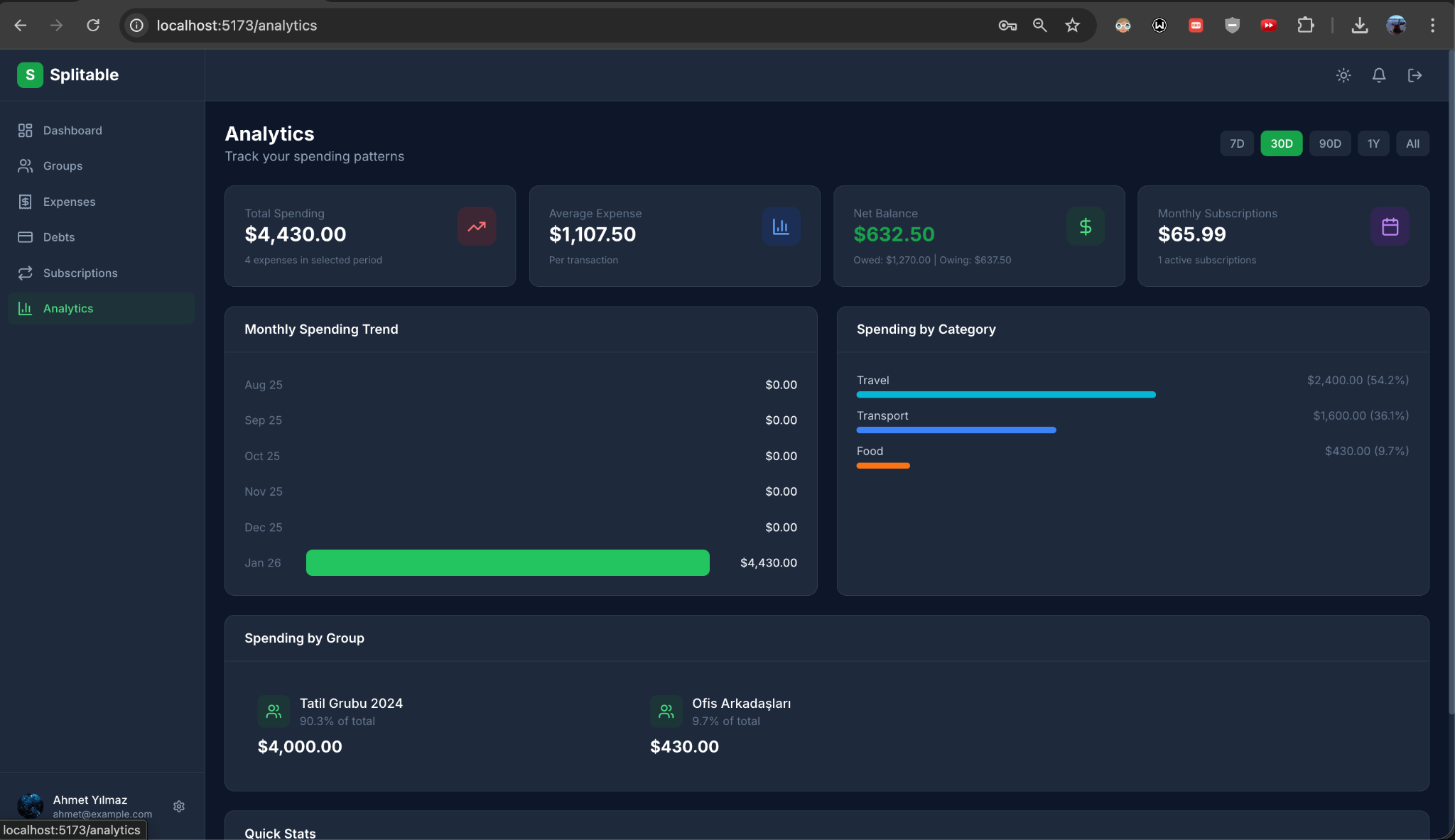
Şekil 7. Gider ekleme ekranı



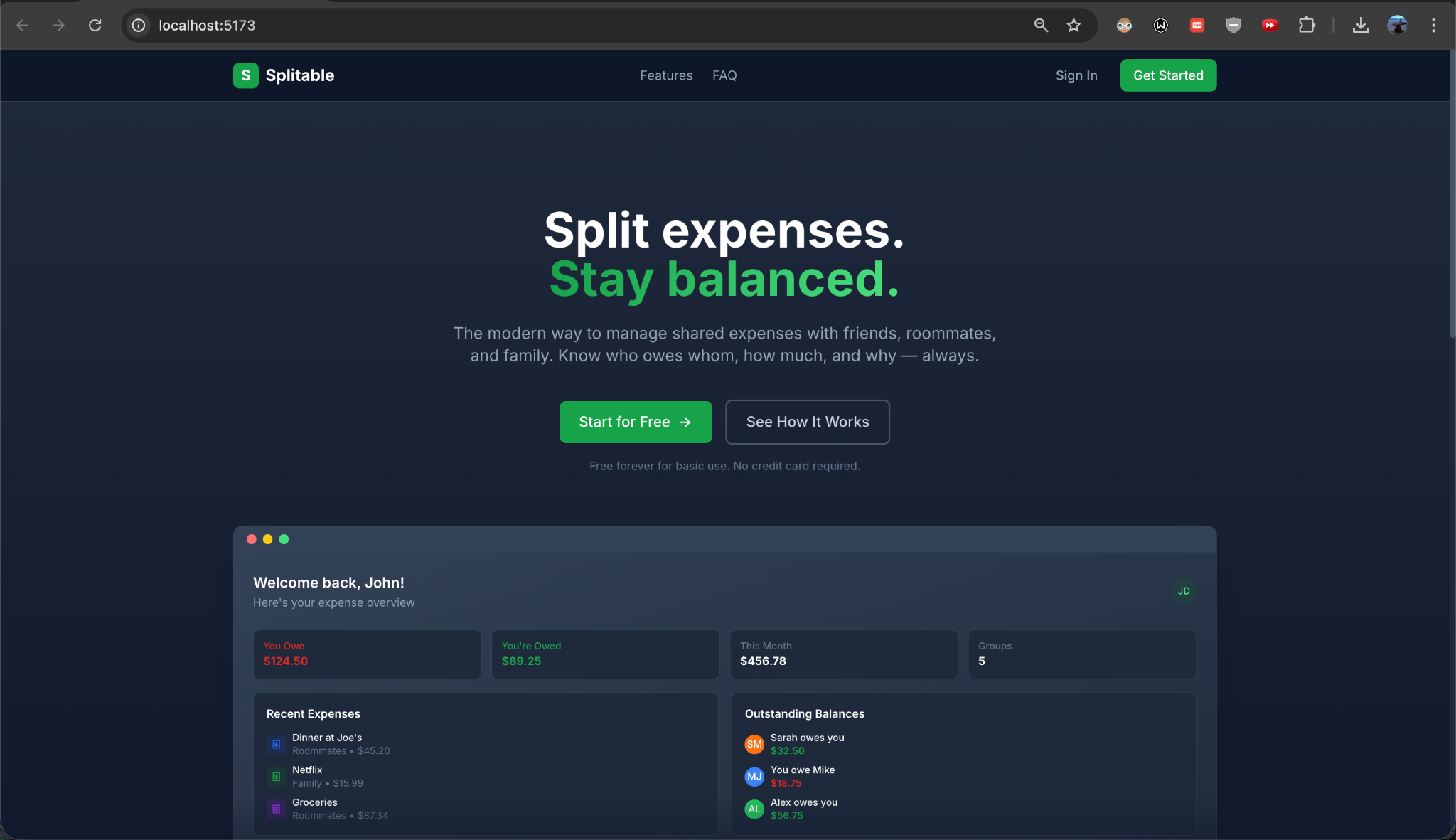
Şekil 8. Borçlar ve bakiyeler ekranı



Şekil 9. Abonelikler ekranı



Şekil 10. Analitik ekranı



Şekil 11. Landing sayfası

Ana ekranda uygulamayı tanıtan bir reklam ve giriş/kayıt butonu olacak. Gerisi responsive bir şekilde web uygulamayla eşleşen bir tasarım.

# 3. SİSTEM TASARIMI VE TEKNOLOJİLER

## 3.1. kullanılan Teknolojiler

### 3.1.1. Web Uygulama Frontend (React + Tailwind CSS)

React.js:

* Web sürümünde kullanıcıların aynı veriye tarayıcıdan erişebilmesi sağlanmıştır.
* Component tabanlı yapı ile web arayüzü arasında kod yeniden kullanılabilirliği.
* React Router ile sayfa yönlendirmeleri ve durum yönetimi.
* Axios veya React Query ile API bağlantısı.

Tailwind CSS:

Web arayüzünde görsel tasarım için Tailwind CSS kullanılmıştır.

* Hızlı prototipleme ve tutarlı tasarım sağlar.
* Responsive (mobil-öncelikli) tasarımı kolaylaştırır.
* Renk paleti, aralıklar, tipografi ve koyu/açık tema gibi tasarım token’ları projeye özel tanımlanabilir.
* Web tarafındaki Tailwind tasarım dili, mobil taraftaki NativeWind ile büyük oranda uyumlu görsel dil sağlayacaktır.

### 3.1.2. Backend (Node.js + Express.js)

Node.js + Express.js:

* Uygulamanın API katmanı Node.js üzerinde Express.js framework’ü kullanılarak oluşturulmuştur.
* RESTful API yapısı ile web istemcilerine veri sağlanmıştır.
* JWT tabanlı kimlik doğrulama (access + refresh token).
* Hata yönetimi ve global middleware yapısı (auth, validation, rate limit).
* Geliştirici dostu modüler mimari: /routes, /controllers, /services, /models.

### 3.1.3. Veri Tabanı (PostgreSQL)

PostgreSQL:

* Güçlü veri bütünlüğü, ACID uyumluluğu ve ilişkisel yapı.
* JSON veri tipi desteği sayesinde esnek modelleme (örneğin: borç kalemleri, gruplar).
* Yüksek ölçeklenebilirlik ve güvenilirlik.

1. İstemci Katmanı (Client Layer)

* Kullanıcılar bu arayüzler üzerinden borç ekleme, ödeme gönderme, grup oluşturma gibi işlemler yapar.
* Tüm kullanıcı etkileşimleri API çağrıları aracılığıyla sunucu katmanına iletilir.

2. Sunucu Katmanı (Application / API Layer)

* Node.js + Express.js üzerine kuruludur.
* Uygulamanın tüm iş mantığı burada bulunur:
  + - Kimlik doğrulama (JWT)
    - Borç ekleme, güncelleme, silme
    - Ödeme onay akışları
    - Grup yönetimi ve yetkilendirme
* API katmanı, istemcilerden gelen istekleri işler, veritabanına erişir ve uygun yanıtları döner.

3. Veri Katmanı (Database + Storage Layer)

* Tüm metinsel veriler PostgreSQL üzerinde saklanır.
* ORM katmanı (Drizzle), veri tabanı işlemlerini güvenli ve yönetilebilir hale getirir.

## 3.3. Database Tasarımı

### 3.3.1. ENUM’lar

**system\_role**

|  |  |  |
| --- | --- | --- |
| **Alan** | **Tip** | **Açıklama** |
| value | ENUM | admin, user |

**split\_type**

|  |  |  |
| --- | --- | --- |
| **Alan** | **Tip** | **Açıklama** |
| value | ENUM | EQUAL, UNEQUAL, PERCENT, SHARE |

**debt\_status**

|  |  |  |
| --- | --- | --- |
| **Alan** | **Tip** | **Açıklama** |
| value | ENUM | open, partially\_paid, paid, disputed, cancelled |

**payment\_status**

|  |  |  |
| --- | --- | --- |
| **Alan** | **Tip** | **Açıklama** |
| value | ENUM | submitted, approved, rejected, reverted |

**subscription\_period**

|  |  |  |
| --- | --- | --- |
| **Alan** | **Tip** | **Açıklama** |
| value | ENUM | weekly, monthly, yearly |

**attachment\_type**

|  |  |  |
| --- | --- | --- |
| **Alan** | **Tip** | **Açıklama** |
| value | ENUM | receipt, payment\_proof, avatar, group\_photo, other |

**notif\_channel**

|  |  |  |
| --- | --- | --- |
| **Alan** | **Tip** | **Açıklama** |
| value | ENUM | push, email, in\_app |

### 3.3.2. Tablolar

**users**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| user\_id | BIGINT (PK) | Kullanıcı id’si |
| name | VARCHAR(120) | Ad ve Soyad |
| email | VARCHAR(254) (UNIQUE) | E-mail adresi |
| password\_hash | VARCHAR(255) | Şifre (bcrypt ile hashlenmiş) |
| default\_currency | CHAR(3) FK currencies | Kullanıcının varsayılan para birimi. |
| profile\_image\_url | TEXT | Profil görseli URL veya path. |
| system\_role | system\_role | Sistem çapında rol (user/admin). |
| created\_at | TIMESTAMPTZ | Oluşturulma zamanı. |
| updated\_at | TIMESTAMPTZ | Güncellenme tarihi. |

**user\_ibans**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| iban\_id | BIGINT (PK) | Kimlik. |
| user\_id | BIGINT FK users | Sahip kullanıcı. |
| iban | VARCHAR(34) | IBAN değeri. |
| label | VARCHAR(50) | Etiket (örn. “Ziraat”). |
| is\_default | BOOLEAN | Varsayılan mı. |
| created\_at | TIMESTAMPTZ | Oluşturulma zamanı. |

**currencies**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| code | CHAR(3) (PK) | ISO 4217 kodu (örn. TRY). |
| name | VARCHAR(50) | Ad (Türk Lirası). |
| symbol | VARCHAR(8) | Sembol (₺). |
| decimals | SMALLINT | Ondalık basamak. |

**exchange\_rates**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| base\_code | CHAR(3) FK currencies | Baz para. |
| quote\_code | CHAR(3) FK currencies | Karşı para. |
| rate | NUMERIC(18,8) | Kur değeri. |
| as\_of | TIMESTAMPTZ | Geçerlilik anı. (PK: base\_code, quote\_code, as\_of) |

**groups**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| group\_id | BIGINT (PK) | Kimlik |
| name | VARCHAR(120) | Grup adı. |
| description | TEXT | Açıklama. |
| photo\_url | TEXT | Grup görseli url veya path. |
| preferred\_currency | CHAR(3) FK currencies | Grup para birimi. |
| creator\_id | BIGINT FK users | Grubu oluşturan. |
| archived\_at | TIMESTAMPTZ | Arşivlenme zamanı. |
| created\_at | TIMESTAMPTZ | Oluşturulma. |
| updated\_at | TIMESTAMPTZ | Güncellenme tarihi. |

**group\_roles**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| role\_id | BIGINT (PK) | Kimlik. |
| group\_id | BIGINT FK groups | İlgili grup. |
| name | VARCHAR(50) | Rol adı (owner/admin/member/viewer veya özel). |
| description | VARCHAR(200) | Açıklama. |
| permissions | JSONB | İzinler (örn. can\_add\_expense). |
| built\_in | BOOLEAN | Sistem varsayılanı mı? |

**group\_members**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| group\_id | BIGINT FK groups | Grup. (PK: group\_id, user\_id) |
| user\_id | BIGINT FK users | Üye kullanıcı. |
| role\_id | BIGINT FK group\_roles | Üyenin grup rolü. |
| joined\_at | TIMESTAMPTZ | Katılma zamanı. |

**categories**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| category\_id | BIGINT (PK) | Kimlik. |
| name | VARCHAR(50) | Kategori adı. |
| icon | VARCHAR(60) | Simge adı/anahtarı. |
| is\_system | BOOLEAN | Sistem varsayılanı mı? |
| created\_by | BIGINT FK users (nullable) | Oluşturan kullanıcı. |

**attachments**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| attachment\_id | BIGINT (PK) | Kimlik. |
| owner\_user\_id | BIGINT FK users (nullable) | Sahip kullanıcı. |
| group\_id | BIGINT FK groups (nullable) | İlgili grup. |
| type | attachment\_type | Tür (fiş/dekont vb.). |
| mime\_type | VARCHAR(100) | İçerik tipi. |
| storage\_provider | VARCHAR(30) | Depo (örn. cloudinary). |
| url | TEXT | Dosya URL. |
| size\_bytes | BIGINT | Boyut (B). |
| width | INT | Genişlik (görsel). |
| height | INT | Yükseklik (görsel). |
| created\_at | TIMESTAMPTZ | Yüklenme tarihi. |

**expenses**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| expense\_id | BIGINT (PK) | Kimlik. |
| group\_id | BIGINT FK groups (nullable) | Grup harcaması ise. |
| created\_by | BIGINT FK users | Harcamayı ekleyen. |
| payer\_id | BIGINT FK users (nullable) | Ödemeyi yapan kişi. |
| title | VARCHAR(140) | Başlık. |
| description | TEXT | Açıklama. |
| category\_id | BIGINT FK categories (nullable) | Kategori. |
| total\_amount | NUMERIC(12,2) | Toplam tutar. |
| currency\_code | CHAR(3) FK currencies | Para birimi. |
| expense\_date | DATE | Harcama tarihi. |
| due\_date | DATE | Son ödeme tarihi (opsiyonel). |
| split\_method | split\_type | Bölüşüm yöntemi. |
| meta | JSONB | Ek veriler. |
| created\_at | TIMESTAMPTZ | Oluşturulma tarihi. |
| updated\_at | TIMESTAMPTZ | Güncellenme tarihi. |

**expense\_attachments**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| expense\_id | BIGINT FK expenses | Harcama. (PK: expense\_id, attachment\_id) |
| attachment\_id | BIGINT FK attachments | Ek dosya. |

**debts**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| debt\_id | BIGINT (PK) | Kimlik. |
| expense\_id | BIGINT FK expenses (nullable) | Harcamadan doğan borçsa. |
| group\_id | BIGINT FK groups (nullable) | Grup bağlama. |
| creditor\_id | BIGINT FK users | Alacaklı. |
| debtor\_id | BIGINT FK users | Borçlu. |
| title | VARCHAR(140) | Başlık. |
| description | TEXT | Açıklama. |
| amount | NUMERIC(12,2) | Borç tutarı. |
| currency\_code | CHAR(3) FK currencies | Para birimi. |
| due\_date | DATE | Vade. |
| status | debt\_status | Durum. |
| created\_at | TIMESTAMPTZ | Oluşturulma. |
| closed\_at | TIMESTAMPTZ | Kapanma (ödenince). |
| subscription\_cycle\_id | BIGINT FK subscription\_cycles (nullable) | Abonelikten üretildiyse. |

**payments**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| payment\_id | BIGINT (PK) | Kimlik. |
| debt\_id | BIGINT FK debts | İlgili borç. |
| payer\_id | BIGINT FK users | Ödeyen (genelde borçlu). |
| receiver\_id | BIGINT FK users (nullable) | Alıcı (genelde alacaklı). |
| amount | NUMERIC(12,2) | Ödeme tutarı. |
| currency\_code | CHAR(3) FK currencies | Para birimi. |
| method | VARCHAR(40) | Yöntem (bank\_transfer/cash/card/… veya “manual”). |
| proof\_attachment\_id | BIGINT FK attachments (nullable) | Dekont/kanıt. |
| note | VARCHAR(240) | Not. |
| status | payment\_status | Onay durumu. |
| reviewed\_by | BIGINT FK users (nullable) | Onaylayan/red eden. |
| reviewed\_at | TIMESTAMPTZ | İnceleme zamanı. |
| created\_at | TIMESTAMPTZ | Oluşturulma. |

**debt\_status\_history**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| history\_id | BIGINT (PK) | Kimlik. |
| debt\_id | BIGINT FK debts | İlgili borç. |
| from\_status | debt\_status (nullable) | Önceki durum. |
| to\_status | debt\_status | Yeni durum. |
| changed\_by | BIGINT FK users (nullable) | Değiştiren kullanıcı. |
| reason | VARCHAR(240) | Gerekçe/not. |
| created\_at | TIMESTAMPTZ | Zaman damgası. |

**subscriptions**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| subscription\_id | BIGINT (PK) | Kimlik. |
| group\_id | BIGINT FK groups (nullable) | Grup bağlamı. |
| owner\_user\_id | BIGINT FK users | Abonelik sahibi. |
| service\_name | VARCHAR(100) | Servis (Netflix, Spotify). |
| plan\_name | VARCHAR(100) | Plan adı (ops.). |
| amount | NUMERIC(12,2) | Dönem toplam tutarı. |
| currency\_code | CHAR(3) FK currencies | Para birimi. |
| period | subscription\_period | Periyot (weekly/monthly/yearly). |
| billing\_day | SMALLINT | Faturalama günü (1–31). |
| start\_date | DATE | Başlangıç. |
| end\_date | DATE | Bitiş (ops.). |
| status | VARCHAR(20) | active/paused/cancelled. |
| default\_payer\_id | BIGINT FK users (nullable) | Varsayılan ödeyen. |
| created\_at | TIMESTAMPTZ | Oluşturulma. |

**subscription\_participants**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| subscription\_id | BIGINT FK subscriptions | Abonelik. (PK: subscription\_id, user\_id) |
| user\_id | BIGINT FK users | Katılımcı. |
| share\_type | split\_type | Paylaşım tipi. |
| share\_value | NUMERIC(10,2) | PERCENT: %; UNEQUAL: tutar; SHARE: ağırlık. |
| is\_active | BOOLEAN | Aktif katılımcı mı. |

**subscription\_cycles**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| cycle\_id | BIGINT (PK) | Kimlik. |
| subscription\_id | BIGINT FK subscriptions | Abonelik. |
| cycle\_start | DATE | Dönem başlangıcı. |
| cycle\_end | DATE | Dönem bitişi. |
| invoice\_date | DATE | Fatura tarihi. |
| total\_amount | NUMERIC(12,2) | Dönem tutarı. |
| status | VARCHAR(20) | open/billed/cancelled. |
| generated\_at | TIMESTAMPTZ | Üretilme zamanı. |

**group\_invites**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| invite\_id | BIGINT (PK) | Kimlik. |
| group\_id | BIGINT FK groups | Davet edilen grup. |
| invited\_by | BIGINT FK users | Davet eden. |
| role\_id | BIGINT FK group\_roles | Önerilen rol. |
| token | UUID (UNIQUE) | Davet kodu. |
| expires\_at | TIMESTAMPTZ | Son kullanım. |
| accepted\_by | BIGINT FK users (nullable) | Kabul eden kullanıcı. |
| accepted\_at | TIMESTAMPTZ | Kabul zamanı. |
| status | VARCHAR(20) | pending/accepted/revoked/expired. |
| created\_at | TIMESTAMPTZ | Oluşturulma. |

**notifications**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| notification\_id | BIGINT (PK) | Kimlik. |
| user\_id | BIGINT FK users | Hedef kullanıcı. |
| channel | notif\_channel | Bildirim kanalı. |
| type | VARCHAR(80) | Olay tipi (örn. debt.created). |
| payload | JSONB | İçerik/gövde. |
| read\_at | TIMESTAMPTZ | Okunma zamanı. |
| created\_at | TIMESTAMPTZ | Oluşturulma. |

**activity\_logs**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| log\_id | BIGINT (PK) | Kimlik. |
| group\_id | BIGINT FK groups (nullable) | Grup bağlamı. |
| actor\_id | BIGINT FK users (nullable) | Eylemi yapan. |
| type | VARCHAR(80) | Olay türü. |
| ref | JSONB | Referanslar (örn. {expense\_id:123}). |
| message | TEXT | Metinsel açıklama. |
| created\_at | TIMESTAMPTZ | Zaman damgası. |

**email\_verifications**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| user\_id | BIGINT (PK) FK users | Kullanıcı |
| token | UUID (UNIQUE) | Doğrulama kodu. |
| sent\_at | TIMESTAMPTZ | Gönderim. |
| verified\_at | TIMESTAMPTZ | Doğrulanma. |

**password\_resets**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| reset\_id | BIGINT (PK) | Kimlik. |
| user\_id | BIGINT FK users | Kullanıcı. |
| token | UUID (UNIQUE) | Sıfırlama token’ı. |
| created\_at | TIMESTAMPTZ | Oluşturulma. |
| used\_at | TIMESTAMPTZ | Kullanılma. |
| expires\_at | TIMESTAMPTZ | Son kullanım. |

**sessions**

|  |  |  |
| --- | --- | --- |
| Alan | Tip | Açıklama |
| session\_id | BIGINT (PK) | Kimlik. |
| user\_id | BIGINT FK users | Oturum sahibi. |
| refresh\_token | TEXT (UNIQUE) | Yenileme belirteci. |
| user\_agent | TEXT | İstemci bilgisi. |
| ip\_address | TEXT | IP adresi. |
| created\_at | TIMESTAMPTZ | Oluşturulma. |
| expires\_at | TIMESTAMPTZ | Bitiş zamanı. |

## 3.4. Güvenlik

### 3.4.1. Kapsam, Varsayımlar ve Olası Tehdit Modeli

* Varlıklar: Kullanıcı hesabı/verileri (PII), grup ve borç/ödeme kayıtları, abonelikler, fiş/dekont görselleri, erişim belirteçleri (JWT), e-posta doğrulama/parola sıfırlama token’ları.
* Tehditler (STRIDE kısaltmasıyla örnek):
  + Spoofing: Hesap ele geçirme (zayıf parola, kimlik avı, token çalınması).
  + Tampering: İstek manipülasyonu (IDOR), dosya türü/ içerik manipülasyonu.
  + Repudiation: İnkâr—işlem ve onay akışında iz sürülememesi.
  + Information Disclosure: PII/fiş görseli sızıntısı, aşırı veri döndürme.
  + DoS: API rate limit yokluğu, pahalı sorgular.
  + Elevation of Privilege: Grup rolü atlama, yetki kontrollerinin atlanması.
* Saldırı yüzeyi: Web istemci, public API uçları, dosya upload uçları, e-posta akışları (verify/reset), push bildirim entegrasyonları.

### 3.4.2. Veri Sınıflandırma

|  |  |  |  |
| --- | --- | --- | --- |
| Veri | Örnek | Sınıf | Koruma |
| Kimlik bilgisi | e-mail, ad-soyad | Kişisel | Erişim kontrolü |
| Şifreler | password\_hash | Hassas | bcrypt ile güçlü hash |
| Finansal veri | borç/ödeme verileri | Hassas | Erişim kontrolü |
| Dosyalar | fiş, dekont, görseller | Hassas | İmzalı URL, içerik tipi whitelist (pdf, jpeg, png) |
| Token’lar | JWT, tokenlar | Kritik | Kısa ömür, rotation, sızıntıya karşı izleme |

### 3.4.3. Güvenlik Mimarisi

* Uçtan uca TLS 1.3; HSTS, HTTPS, güvenli cookie’ler.
* JWT (access) + Refresh Token Rotation: Access kısa ömür (≤15 dk), refresh rotation ve çalınma algısı; çalıntı refresh tespitinde oturum kapatma.
* RBAC (Rol Tabanlı Erişim Kontrolü): group\_roles + group\_members + permissions(JSONB) ile en az ayrıcalık. Her API’da “kullanıcı bu kaynağı görmeye yetkili mi?” kontrolü.
* Katmanlı doğrulama: API seviyesinde şema validasyonu (tüm body/query/path), dosya upload whitelist (pdf, jpeg, png…), boyut sınırı, MIME/doğrulama.

### 3.4.4. Özetle Yapılacak Kontroller

Auth & Sessions:

* Parolalar bcrypt (cost ≥ 12), parola politikası aktif.
* Access JWT ≤ 15 dk, refresh rotation + çalınma tespiti.
* Email verify/reset token kısa ömür, tek kullanımlık, rate-limit.

Yetkilendirme:

* Tüm kritik uçlarda sahiplik + rol izni kontrolü.
* Listelemelerde sadece yetkili grup/kayıtlar dönüyor.
* Ödeme onayı ve geri alma yalnız yetkili rolde.

API & Girdi:

* Şema validasyonu; enum/uzunluk/sayı aralıkları.
* Rate-limit, güvenli CORS, hata mesajları steril.
* IDOR testleri: farklı kullanıcı/ grup ile denendi ve geçilmedi.

Dosya:

* MIME + uzantı + magic-number doğrulama; boyut sınırı.
* İmzalı, süreli URL; public listeleme kapalı.
* EXIF temizliği, zararlı içerik tespiti.

Veri/DB:

* Parametreli sorgular; DB kullanıcı yetkileri kısıtlı.
* Yedekleme politikası ve geri yükleme testi mevcut.
* Loglarda PII maskeleniyor.

Client:

* Mobilde token’lar güvenli depolama.
* Web’de CSP, HSTS, Safe cookies, XSS korumaları.
* Sırlar istemci paketinde yok.

# 4. PROJE YÖNETİMİ VE GELİŞTİRME SÜRECİ

## 4.1. Proje Planı ve Görev Dağılımı

### 4.1.1. Proje Ekibi

|  |  |
| --- | --- |
| Üyeler | Sorumluluklar |
| Kerem YILMAZ | • Güvenlik (JWT, RBAC/rol kontrolleri, rate-limit, veri doğrulama) • Abonelik sistemi (döngü/borç üretimi) • Analitik & istatistik çıktıları • Bildirim sistemi geliştirmesine destek (push/e‑posta/in‑app) • Arama/filtreleme/sıralama • Dışa/İçe aktarma (CSV/PDF, import) • Admin paneli • Grup & rol yönetimi (üyelik, davet, yetkilendirme) |
| Burak ASARCIKLI | • Web UI (dashboard, listeleme sayfaları, responsive tasarım) • Veri tabanı ve ORM (PostgreSQL şeması, migration’lar, ilişkiler) • Kullanıcı kayıt/giriş ve hesap yönetimi (doğrulama, parola sıfırlama) • Borç & ödeme sistemi (kısmi ödeme, talep, onay akışı) • Bildirim sistemi entegrasyonu (Kerem ile birlikte) |

### 4.1.2. Görev Dağılımı

|  |  |  |  |
| --- | --- | --- | --- |
| Özellik | Açıklama | K.Y. | B.A. |
| Web UI | Web arayüzü, dashboard, listeleme sayfaları ve responsive tasarımın oluşturulması. | ✓ | ✓ |
| Veri Tabanı ve ORM | PostgreSQL şeması, tablolar, ilişkiler, migration’lar ve ORM yapılandırmasının geliştirilmesi. | ✓ | ✓ |
| Kullanıcılar ve Kayıt/Giriş | Kayıt, giriş, JWT doğrulama, parola sıfırlama ve hesap yönetimi süreçlerinin yazılması. |  | ✓ |
| Borç ve Ödeme Sistemi | Borç ekleme, kısmi ödeme, ödeme talebi, onay akışı ve borç durumlarının yönetilmesi. |  | ✓ |
| Grup ve Rol Yönetimi | Grup oluşturma, üye daveti, rol atama/yaratma ve yetki kontrollerinin uygulanması. | ✓ |  |
| Abonelik Sistemi | Aylık döngü oluşturma, katılımcı payları ve otomatik borç üretme mekanizmasının geliştirilmesi. | ✓ |  |
| Arama, Filtreleme ve Sıralama | Kişi, tarih, kategori ve tutara göre arama, filtreleme ve liste sıralama fonksiyonları. |  | ✓ |
| Analitik & İstatistik | Harcama grafikleri, kategori dağılımları ve kullanıcı bazlı finansal özetlerin oluşturulması. | ✓ |  |
| Güvenlik | JWT güvenliği, rate limit, veri doğrulama, rol kontrolü ve dosya güvenliği uygulamaları. | ✓ |  |
| Dışa/İçe Aktarma | CSV/PDF çıktılar, Splitwise/Tricount içe aktarma ve veri dışa aktarma özellikleri. |  | ✓ |
| Bildirim Sistemi | Push bildirimler, e-posta bildirimleri ve uygulama içi bildirimlerin yönetilmesi. | ✓ | ✓ |
| Admin Paneli | Kullanıcı, grup, log, istatistik ve sistem yönetimine yönelik admin arayüzünün geliştirilmesi. |  | ✓ |

## 4.2. Geliştirme Yöntemi

Splitable projesi iki geliştirici tarafından yürütüldüğü için, geliştirme süreci basitleştirilmiş bir Agile yaklaşımı ile yönetilmiştir.

Her hafta başında kısa bir planlama yapılır ve o hafta kişilerin müsaitlik durumu incelenir. Hafta sonlarında geliştirilen özellikler test edilir.

# 5. GELECEK PLANLARI VE SONUÇ

## 5.1. Gelecek Planları

Splitable projesi temel bir borç–alacak takip uygulamasından daha fazlasını hedeflemektedir. Mevcut sürümde temel finans paylaşımı, grup yönetimi, abonelik takibi, dosya yükleme ve güvenlik katmanları bulunmakla birlikte; uzun vadede uygulamanın kullanıcı kitlesi ve kullanım alanı genişledikçe bazı geliştirmeler planlanmaktadır.

Doğrudan Ödeme İşlemleri:

* Ödeme Sistemleri ve Entegrasyonlar: Uygulama içinden doğrudan ödeme yapabilmek için entegrasyonlar.
* Ortak Abonelik Sistemi: Varsayılan abonelik sistemlerinde ortak bir sanal kart oluşturup bu kart üzerinden otomatik ödeme işlemi.

Otomatik Fiş/Dekont okuma:

* Fiş Okuma: Yüklenen fişten otomatik veri çıkarma. Kategori, tutar…
* Dekont Okuma: Dekontu otomatik okuyup doğrulayıp onaylama.

Offline Mod:

* Tek Kişilik Kullanım: Misafir kullanıcının kendi grubunu ve katılımcıları kendisi oluşturup işlemleri tek başına yapması.
* Offline Kullanıma Devam Edebilmek: Çevrim dışı bir şekilde online kurulmuş bir grup içerisinde işlemler yapabilmek ve çevrim içi ortama geçildiğinde bu işlemleri senkron edebilmek.

Gelişmiş Analizler:

* Abonelikler ve Harcamalar: Abonelikler ve harcamalar için yapılan analizlerden sonuç çıkarıp, çıkarılan verilerle akıllı öngörülerde bulunulması.

## 5.2. Sonuç ve Beklenen Katkılar

Splitable, bireylerin ve arkadaş gruplarının finansal ilişkilerini daha düzenli, şeffaf ve güvenli şekilde yönetmelerini sağlayan modern bir çözüm olarak tasarlanmıştır. Rakip uygulamalarda ücretli olan birçok ileri seviye özellik, Splitable’da temel birer fonksiyon olarak sunulmakta ve bu yönüyle uygulama daha erişilebilir bir deneyim sağlamaktadır.

Proje tamamlandığında:

* Ortak harcamalarını kolayca paylaşabilecek,
* Aboneliklerini düzenli takip edebilecek,
* Ödemelerini görsel olarak belgeleyebilecek,
* Rollere göre güvenli bir grup ortamında çalışabilecek,
* Harcama alışkanlıklarını grafikler ve raporlarla analiz edebilecek,
* Web ve mobil arasında kesintisiz bir deneyim yaşayabilecektir.

Splitable’ın geliştirilebilir, ölçeklenebilir ve güvenli mimarisi sayesinde proje ileride geniş topluluklar, öğrenci evleri, ekipler ve hatta küçük işletmeler tarafından bile kullanılabilecek bir finans paylaşım platformuna dönüşme potansiyeline sahiptir.

# 6. KAYNAKÇA

[3] React Documentation. https://react.dev/

[4] Tailwind CSS Documentation. https://tailwindcss.com/docs

[5] Node.js Documentation. https://nodejs.org/en/docs

[6] Express.js Documentation. https://expressjs.com/

[7] PostgreSQL Documentation. https://www.postgresql.org/docs/

[10] JSON Web Token (JWT) – RFC 7519. https://www.rfc-editor.org/rfc/rfc7519

[11] The Transport Layer Security (TLS) Protocol Version 1.3 – RFC 8446. https://www.rfc-editor.org/rfc/rfc8446

[12] HTTP Strict Transport Security (HSTS) – RFC 6797. https://www.rfc-editor.org/rfc/rfc6797

[13] OWASP Top 10. https://owasp.org/www-project-top-ten/

[14] OWASP Application Security Verification Standard (ASVS). https://owasp.org/www-project-application-security-verification-standard/

[15] Provos, N., & Mazières, D. (1999). A Future-Adaptable Password Scheme. USENIX Annual Technical Conference..

[16] Splitwise. https://www.splitwise.com/

[17] Tricount. https://www.tricount.com/

[18] Splitable GitHub Deposu. <https://github.com/keremy321/Splitable>