

## Rast Mobile Bilgi Teknolojileri LTD. Şti Frontend UI Datagrid Task

### İzlenen Adımlar

- React.js için her zaman kullandığım bir klasör yapısı ile projeyi başlattım. VS Code ile kodlama çalışmalarımı ilerlettim.
- Gönderilen task'i inceleyerek nelere ihtiyacım olduğunu belirledim. Bu amaçla projeme belirli paketleri dahil ettim. Bunlar; axios, bootstrap, formik, yup, react-bootstrap, react-icons, react-router-dom, sass ve son olarak da devextreme-react paketlerinin kurulumunu yaptım.
- Tasarıma baktığımda tek bir template ve tek bir sayfa(home-page) yapısı oluşturmamın yeterli olacağını düşündüm. Bunlara ek olarak not-found-page sayfasını da dosyalarım arasına ekledim.
- Tasarım gereği navbar kısmı için Header adında bir component oluşturdum. Datatable yapısını şekillendirmek için de Datagrid adında bir component oluşturdum.
- Oluşturduğum dosyaların path'lerini, react-router-dom aracılığıyla belirledim.
- Helpers klasöründe, settings adıyla bir dosya oluşturup firma bilgilerinin sabit bir yerden çekilmesini sağladım.
- Assets klasörü içerisinde firma logosunu ve sass paketi ile bazı mixins ve variables değerleri oluşturdum. Bazı tag yapılarına özgü stilleri css ile ezemeyeceğim için variables dosyasında bs-fixed değerlerini değiştirdim.
- Mock api bağlantıları için api klasörünü oluşturdum ve endpointleri orada şekillendirdim.
- Header componentinde, responsive bir navbar oluşturdum. Bu işlemi gerçekleştirirken React Bootstrap bileşenlerini kullandım. Ayrıca sosyal medya ikonlarını react-icons paketinin fontawesome ikonlarını arasından belirledim. Ekranın 768px altına düşmesi durumunda sayfa linklerinin ve sosyal medya ikonlarının nasıl gözükeceğini şekillendirdim.
- Datatable bileşenleri için daha öncesinde React Data Table Component bileşenlerini kullandığım için DevExtreme React bileşenlerini çözmek çok vaktimi almadı. DevExtreme React bileşenlerini kullanarak datagrid yapısını oluşturdum. Datagrid bileşeni içerisinde, arama çubuğu, filtreleme butonu, yeni hesap ekleme butonu ve veri tablosunun yer almasını sağladım.
- Datagrid yapısı ile veri yüklenebilmesine, yeni verilerin eklenebilmesine, verilerin filtrelenebilmesine imkan tanıdım.
- Ayrıca, verilerin lokal depolamada saklanması da sağladım. Buna göre kullanıcının lokal deposunda daha önceden oluşturduğu veriler varsa, uygulama yüklendiğinde backend

bağlantısı çalışmayacak ve lokaldeki veriler kullanıcının önüne getirilecektir. Localden her ihtimale karşı verilerin alınamaması durumunda ise kullanıcının verileri backend'den getirilecektir.

- Verilerin backend'den getirilebilmesi için mockapi web sitesini kullanarak mock bir api bağlantısı ve field'ları oluşturdum. Oluşturduğum field değerlerini string olarak ayarladım ve ilgili endpointleri get ve post metotları kullanarak işledim. Ayrıca put ve delete metotlarını da api klasörümde şekillendirdim.

- Verileri front-end tarafında tutarken form validation yapısından yaralandım. Bunun için de formik ve yup paketlerini devreye soktum.

- Datagrid yapısı task'de istenilen görsele benzetmek için Figma'daki değerleri dikkatlice inceledim. Her bir detaya ilişkin gerekli css'leri kodladım ve projeyi responsive bir yapıya kavuşturdum.

- Bunun yanında uygulamaya firmanıza ait favicon ekledim.

### **Projeden Ne Anladım**

- Proje öncelikle bana datagrid yapıları için kullanabileceğim farklı paketleri kullanarak güzel bir deneyim kazandırdı.

- Restful API tasarım kurallarına dikkat ederken verileri tarayıcıyı yormadan kullanıcının localStorage'inden çekebileceğim bir yapıyı oluşturmamı sağladı.

- Datagrid yapıları için kullanılan paketlerin css ile düzenlenmesindeki zorlukları görmemi sağlayarak, bu zorlukların üzerine nasıl gideceğimi gösterdi.

- Responsive bir tasarım için media taglarının ne kadar önemli olduğunu tekrardan anlamamı sağladı.

İyi çalışmalar.