Flare UI Tasarım Dokümanı (Güncellenmiş)

# 📌 Genel Amaç

Flare Administration UI, kullanıcıların Flare platformundaki projeleri yönetebileceği, versiyonlar oluşturabileceği, publish edebileceği, Spark ile entegre çalışabileceği, concurrency kontrolü yapabileceği, eksik bilgi doğrulamaları yapabileceği bir web tabanlı arayüzdür.

# 🏗 Ana Özellikler

- Kullanıcı Giriş (Authentication) → admin login, şifre değiştirme  
- Ortam Konfigürasyonu → work\_mode, spark\_endpoint, generation\_config  
- Proje Yönetimi → yeni proje ekleme, listeleme, düzenleme, silme  
- Intent ve API Yönetimi → intent ekleme/silme, API tanımı ekleme/silme  
- Spark Yönetimi → startup, enable, disable, delete  
- Concurrency Kontrolü → client\_last\_updated timestamp eşleşmesi  
- Eksik Bilgi Kontrolleri → boş geçilemez alanlar backend tarafından doğrulanır  
- Basit HTML + Bootstrap + JavaScript frontend (modüler dosya yapısı)

## 🗂 Frontend Dosya Yapısı

/static/index.html → ana HTML arayüzü  
/static/style.css → görsel stil dosyası  
/static/js/common.js → genel API yardımcıları  
/static/js/auth.js → login, şifre işlemleri  
/static/js/config.js → config işlemleri  
/static/js/project.js → proje, intent, API işlemleri  
/static/js/spark.js → Spark işlemleri  
/static/js/test.js → test runner işlemleri

# 🧪 Flare Admin Test Planı

## 1️⃣ Login Modülü

- Doğru kullanıcı adı ve şifre ile login başarılı  
- Yanlış kullanıcı adı ile login reddedilir  
- Yanlış şifre ile login reddedilir  
- Şifre değiştirme sonrası eski şifre ile login başarısız, yeni şifre ile başarılı

## 2️⃣ Config Modülü

- Mevcut config yüklenir (GET /config/get)  
- Config’te work\_mode veya spark\_endpoint boş girilirse update reddedilir  
- Config doğru güncellendiğinde backend dosyası güncellenir ve tekrar okunabilir

## 3️⃣ Project Modülü

- Yeni proje eklenir (POST /project/add)  
- Aynı isimle tekrar proje eklenemez (hata döner)  
- Proje listesi doğru döner (GET /project/list)  
- Proje update edilirken concurrency (last\_updated) kontrolü yapılır  
- Publish işlemi eksik repo\_id veya fine\_tune\_zip varsa reddedilir  
- Intent eklenir (POST /project/add\_intent)  
- Aynı intent ismi tekrar eklenemez (hata döner)  
- Intent silinir (POST /project/delete\_intent)  
- API tanımı eklenir (POST /project/add\_api)  
- Aynı API tekrar eklenemez (hata döner)  
- API tanımı silinir (POST /project/delete\_api)

## 4️⃣ Spark Modülü

- Startup payload gönderilir ve Spark’tan başarılı dönüş alınır (POST /spark/startup)  
- Project list Spark’tan çekilir (GET /spark/project\_list)  
- Enable, disable ve delete istekleri Spark’a gönderilir ve backend doğru yanıt verir

## 5️⃣ Test Runner

- Run Tests butonu çağrıldığında mock mesaj döner (POST /test/run)

## 6️⃣ Frontend UI

- Tüm butonlar doğru JS fonksiyonlarına bağlı ve çalışır  
- Başarılı çağrılarda yeşil/gri mesaj, hatalarda kırmızı mesaj görsel geri bildirim alınır  
- JSON alanlarında yanlış JSON girilirse hata loglanır (console’da)