

# FootballAI - Implementation Guide

Bu döküman, FootballAI projesine eklenen yeni özelliklerin kullanım kılavuzudur.

## Yapılan Değişiklikler

### Backend (services/match-service)

#### 1. Auth Service

- JWT tabanlı authentication sistemi
- Kullanıcı kaydı ve girişi
- Token doğrulama middleware
- **Dosyalar:**
  - `src/services/auth.service.ts`
  - `src/controllers/auth.controller.ts`
  - `src/routes/auth.routes.ts`
  - `src/middleware/auth.middleware.ts`
  - `src/types/auth.types.ts`

#### 2. Stats Service

- Lig puan durumu (Redis cache: 1 saat)
- Takım istatistikleri ve form hesaplama
- Kafa kafaya (H2H) verileri
- **Dosyalar:**
  - `src/services/stats.service.ts`
  - `src/controllers/stats.controller.ts`
  - `src/routes/stats.routes.ts`

#### 3. AI Prediction Service

- Gemini 2.0 Flash entegrasyonu
- Detaylı prompt engineering
- Zod ile JSON output validation
- **Dosyalar:**
  - `src/services/ai-prediction.service.ts`
  - `src/controllers/prediction.controller.ts`
  - `src/routes/prediction.routes.ts`
  - `src/types/prediction.types.ts`

#### 4. Config Güncellemeleri

- JWT ve Gemini AI konfigürasyonları eklendi
- Environment variable validation güncellendi
- **Dosya:** `src/config/index.ts`

#### 5. Package.json Güncellemeleri

Yeni bağımlılıklar eklendi:

- `jsonwebtoken` - JWT token üretimi

- `bryptjs` - Şifre hashleme
- `@google/generative-ai` - Gemini AI entegrasyonu
- `@types/jsonwebtoken` - TypeScript tipleri
- `@types/bcryptjs` - TypeScript tipleri

## Frontend (apps/web)

### 1. API Client

- Axios tabanlı HTTP client
- JWT interceptor ile otomatik token yönetimi
- **Dosya:** `src/lib/api-client.ts`

### 2. Auth Hook

- Register, login, logout fonksiyonları
- Zustand store entegrasyonu
- **Dosya:** `src/hooks/useAuth.ts`

### 3. Zustand Stores

- Auth state yönetimi (localStorage persistence)
- Prediction state yönetimi
- **Dosyalar:**
  - `src/stores/auth.store.ts`
  - `src/stores/prediction.store.ts`

### 4. UI Components

- Match Prediction component (AI tahmin görüntüleme)
- AI Highlights component (Dashboard için)
- Prediction hook (API çağrıları için)
- **Dosyalar:**
  - `src/components/prediction/MatchPrediction.tsx`
  - `src/components/prediction/AIHighlights.tsx`
  - `src/hooks/usePrediction.ts`

## Environment Variables

Backend `.env` dosyasına eklenecek değişkenler:

```
# JWT Authentication
JWT_SECRET="your-super-secret-jwt-key-min-32-characters-long"
JWT_EXPIRES_IN="7d"

# Gemini AI
GEMINI_API_KEY="AIzaSy..."
GEMINI_MODEL="gemini-2.0-flash-exp"
```

Frontend `.env.local` dosyasına eklenecek değişkenler:

```
NEXT_PUBLIC_API_URL="http://localhost:3001"
```

# Kullanım Örnekleri

## Backend API Endpoints

### 1. Kullanıcı Kaydı

```
POST /api/auth/register
Content-Type: application/json

{
  "email": "user@example.com",
  "password": "12345678",
  "name": "Test User"
}
```

### 2. Kullanıcı Girişи

```
POST /api/auth/login
Content-Type: application/json

{
  "email": "user@example.com",
  "password": "12345678"
}
```

### 3. Mevcut Kullanıcı Bilgisi

```
GET /api/auth/me
Authorization: Bearer <token>
```

### 4. Lig Puan Durumu

```
GET /api/stats/standings/:leagueId/:season
# Örnek: /api/stats/standings/203/2024
```

### 5. Takım İstatistikleri

```
GET /api/stats/team/:teamId/:leagueId/:season
# Örnek: /api/stats/team/541/203/2024
```

### 6. Kafa Kafaya İstatistikleri

```
GET /api/stats/h2h?t1=541&t2=542
```

### 7. AI Maç Tahmini (JWT Gerekli)

```
GET /api/predictions/:fixtureId
Authorization: Bearer <token>
# Örnek: /api/predictions/12345
```

## Frontend Kullanımı

### 1. Auth Hook Kullanımı

```
import { useAuth } from '@/hooks/useAuth';

function LoginPage() {
  const { login, isLoading, error } = useAuth();

  const handleLogin = async (email: string, password: string) => {
    const user = await login({ email, password });
    if (user) {
      // Giriş başarılı
      router.push('/dashboard');
    }
  };

  return (
    // Login form...
  );
}
```

### 2. Match Prediction Component

```
import { MatchPrediction } from '@/components/prediction/MatchPrediction';

function MatchDetailPage({ fixtureId }: { fixtureId: number }) {
  return (
    <div>
      <MatchPrediction
        fixtureId={fixtureId}
        homeTeamName="Galatasaray"
        awayTeamName="Fenerbahçe"
      />
    </div>
  );
}
```

### 3. AI Highlights Component

```
import { AIHighlights } from '@/components/prediction/AIHighlights';

function Dashboard() {
  return (
    <div>
      <AIHighlights />
      {/* Diğer dashboard içerikleri */}
    </div>
  );
}
```

## 4. Zustand Store Kullanımı

```
import { useAuthStore } from '@/stores/auth.store';
import { usePredictionStore } from '@/stores/prediction.store';

function MyComponent() {
  const { user, isAuthenticated } = useAuthStore();
  const { predictions, getPrediction } = usePredictionStore();

  return (
    // Component content...
  );
}
```

## Package Kurulumu

### Backend

```
cd services/match-service
pnpm install
```

### Frontend

```
cd apps/web
pnpm install
```

### Root (Tüm workspace)

```
pnpm install
```

## Test Etme

### Backend Servisi Başlatma

```
cd services/match-service
pnpm dev
```

### Frontend Başlatma

```
cd apps/web
pnpm dev
```

## API Testi (cURL)

```
# Register
curl -X POST http://localhost:3001/api/auth/register \
-H "Content-Type: application/json" \
-d '{"email":"test@test.com", "password":"12345678", "name":"Test User"}'

# Login
curl -X POST http://localhost:3001/api/auth/login \
-H "Content-Type: application/json" \
-d '{"email":"test@test.com", "password":"12345678"}'

# Get Standings
curl http://localhost:3001/api/stats/standings/203/2024

# Get Prediction (with token)
curl http://localhost:3001/api/predictions/12345 \
-H "Authorization: Bearer YOUR_JWT_TOKEN"
```



## Güvenlik Notları

1. **JWT\_SECRET:** Production'da mutlaka güçlü bir secret key kullanın (min 32 karakter)
2. **GEMINI\_API\_KEY:** API anahtarınızı public repositories'de paylaşmayın
3. **Rate Limiting:** Production'da API rate limit ayarlarını optimize edin
4. **CORS:** Frontend domain'inizle sınırlayın



## Sonraki Adımlar

1. Backend servisleri tamamlandı
2. Frontend components tamamlandı
3. Database migration (Prisma schema güncellemesi gerekebilir)
4. Environment variables ayarları
5. Production deployment
6. Testing ve debugging



## Troubleshooting

### “JWT\_SECRET is required” Hatası

```
# .env dosyasında JWT_SECRET ekleyin
JWT_SECRET="your-secret-key-min-32-characters"
```

### “GEMINI\_API\_KEY not set” Uyarısı

```
# .env dosyasında Gemini API key ekleyin
GEMINI_API_KEY="AIzaSy..."
```

## “Module not found” Hatası

```
# Dependencies'leri yeniden yükleyin  
pnpm install
```

## TypeScript Type Hatası

```
# Prisma client'i yeniden generate edin  
cd packages/database  
pnpm db:generate
```



## Referanslar

- [Prisma Docs](https://www.prisma.io/docs) (<https://www.prisma.io/docs>)
- [Google Gemini AI](https://ai.google.dev/docs) (<https://ai.google.dev/docs>)
- [Zustand Docs](https://docs.pmnd.rs/zustand) (<https://docs.pmnd.rs/zustand>)
- [JWT.io](https://jwt.io) (<https://jwt.io>)
- [Zod Validation](https://zod.dev) (<https://zod.dev>)

**Tarih:** 5 Şubat 2026

**Implementasyon:** DeepAgent AI

**Repository:** <https://github.com/yuksel-arslan/footballai>