Comp2Dust

Project Evaluation Document

Alpha

Başlangıçta bir sürü kaynaktan jpeg algoritmasını araştırdım, kodlar buldum. Fakat bulduğum makaleler konsepti yalın bir şekilde anlatsa da bunu nasıl kodlayacağımı bilmiyordum. Bulduğum kodların ise çoğu anlaşılmaz derecede karmaşık veya kapsüllenmişti. Amaca en çok yaklaştıracak kodları ise Matlab üzerinde buldum fakat bunları web için nasıl gerçekleştireceğimi bilmiyordum. Hala bunun nasıl yapılabileceğini merak ediyorum. Algoritmayı baştan yazamayacağıma karar verince, hazır paketleri kullanmaya karar verdim.

Bu dönem makine mühendisliğinden yatay geçiş yaptım ve birden proje ödevleriyle karşılaşmayı beklemiyordum. Ödevin konusu da hiç fikrimin olmadığı bir şey olunca biraz yetersizlik hissine kapıldım. Daha önceki dönemlerde ne öğrenmiş olabilirler ki diyordum. Fakat öğrenci gruplarında diğerlerinin de aynı durumda olduğunu gördüm.

Alpha sürümü için kullandığım npm paketi compress-js idi. Paket client tarafında çalıştığı için back-end tarafında çok uğraşmam gerekmemişti. Daha önceden sadece bir kurstan izlediğim html, css ve biraz bootstrap bilgim vardı. Bu aşamada jquery, Express ve basit derecede NodeJS, MongoDB öğrendim. Vize için verdiğim alpha sürümü çalışıyordu fakat daha fazla nasıl ilerleteceğimi bilmiyordum, vizelerden sonra jpeg algoritmasını tekrar araştırdım ve yine kendi kendime yazamayacağıma karar verdim.

Final

Vizelerden sonra hemen ReactJs öğrenmeye koyuldum ve front-end i baştan inşa ettim. Server tarafında tahmin ettiğimden çok daha fazla şeyle uğraşmam gerekti. Her sorunu aşama aşama öğrenerek ilerledim.

Karşılaştığım en büyük sorunlardan bazıları

* ReactJs’in ve server’ın farklı portlarda çalıştığını öğrendim ve bunu çözmem bir gün sürdü.
* Javascriptteki arrow ve asenkron fonksiyonları anlamam zaman aldı
* İlk başta yüklenen ve işlenen dosyaları client tarafında tutuyordum. Bu dosyalar her seferinde değiştiği zaman sayfanın tekrar yüklenmesine sebep oluyordu. Dosyaları server’a taşıdığım zaman bu sefer client tarafından erişilemediğini fark ettim. Burada client ve server’ın farkını anlamaya başladım. Bu sorunuda express’le static dosyaları sunarak çözdüm.
* Yüklenen ve işlenen dosyaları herkes için ayrı tutmam gerekti. Bunun için uzun bir süre araştırma yaptım fakat tam çözümü bulamadım. Bunu Express sessionID lerini dosya adına iliştirerek çözdüm. “Best practice” nasıl olur bilmiyorum.
* Yüklenen dosyaları, kullanıcı siteyi terk edince nasıl silmem gerektiğini çözmem gerekti ama Heroku’nun static dosyları kendiliğinden temizlediğini gördüm.
* Heroku’ya aplikasyonu yüklemek alpha versiyonun’dan daha çok işlem gerektirdi. Bununun için aplikasyonun optimize edilmiş versiyonunu yapılandırmam gerekti.
* Birkaç sıkıştırma algoritmasını çıkartmak zorunda kaldım. Bazıları düzgün çalışmıyordu ve ürettiği hataları yakalayamıyordum.
* Çıkış resminin dosya tipi değiştiği zaman, resim galerisinde görüntüleyemedim. Bunun sebebi algoritmamın sadece giriş dizisini okumasıydı. Bu yüzden webp algoritmasını çıkartmak zorunda kaldım
* MongoDB için verileri kayıt edebildim. Fakat bunlarla bir işlem yapacak bir yol bulamadım ve zamanım yetmedi.

Sonuç olarak elimden geleni yaparak, portfolyoma ekleyebileceğim bir ürün çıkartmaya çalıştım.