WeTrackX – İsterler (Next.js + Nest.js + PostgreSQL)

Case Çalışması İçin Yazılım İsterleri Dokümanı

# 0) Amaç (Yalın)

Bu dokümanın amacı WCANX case çalışması kapsamında geliştirilecek kişisel finans uygulamasının gereksinimlerini ortaya koymaktır.

• Kullanıcıların gelir-gider işlemlerini kaydedebilmesi

• İşlemleri kategorize edebilmesi

• Basit raporlar ve özetler alabilmesi

Bu çalışma, işlevsellik ve kod kalitesi odaklıdır. Deploy süreci kapsam dışıdır.

# 1) Minimum Teknik Zorunlular

• Backend: Nest.js + Prisma

• Veritabanı: PostgreSQL (NUMERIC(14,2) alanlarla)

• Frontend: Next.js (App Router)

• Authentication: JWT tabanlı

• Grafikler: Chart.js veya Recharts

Not: Çalışma yalnızca lokal ortamda koşacak; README’de kurulum ve çalıştırma adımları yer almalıdır.

# 2) Veri Modeli

**Varlıklar:**

• User: id, email(unique), passwordHash, name, createdAt

• Category: id, userId, name, type(INCOME/EXPENSE), color?, priority(default 0), createdAt

• Transaction: id, userId, categoryId, title, note?, amount(Decimal), date, createdAt, updatedAt

**Kurallar:**

• INCOME işlemleri amount > 0

• EXPENSE işlemleri amount < 0

• Kullanıcı bazında aynı name+type kombinasyonu tekil olmalı (liste/filtre performansı)

• Transaction.date üzerinde index

Yer: backend/prisma/schema.prisma

# 3) API Yüzeyi

**3.1 Auth**

* **POST /auth/register** → { user, token }
  + Body: { name, email, password }
* **POST /auth/login** → { user, token }
  + Body: { email, password }
* **GET /auth/me** (JWT) → { user }

**3.2 Categories (JWT)**

* **GET /categories?type=INCOME|EXPENSE** → [...categories]
* **POST /categories** → 201
  + Body: { name, type, color?, priority? }
  + Çakışma: aynı name+type için 409
* **PATCH /categories/:id** → 200
* **DELETE /categories/:id** → 204
  + İlişkili işlem varsa 409 (silme yok)

**3.3 Transactions (JWT)**

* **GET /transactions** → { items, total, page, pageSize }
  + Query: from, to, categoryId?, q?, page?, pageSize?, sort?=date|amount, order?=asc|desc
* **POST /transactions** → 201
  + Body: { title, note?, amount, categoryId, date }
  + **İşaret kontrolü:** kategori tipine aykırı ise 400
* **PATCH /transactions/:id** → 200 (aynı doğrulamalar)
* **DELETE /transactions/:id** → 204

**3.4 Reports (JWT)**

* **GET /reports/summary?from=YYYY-MM-DD&to=YYYY-MM-DD**  
  → { income, expense, balance }
* **GET /reports/by-category?period=month&date=YYYY-MM**  
  → [ { categoryId, name, type, total } ]
* **Not:** Timeseries opsiyonel. Case için summary + by-category yeterli.

# 4) Frontend

**4.1 Sayfalar**

* **/register**: Form (name/email/password) → success → /login
* **/login**: Form (email/password) → success → /dashboard
* **/dashboard**:
  + Üstte **Aylık özet kartları**: income / expense / balance ( /reports/summary )
  + **Son 5 işlem** listesi
* **/transactions**:
  + Liste: tarih sıralı, **filtreler** (from–to, kategori, arama), **sayfalama**
  + Ekle/Düzenle/Sil dialogları
* **/categories**:
  + Liste + Ekle/Düzenle/Sil
  + Renk seçimi, öncelik alanı
* **/reports**:
  + **Kategori bazlı dağılım**: pie/bar ( /reports/by-category )

**4.2 UX**

* **Toast**: başarılı/hatada bilgilendirme
* **Form doğrulama**: zorunlu alanlar, amount 2 ondalık, date picker
* **Responsive**: mobil/masaüstü

# 5) Doğrulama & Hata Mesajları

• Email formatı, şifre min. 8 karakter

• Kategori duplicate durumda 409

• Transaction amount işaret kontrolü

• Hata formatı: { "statusCode": 400, "message": "..." }

# 6) Seed Verisi

• Demo kullanıcı: demo@demo.com / Demo123!

• Öntanımlı kategoriler: Maaş (INCOME), Yemek, Ulaşım, Eğlence (EXPENSE)

• Son 30 gün için örnek işlemler

Yer: backend/prisma/seed.ts

# 7) README – Kurulum Adımları

**Backend:**

cd backend

cp .env.example .env

npm install

npx prisma migrate dev --name init

npx prisma db seed

npm run start:dev

**Frontend:**

cd frontend

cp .env.local.example .env.local

npm install

npm run dev

# 8) Kabul Kriterleri

• Auth uçları (register/login/me) çalışır

• Category CRUD sorunsuz, duplicate engellenir

• Transaction CRUD, filtreleme ve işaret kontrolü yapılır

• Report uçları doğru sonuç döner

• UI: Dashboard, Transactions, Categories, Reports işler

• README’den adımları izleyen kurulum yapılabilir

# 9) Yapılacaklar Listesi

1. **Backend iskelet** + Prisma + PostgreSQL bağlantısı
2. **Auth** (register/login/me) + şifre hash + JWT guard
3. **Category** CRUD + duplicate kontrolü + seed
4. **Transaction** CRUD + filtre/sayfalama + **amount işaret kuralı**
5. **Reports** (summary, by-category)
6. **Frontend** sayfaları (login/register → dashboard → transactions → categories → reports)
7. **Grafikler** (reports ekranı)
8. **README son kontrol** + kısa ekran görüntüsü (opsiyon)