

Tayfur Ünal

mtayfurunal@gmail.com



open business software solutions

Summer Internship 2020

Çözüm Mimarisi

Backend:

Staj programımız kapsamında bizden yapmamız istenilen mentorluk uygulamasının tek kısıtı Java tabanlı bir uygulama olmasıydı. Bunu gerçekleştirmek için şu an tüm dünya genelindeki Java geliştiricileri tarafından en çok tercih edilen *Spring Framework* çatısı altında *Spring Boot* kullanmaya karar verdim.

1. Spring Boot:

Spring Boot'u tercih etme sebebim başlıca şu üç ana sebepten kaynaklandı.

1- Arkasındaki büyük topluluk gücünden faydalanabilmek ve bu sayede en güncel çözümler ile sıkıntısız şekilde çalışabilmek. Yine bu topluluğun getirdiği bir fayda olarak çok fazla kaynağa erişebilmem ve karşılaştığım problemlere cevaplar bulabilecek olmam.

2- Uygulamayı başlattığım an bana hazır ve hızlı bir geliştirme ortamı sunması. Kendi içinde bir web server ile gelmesi, ilk etapta hiçbir ek konfigürasyon yapma ihtiyacı duymadan hızlıca işimin geliştirme kısmına odaklanma imkanını bana sunması.

3- Spring Framework çatısı altında bulunan Spring Security, Spring Data, Spring Social, Spring Mail gibi pek çok yardımcı aracı kolayca uygulamama entegre edebilme imkanı sunması ve bu sayede oluşacak muhtemel ihtiyaçlarıma pratik çözümler bulabilme imkanım.

2. PostgreSQL:

Uygulamamın veri mimarisini yönetebilmem için ilişkisel bir veritabanına ihtiyacım vardı. MSSQL, MySQL ve pek çok seçeneğin arasından PostgreSQL kullanma nedenim ise şu sebeplerden kaynaklanıyor:

- Ücretsiz ve açık kaynaklı olması; ek lisans ücretleri ödeme zorunluluğum olmaması.
- Platform bağımsız çalışabilmesi.
- Rakipleriyle performans kıyaslamalarında iyi dereceler elde etmesi.
- Arkasındaki büyük topluluk gücü.

3. Elasticsearch:

Uygulama kapsamında yapmam gereken *Free Text Search* işlemi konusunda daha önce tecrübem yoktu fakat araştırmalarımın ve dokümanın beni yönlendirmesi ile bu işlem için tercihim Elasticsearch'den yana kullandım.

- Uygulama, entegre kolaylığı. Spring Boot uygulaması ile beraber kullanımı için çok fazla ek konfigürasyona ihtiyaç yok.
- Performans, hız ve ölçeklenebilirlik açısından benchmark testlerinde ön sıralarda yer alıyor.

4. Quartz:

Uygulamadaki kullanıcıların dahil oldukları mentorluk süreçlerine ait fazların bitiş sürelerine bir saat kala mail ile bilgilendirmem gerekiyordu. Bunun için yaptığım araştırmalarda *Quartz Planlayıcısına* denk geldim ve tam olarak aradığım şey olduğuna karar verdim. Ayrıca Spring Boot ile beraber kullanabilme kolaylığı da bir diğer tercih nedenim oldu. İşleyiş mantığı kısaca şöyle çalışıyor; planlayıcı fonksiyona gelen istekler zamana bağlı olarak *Job* olarak veritabanına kaydediliyor ve kayıtlı zamanı gelince otomatik olarak tetiklenerek kullancılara mail gönderimi yapıyor.

Frontend:

Spring Boot ile geliřtirdiđim API'ları iřleyip, kullanıcıların kullanımına sunmak için bir Javascript kütüphanesi olan React'ı kullandım.

1. React

2020 yılı itibariyle dünyanın en çok kullanılan frontend kütüphanesi konumunda bulunan React'ın sunduđu temiz ve hızlı geliştirme ortamı benim bu tercihimde en önemli rolü oynadı.

- Karmařık olmayan yapı ve öğrenme kolaylıđı.
- Büyük topluluk gücü.
- Hızlı geliştirme imkanı.

2. Redux

Redux uygulama state'ini global olarak tutma imkanı sađlayan bir kütüphane. Benim kullanma nedenim ise tüm uygulamamın state'ini global olarak tutmaktan ziyade authentication işlemleri için giriş yapan kullanıcı bilgisini redux sayesinde saklayıp daha sonra oturum boyunca kullanmak istememdi.

3. Bootstrap

Popüler bir HTML, CSS, JS kütüphanesi olan Bootstrap'i benim tercih etme nedenim sađladığı hazır component'ler sayesinde frontend dizaynımı hızlıca çıkarabilmek ve uygulamama responsive tasarım çözümü sađlayabilmekti.