**PROJEYE DAHİL EDİLENLER**

* react-scripts 🡪 npm start ile projeyi çalıştırabilmek için kullandım.
* Css ve bootstrap kullanabilmek için bootstrap in sitesindeki linkleri projeme dahil ettim.
* Projemde ikon kullanabilmek için font awesome sitesindeki linki projeme dahil ettim.
* json-server 🡪 sanal bir api ile çalışmak için kullandım.
* axios 🡪 axios un get metoduyla verileri tek seferde json formatında almak için kullandım.
* react-router 🡪 single app bir yapı gerçekleştirdim. Routing yapısını kullanmam gerektiği için dahil ettim.
* Form-serialize🡪 Forma girilen name değerlerini bir nesne şeklinde almak için kullandım. Formda yazılan değerler url kısmına ekleniyordu (localhost:300/add?name… şeklinde) bunu önlemek için değerleri nesne şeklinde tutmak istedim.

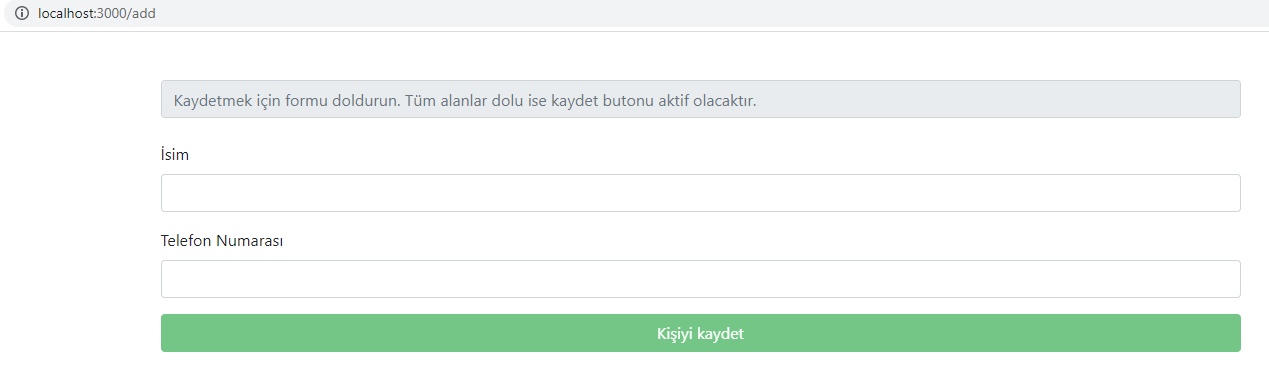
**PROJENİN AKIŞI**

1. Node.js kurulumu yaptım.
2. Komut isteminde “npx create-react-app projeadı” komutuyla yeni bir react projesi oluşturdum.
3. “npm start” komutunu kullanmak için öncelikle “npm react-scripts” komutuyla react-script i projeme dahil ettim
4. Src klasöründe “api” ve “components” olmak üzere iki klasör açtım. “api” klasöründe sanal veritabanımı , “components” klasöründe ise oluşturduğum componentleri ayrı ayrı dosyalarda tuttum.
5. “npm i json-server” komutuyla json-server ı indirdim. “api” klasöründe “db.json” adında bir javaScript dosyası oluşturdum. Verilerimi bu dosyaya yazdım.
6. “npm install axios” komutuyla axios u projeme dahil ettim. “api” klasöründe sanal api yardımıyla tuttuğum verileri axios un get metoduyla json şeklinde almış oldum.
7. En temel componentim “App.js”. Burada sanal api bağlantılarım, kullanıcı silme, ekleme, arama ve güncelleme fonksiyonlarımı oluşturdum.
8. Rehberdeki kişiler arasında arama yapabileceğim ve yeni kayıt ekleyebileceğim “SeacrhBar.js” componenti oluşturdum.

**Görseli:**



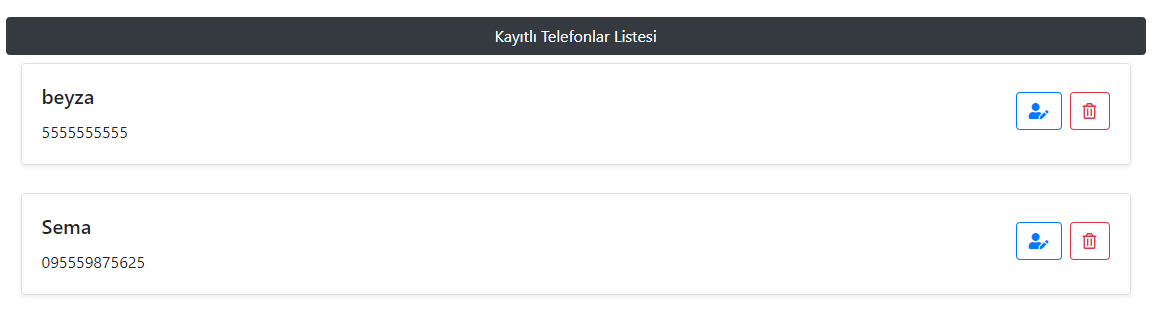
1. Yeni kayıt eklemek buton koydum. Url değiştiğinde farklı içerik gösterecek şekilde bir routing işlemi oluşturdum ve form yapımı localhost:3000/add url ine yönlendirdim. Burada 2 özelliği baz alarak çalıştım. Birincisi isim ve telefon numarasının boş olmaması durumu, ikincisi de telefon numarasının number tipinde ve başında 0 olmadan maksimum 10 karakter girilmesi durumu. Bu özellikler sağlandığında kaydet butonunun aktif olmasını sağladım.

**Görseli:**

1. Sonrasında rehberdeki kişilerin listelendiği bir “UserList.js” componenti oluşturdum. Burada bootstrapteki card yapısını kullandım. Rehberdeki kaydın adının ve numarasının gösterildiği ve kaydı silme/güncelleme butonlarının olduğu bir yapı oluşturdum. Bu componentimin içinde silme butonuna “App.js” componentinde oluşturduğum silme işlemini gerçekleştirdim.

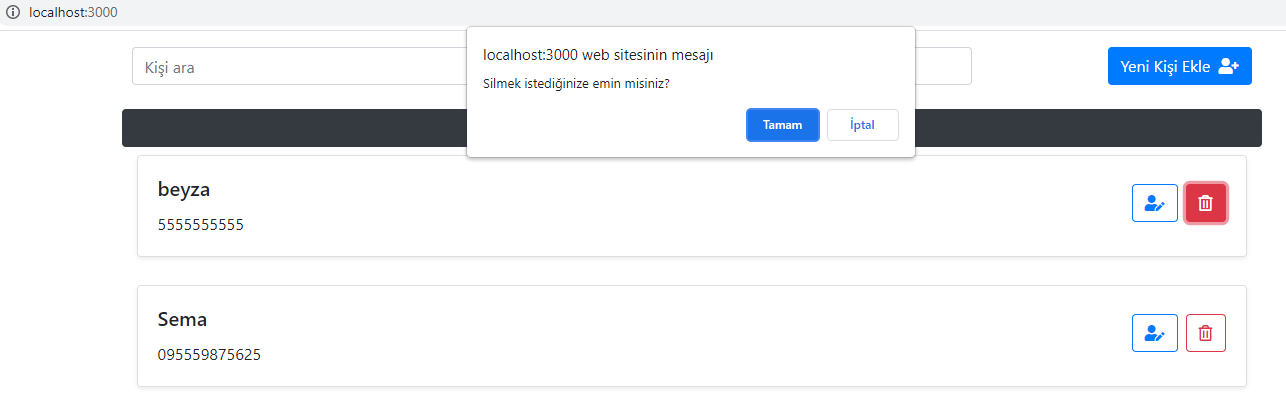
Burada listeme olurken isim çok uzunsa 50 karakterden sonrasında … olarak gözükmesini sağladım. Bunu truncate ile yaptım.

**Görseli:**



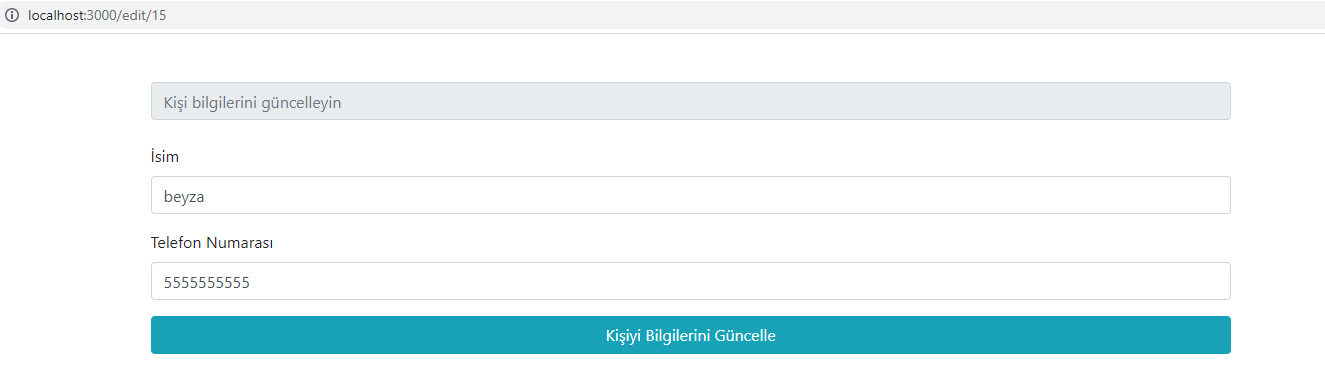
1. Silme işlemi gerçekleşirken direkt silmemesi önce sorması için Windows pop up ını kullandım.

**Görseli:**



1. Kişilerin listelendiği yapıda herhangi bir kayıttaki bilgilerin güncellenmesi işlemleri için “EditUser.js” componenti oluşturdum. Burada listede tıklanılan kişinin id bilgisine göre adını ve numarasını getiren bir form oluşturdum. Url değiştiğinde farklı içerik gösterecek şekilde bir routing işlemi oluşturdum ve form yapımı localhost:3000/edit/(seçilen id değeri) url ine yönlendirdim.

**Görseli:**



**PROJEYİ ÇALIŞTIRMA**

Visual Studio Code da terminale “npm start” dediğimde projem localhost:3000 de çalışır fakat verilerim json-server ile db.json dosyasından geldiği için verilerim gözükmez önce json-server ı çalıştırmam gerekir. Bu yüzden Visual Studio Code da terminale önce “npx json-server --watch src/api/db.json --port 3002” yazarak verilerimin 3002 portunda çalışmasını sağlıyorum. Daha sonra başka bir terminalde “npm start” dediğimde verilerimle beraber projem çalışıyor.