

Esnek AI Raporlama Projesi - Adım Adım Task Listesi

PROJE GENEL BAKIŞ

İş Akışı Özeti

1. Kullanıcı Kaydı/Girişi → Default 100 kredi
 2. Dosya Yükleme → Excel/PDF yükleme
 3. Kolon Eşleme → "A kolonu maaş, B kolonu departman" gibi
 4. AI Talep → "Departmanlara göre ortalama maaş grafiği çıkar"
 5. Sonuç → Grafik/Rapor/PDF çıktısı
-

ADIM 1 - TEMEL SİSTEM (4 Hafta)

KULLANICI YÖNETİMİ SAYFASI

Backend Tasks

- ☐ User Entity Oluşturma
- ☐ User.java model (id, email, password, firstName, lastName, role, isActive, createdAt)
- ☐ UserRepository interface
- ☐ User validation annotations ekle
- ☐ Authentication Controller
- ☐ POST /api/auth/register endpoint
- ☐ POST /api/auth/login endpoint
- ☐ POST /api/auth/refresh-token endpoint
- ☐ Password hashing (BCrypt)
- ☐ JWT Implementation
- ☐ JWT utility class
- ☐ JWT filter for authentication
- ☐ Token expiry handling (15 min access, 7 day refresh)
- ☐ User Service
- ☐ User registration logic
- ☐ Email uniqueness check
- ☐ Login validation
- ☐ Profile update methods

Frontend Tasks

- ☐ Login Sayfası

- ☐ Login form UI (email, password)
- ☐ Form validation (frontend)
- ☐ Login API call
- ☐ Error handling ve mesajlar
- ☐ Remember me checkbox
- ☐ **Register Sayfası**
- ☐ Register form UI
- ☐ Password confirmation
- ☐ Terms & conditions checkbox
- ☐ API integration
- ☐ Success redirect to login
- ☐ **Profil Sayfası**
- ☐ Profile display UI
- ☐ Edit profile form
- ☐ Password change form
- ☐ Profile update API call

Database Tasks

- ☐ **Users Tablosu**

sql

```
CREATE TABLE users (  
  id BIGSERIAL PRIMARY KEY,  
  email VARCHAR(255) UNIQUE NOT NULL,  
  password_hash VARCHAR(255) NOT NULL,  
  first_name VARCHAR(100),  
  last_name VARCHAR(100),  
  role VARCHAR(50) DEFAULT 'USER',  
  is_active BOOLEAN DEFAULT TRUE,  
  created_at TIMESTAMP DEFAULT NOW()  
);
```

Test Tasks

- ☐ User registration unit test
 - ☐ Login functionality test
 - ☐ JWT token validation test
 - ☐ Profile update test
-

KREDİ SİSTEMİ SAYFASI

Backend Tasks

- ☐ **Credit Entity**
- ☐ UserCredit.java model
- ☐ CreditTransaction.java model
- ☐ Repository interfaces
- ☐ **Credit Service**
- ☐ Default credit assignment (100)
- ☐ Credit usage tracking
- ☐ Credit history logging
- ☐ Credit validation before operations
- ☐ **Credit Controller**
- ☐ GET /api/users/credits - kredi durumu
- ☐ GET /api/users/credit-history - geçmiş
- ☐ POST /api/credits/use - kredi kullanımı

Frontend Tasks

- ☐ **Kredi Dashboard**
- ☐ Kalan kredi gösterimi
- ☐ Kredi progress bar
- ☐ Geçmiş işlemler tablosu
- ☐ "Kredi Satın Al" button
- ☐ **Kredi Geçmişi**
- ☐ İşlem listesi (tarih, tip, miktar)
- ☐ Filtreleme (tarih aralığı)
- ☐ Pagination

Database Tasks

- ☐ **Credit Tables**

sql

```
CREATE TABLE user_credits (  
  id BIGSERIAL PRIMARY KEY,  
  user_id BIGINT REFERENCES users(id),  
  total_credits INT DEFAULT 100,  
  used_credits INT DEFAULT 0,  
  remaining_credits INT GENERATED ALWAYS AS (total_credits - used_credits) STORED  
);  
  
CREATE TABLE credit_transactions (  
  id BIGSERIAL PRIMARY KEY,  
  user_id BIGINT REFERENCES users(id),  
  transaction_type VARCHAR(50),  
  amount INT,  
  description TEXT,  
  created_at TIMESTAMP DEFAULT NOW()  
);
```

Test Tasks

- ☐ Default credit assignment test
- ☐ Credit usage deduction test
- ☐ Credit insufficient validation test

DOSYA YÖNETİMİ SAYFASI

Backend Tasks

- ☐ **File Entity**
- ☐ UploadedFile.java model
- ☐ File validation service
- ☐ S3/MinIO integration
- ☐ **File Upload Service**
- ☐ Multipart file handling
- ☐ File type validation (xlsx, pdf, csv)
- ☐ File size limit (10MB)
- ☐ Unique filename generation
- ☐ **File Controller**
- ☐ POST /api/files/upload
- ☐ GET /api/files (user files)
- ☐ DELETE /api/files/{id}
- ☐ GET /api/files/{id}/download

Frontend Tasks

- ☐ Dosya Yükleme Sayfası
- ☐ Drag & drop zone (react-dropzone)
- ☐ File preview list
- ☐ Upload progress bar
- ☐ File type/size validation
- ☐ Success/error messages
- ☐ Dosya Listesi
- ☐ Uploaded files table
- ☐ File details (name, size, date)
- ☐ Delete confirmation modal
- ☐ File status indicators

Database Tasks

- ☐ Files Table

sql

```
CREATE TABLE uploaded_files (  
  id BIGSERIAL PRIMARY KEY,  
  user_id BIGINT REFERENCES users(id),  
  original_filename VARCHAR(255),  
  stored_filename VARCHAR(255),  
  file_path VARCHAR(500),  
  file_size BIGINT,  
  file_type VARCHAR(100),  
  upload_status VARCHAR(50) DEFAULT 'UPLOADED',  
  created_at TIMESTAMP DEFAULT NOW()  
);
```

Test Tasks

- ☐ File upload success test
- ☐ File type validation test
- ☐ File size limit test
- ☐ File deletion test



DASHBOARD SAYFASI

Frontend Tasks

- ☐ Ana Dashboard
- ☐ Kullanıcı hoş geldin mesajı
- ☐ Kredi durumu widget

- ☐ Son yüklenen dosyalar (5 adet)
- ☐ Hızlı aksiyon butonları
- ☐ Activity feed (son işlemler)
- ☐ **Navigation Menu**
- ☐ Header navigation
- ☐ Sidebar menu (responsive)
- ☐ User dropdown menu
- ☐ Logout functionality

Test Tasks

- ☐ Dashboard loading test
 - ☐ Navigation functionality test
 - ☐ Responsive design test
-



ADIM 2 - DOSYA İŞLEME (6 Hafta)



DOSYA PARSING SAYFASI

Backend Tasks

- ☐ **Excel Parser Service**
- ☐ Apache POI dependency
- ☐ Excel file reading (.xlsx, .xls)
- ☐ Sheet detection
- ☐ Column/row counting
- ☐ Sample data extraction (ilk 5 satır)
- ☐ **PDF Parser Service**
- ☐ PDFBox dependency
- ☐ Table extraction from PDF
- ☐ Text to structured data conversion
- ☐ **CSV Parser Service**
- ☐ OpenCSV library
- ☐ Delimiter detection
- ☐ Header row handling
- ☐ Data type auto-detection
- ☐ **Parse Controller**
- ☐ POST /api/files/{id}/parse
- ☐ GET /api/files/{id}/preview
- ☐ GET /api/files/{id}/columns

Frontend Tasks

- ☐ Dosya Önizleme
- ☐ Parse başlatma butonu
- ☐ Loading spinner during parse
- ☐ Tablo preview (ilk 10 satır)
- ☐ Kolon başlıkları gösterimi
- ☐ Parse error handling

Database Tasks

- ☐ Parsed Data Tables

sql

```
CREATE TABLE file_metadata (  
  id BIGSERIAL PRIMARY KEY,  
  file_id BIGINT REFERENCES uploaded_files(id),  
  sheet_count INT,  
  row_count INT,  
  column_count INT,  
  has_headers BOOLEAN,  
  parse_status VARCHAR(50),  
  error_message TEXT  
);
```

Test Tasks

- ☐ Excel parse accuracy test
- ☐ PDF table extraction test
- ☐ CSV delimiter detection test
- ☐ Large file parsing test

KOLON EŞLEME SAYFASI

Backend Tasks

- ☐ Column Mapping Service
- ☐ Auto data type detection
- ☐ Sample data analysis
- ☐ Mapping save/update operations
- ☐ Mapping Controller
- ☐ POST /api/files/{id}/columns/mapping
- ☐ PUT /api/files/{id}/columns/{columnId}
- ☐ GET /api/files/{id}/mapping

Frontend Tasks

- ☐ Kolon Eşleme UI
- ☐ Drag & drop column interface
- ☐ Column description input fields
- ☐ Data type selection dropdown
- ☐ Sample data preview
- ☐ Save mapping button
- ☐ "Sonraki Adım" navigation
- ☐ Auto-Suggestion
- ☐ Smart column naming
- ☐ Common patterns recognition
- ☐ Data type suggestions

Database Tasks

- ☐ Column Mappings

```
sql

CREATE TABLE column_mappings (
  id BIGSERIAL PRIMARY KEY,
  file_id BIGINT REFERENCES uploaded_files(id),
  column_name VARCHAR(255),
  column_index INT,
  column_description TEXT,
  data_type VARCHAR(100),
  sample_data TEXT[],
  is_active BOOLEAN DEFAULT TRUE
);
```

Test Tasks

- ☐ Column mapping save test
- ☐ Data type detection test
- ☐ Mapping update test

AI ÖZETLEME SAYFASI

Backend Tasks

- ☐ OpenAI Integration
- ☐ OpenAI Java client
- ☐ API key configuration

- ☐ Rate limiting implementation
- ☐ Error handling & retries
- ☐ **AI Service**
- ☐ Data summarization
- ☐ Basic analysis generation
- ☐ JSON response parsing
- ☐ Credit deduction logic
- ☐ **AI Controller**
- ☐ POST /api/ai/summarize
- ☐ GET /api/ai/status/{requestId}

Frontend Tasks

- ☐ **AI Talep Sayfası**
- ☐ Natural language input box
- ☐ Example prompts
- ☐ "Analiz Et" button
- ☐ Processing status indicator
- ☐ Result display area

Database Tasks

- ☐ **AI Requests**

sql

```
CREATE TABLE ai_requests (  
  id BIGSERIAL PRIMARY KEY,  
  user_id BIGINT REFERENCES users(id),  
  file_id BIGINT REFERENCES uploaded_files(id),  
  request_text TEXT,  
  status VARCHAR(50) DEFAULT 'PENDING',  
  credits_used INT DEFAULT 1,  
  created_at TIMESTAMP DEFAULT NOW()  
);
```

Test Tasks

- ☐ OpenAI API integration test
 - ☐ AI response validation test
 - ☐ Credit deduction test
-



ADIM 3 - GELİŞMİŞ RAPORLAMA (6 Hafta)



KULLANICI TALEBİ VE AI İŞLEME

Backend Tasks

- ☐ **Advanced AI Service**
- ☐ Complex query processing
- ☐ Chart configuration generation
- ☐ Multi-step analysis
- ☐ Job queue implementation (Redis/RabbitMQ)
- ☐ **Async Processing**
- ☐ Background job handling
- ☐ Progress tracking
- ☐ Job status updates
- ☐ Email notifications

Frontend Tasks

- ☐ **Gelişmiş Talep UI**
- ☐ Rich text input
- ☐ Analysis type selection (Chart, Table, Report)
- ☐ Template suggestions
- ☐ Real-time preview
- ☐ Multi-step wizard

Test Tasks

- ☐ Complex query processing test
- ☐ Async job execution test
- ☐ Progress tracking test



RAPORLAMA VE GRAFİKLER SAYFASI

Backend Tasks

- ☐ **Chart Generation Service**
- ☐ Chart.js configuration builder
- ☐ Data transformation for charts
- ☐ PDF report generation
- ☐ Excel export functionality
- ☐ **Report Controller**
- ☐ GET /api/reports/{id}

- ☐ POST /api/reports/generate
- ☐ GET /api/reports/{id}/download

Frontend Tasks

- ☐ **Grafik Gösterimi**
- ☐ Chart.js integration
- ☐ Interactive charts
- ☐ Chart type switching
- ☐ Data filtering controls
- ☐ Export options (PNG, PDF)
- ☐ **Tablo Gösterimi**
- ☐ Sortable data tables
- ☐ Pagination
- ☐ Search functionality
- ☐ Column visibility toggle
- ☐ **Rapor İndirme**
- ☐ PDF download
- ☐ Excel export
- ☐ CSV export
- ☐ Share functionality

Database Tasks

- ☐ **Reports Tables**

sql

```
CREATE TABLE ai_results (  
  id BIGSERIAL PRIMARY KEY,  
  request_id BIGINT REFERENCES ai_requests(id),  
  result_type VARCHAR(100),  
  result_data JSONB,  
  file_path VARCHAR(500),  
  created_at TIMESTAMP DEFAULT NOW()  
);
```

Test Tasks

- ☐ Chart generation test
 - ☐ PDF export test
 - ☐ Data filtering test
-



ADIM 4 - KURUMSAL & ADMIN (4 Hafta)



ADMIN PANELİ SİSTEMİ

Backend Tasks

- ☐ **Admin User Management**
- ☐ Admin role implementation
- ☐ User CRUD operations
- ☐ Bulk user operations
- ☐ User activity tracking
- ☐ **Admin Controllers**
- ☐ GET /api/admin/users (paginated)
- ☐ PUT /api/admin/users/{id}/status
- ☐ GET /api/admin/analytics
- ☐ GET /api/admin/system-stats

Frontend Tasks

- ☐ **Admin Dashboard**
- ☐ System overview widgets
- ☐ User statistics charts
- ☐ Revenue analytics
- ☐ Recent activities feed
- ☐ **Kullanıcı Yönetimi**
- ☐ User list with search/filter
- ☐ User detail modal
- ☐ Account activation/deactivation
- ☐ Credit adjustment interface
- ☐ Bulk actions
- ☐ **Kredi Yönetimi**
- ☐ Credit distribution overview
- ☐ Manual credit adjustment
- ☐ Credit usage analytics
- ☐ Refund operations
- ☐ **Sistem İzleme**
- ☐ Server health indicators
- ☐ API usage statistics
- ☐ Error rate monitoring
- ☐ Performance metrics

Database Tasks

☐ Admin Tables

```
sql

CREATE TABLE admin_activities (
  id BIGSERIAL PRIMARY KEY,
  admin_id BIGINT REFERENCES users(id),
  activity_type VARCHAR(100),
  target_user_id BIGINT,
  description TEXT,
  created_at TIMESTAMP DEFAULT NOW()
);

CREATE TABLE system_stats (
  id BIGSERIAL PRIMARY KEY,
  metric_name VARCHAR(100),
  metric_value DECIMAL,
  recorded_at TIMESTAMP DEFAULT NOW()
);
```

Test Tasks

- ☐ Admin authentication test
- ☐ User management operations test
- ☐ Credit adjustment test

ROL BAZLI YÖNETİM

Backend Tasks

- ☐ Role Management
 - ☐ Role-based access control (RBAC)
 - ☐ Permission system
 - ☐ Multi-tenant support preparation
 - ☐ Organization management
- ☐ Security Enhancement
 - ☐ API endpoint protection
 - ☐ Method-level security
 - ☐ Audit logging
 - ☐ Session management

Frontend Tasks

- ☐ Rol Yönetimi UI
- ☐ Role assignment interface
- ☐ Permission matrix
- ☐ Organization hierarchy
- ☐ Access level indicators

Database Tasks

- ☐ RBAC Tables

sql

```
CREATE TABLE roles (  
  id BIGSERIAL PRIMARY KEY,  
  name VARCHAR(50) UNIQUE,  
  description TEXT  
);  
  
CREATE TABLE user_roles (  
  user_id BIGINT REFERENCES users(id),  
  role_id BIGINT REFERENCES roles(id),  
  PRIMARY KEY (user_id, role_id)  
);  
  
CREATE TABLE permissions (  
  id BIGSERIAL PRIMARY KEY,  
  name VARCHAR(100) UNIQUE,  
  resource VARCHAR(100),  
  action VARCHAR(100)  
);
```

ÖDEME SİSTEMİ

Backend Tasks

- ☐ Payment Integration
- ☐ Stripe API integration (international)
- ☐ Iyzico integration (Turkey)
- ☐ Webhook handling
- ☐ Payment verification
- ☐ Subscription Service
- ☐ Plan management

- ☐ Recurring billing
- ☐ Pro-ration handling
- ☐ Invoice generation

Frontend Tasks

- ☐ Ödeme Sayfası
- ☐ Plan selection UI
- ☐ Credit package options
- ☐ Payment form (Stripe Elements)
- ☐ Invoice display
- ☐ Payment history
- ☐ Fatura Yönetimi
- ☐ Invoice list
- ☐ PDF invoice download
- ☐ Payment status tracking

Database Tasks

- ☐ Payment Tables

sql

```
CREATE TABLE payment_plans (  
  id BIGSERIAL PRIMARY KEY,  
  name VARCHAR(100),  
  credits_included INT,  
  price DECIMAL(10,2),  
  currency VARCHAR(10) DEFAULT 'TRY'  
);  
  
CREATE TABLE payments (  
  id BIGSERIAL PRIMARY KEY,  
  user_id BIGINT REFERENCES users(id),  
  plan_id BIGINT REFERENCES payment_plans(id),  
  amount DECIMAL(10,2),  
  currency VARCHAR(10),  
  payment_provider VARCHAR(50),  
  payment_intent_id VARCHAR(255),  
  status VARCHAR(50),  
  created_at TIMESTAMP DEFAULT NOW()  
);
```

Test Tasks

- ☐ Payment processing test

- ☐ Webhook handling test
 - ☐ Credit assignment after payment test
-

ANALYTICS VE RAPORLAMA

Backend Tasks

- ☐ Analytics Service
- ☐ User behavior tracking
- ☐ Business metrics calculation
- ☐ Performance monitoring
- ☐ Report generation

Frontend Tasks

- ☐ Analytics Dashboard
 - ☐ KPI widgets
 - ☐ Usage trends
 - ☐ Revenue reports
 - ☐ User engagement metrics
-

DEPLOYMENT VE TEST

DEPLOYMENT HAZIRLIK

DevOps Tasks

- ☐ Docker Configuration
- ☐ Frontend Dockerfile
- ☐ Backend Dockerfile
- ☐ docker-compose.yml
- ☐ Environment variables setup
- ☐ CI/CD Pipeline
- ☐ GitHub Actions workflow
- ☐ Automated testing
- ☐ Build & deployment process
- ☐ Database migration scripts

Production Tasks

- ☐ Server Setup
- ☐ AWS/Digital Ocean setup

- ☐ SSL certificate configuration
- ☐ Domain configuration
- ☐ Backup strategy

Test Tasks

- ☐ Comprehensive Testing
- ☐ Unit test coverage >80%
- ☐ Integration tests
- ☐ E2E testing (Cypress)
- ☐ Performance testing
- ☐ Security testing



DOKÜMANTASYON



TEKNİK DOKÜMANTASYON

- ☐ API Documentation
- ☐ Swagger/OpenAPI setup
- ☐ Endpoint documentation
- ☐ Authentication guide
- ☐ Error handling guide
- ☐ User Documentation
- ☐ User manual (Türkçe)
- ☐ Video tutorials
- ☐ FAQ section
- ☐ Troubleshooting guide



HIZLI BAŞLANGIÇ PLANI

İlk Gün Yapılacaklar:

- ☐ GitHub repository oluştur
- ☐ Spring Boot backend başlat
- ☐ React frontend başlat
- ☐ PostgreSQL database setup
- ☐ Temel project structure


İlk Hafta Hedefi:

- ☐ User registration/login çalışır durumda

- ☐ Database bağlantıları hazır
- ☐ Temel UI layout tamamlanmış

Her Adım Sonrası:

- ☐ Demo hazırlığı
- ☐ Test case'ler çalıştırılmış
- ☐ Code review yapılmış
- ☐ Documentation update edilmiş

Bu task listesi ile her özelliği step-by-step takip edebilir, tamamladığınız her task'ı  ile işaretleyebilirsiniz. Hangi adımdan başlamak istiyorsunuz?