

Codenight Case (Orta Seviye): TV+ Sosyal İzleme – Ortak İzleme Odası, Senkron, Oylama ve Split

Amaç

Ekipler **6–8 saat** içinde TV+ içerikleri için “arkadaşlarla aynı anda izleme” deneyimini simüle eden bir web/miniap geliştirir.

Özellik seti: **oda oluşturma, senkron oynatma (host kontrolü), oylama ile içerik seçimi, sohbet/emoji tepkileri, masraf paylaşımı (snack/pizza) ve hatırlatma.**

Senaryo (Kısa)

Öğrenciler TV+’ta film/dizi/maç izlerken birlikte takılmak istiyor. **Sosyal İzleme**, bir “oda” kurup arkadaşları davet eder, aday içerikleri oylatır, host play/pause yapınca herkeste senkronlar, sohbet/emoji gönderir ve gecenin giderlerini **split** eder. Etkinlikten 1 saat önce gruba **hatırlatma** (mock push) yollar.

MVP (Olmazsa Olmaz)

1. Oda & Davet

- Oda oluştur (başlık, tarih/saat).
- Davet linki/QR (mock).
- Roller: **host** (tek kişi), **member**.

2. İçerik Oylaması

- TV+ kataloğundan (mock) aday listesi → kullanıcılar oy verir.
- En çok oy alan içerik “seçili içerik”.

3. Senkron Oynatma (Mock)

- Host play/pause/seek → odadaki herkese aynı “zaman damgası” yayılır.
- Basit **drift düzeltme**: client farkı > 2 sn ise seek.

4. Sohbet & Emoji Tepkileri

- Metin mesajları ve hızlı emoji (👏 😂 😮 ❤️).
- Basit spam koruması: kullanıcı başına **2 sn** rate-limit.

5. Masraf Paylaşımı (Split)

- Gider ekle (tutar, açıklama).
- **Eşit** ya da **ağırlıklı** (weight) bölüşüm.
- Kişi bazlı **net bakiye** hesapla.

6. Hatırlatma

- Etkinlikten **1 saat** önce “oda özetini” mock push et: içerik, saat, katılımcı sayısı.
-

Artı Puan (Seçmeli)

- **Watch party sayacı** (başlayana kalan süre).
 - **Mini anket** (ör. “altyazı mı dublaj mı?”).
 - **Bağlantı kalitesi göstergesi** (mock ping).
 - **Alt yazı dili senkronu**.
-

Basit Mimari

- **HTTP API + WebSocket (mock)**
 - WS kanalı: /ws/rooms/{room_id} → event tipi: play, pause, seek, chat, emoji, sync_ping.
 - **Senkron kuralı:** host’tan gelen seek_ts tüm client’larda uygulanır; client her 10 sn sync_ping atar.
-

API (Öneri)

POST /rooms

BODY: { title, start_at, host_id }

-> 200 { room_id, invite_url }

POST /rooms/{id}/candidates

BODY: { items:[{content_id}...] }

POST /rooms/{id}/vote

BODY: { user_id, content_id }

GET /rooms/{id}/summary

-> { selected_content, votes:{content_id:count}, members, start_at }

POST /rooms/{id}/expenses

BODY: { user_id, amount, note, weight? }

GET /rooms/{id}/balances

-> { totals, per_user:[{user_id,balance}] }

POST /rooms/{id}/remind (mock)

-> { status:"ok" }

Örnek Veri (CSV Şemaları)

users.csv

user_id,name,avatar

catalog.csv (TV+ mock içerik)

content_id,title,type,duration_min,tags

type: movie|series|sports

rooms.csv

room_id,title,start_at,host_id

candidates.csv

room_id,content_id

votes.csv

room_id,content_id,user_id

sync_events.csv (isteğe bağlı log)

room_id,user_id,action,ts,position_sec (*action: play/pause/seek/join/leave*)

chat.csv

room_id,user_id,message,created_at

emojis.csv

room_id,user_id,emoji,created_at

expenses.csv

expense_id,room_id,user_id,amount,note,weight

Kurallar (Özet)

- **Kazanan içerik:** en çok oy; eşitlikte host seçer.
 - **Senkron:** host referans zaman → tüm client'lar position_sec'e seek.
 - **Split:** $pay_i = total * (w_i / \sum w)$; weight boşsa 1.
 - **Rate-limit:** aynı user'dan ardışık chat/emoji ≤ 2 sn.
-

UI/UX Beklentisi

- **Tek sayfa** layout: sol "video player" (mock süre göstergesi), sağ sekmeler: **Oylama / Sohbet / Split**.
 - "Senkronda değil misin?" butonu → anında seek.
 - Oda üst bar: seçili içerik adı + kalan süre sayacı.
-

Puanlama (100)

- **Çalışabilirlik (30):** oda → oy → senkron → split akışı
- **Gerçek zamanlılık (20):** WS eventleri düzgün
- **Veri modeli (20):** oy & split doğruluğu
- **UI/UX (20):** net, mobil uyumlu

- **Bonus (10):** mini anket, ping göstergesi, altyazı senkronu