

Curriculum vitae

Personal data

Name: Berk E.

Technical Knowledge and Skills

Javascript, CSS, Angular, React, Node.js



Berk is an ambitious JavaScript/TypeScript developer utilizing modern web and hybrid mobile application frameworks; React, React Native, Ionic, Angular and StencilJS. Besides JavaScript/Typescript, SCSS, LESS, SASS are used as CSS preprocessors considering CSS design patterns to obtain clean and readable CSS code.

Education

September 2014 –
May 2018

Istanbul, Turkey
Economics with IT Management degree, Istanbul University

Work experience

January 2019 -
Present

Digital Architects, Turkey, Istanbul
Frontend Developer

Project and Responsibilities:

- Developed end to end frontend of ThyssenKrupp project CarValoo utilizing React, Redux and D3.js. In CarValoo project, custom SVG charts were needed to visualise car accidents, so D3.js was utilized. Some open source libraries were published by him after the project.

- Was a part of German insurance company (Honorar Konzept) ERP project utilizing Angular and Kendo UI. Tasks were connected mostly with CRUD operations, and the guideline of the project mainly focused on adopting Kendo from Teleric on both backend and frontend.

January 2017 -
January 2019

BKB Software Bilisim, Turkey, Ankara
Frontend Developer

Projects and Responsibilities:

- As part of a special ERP project Sezin Healthcare, he worked as a remote developer from out of town, holding daily agile meetings. All frontends including Android app (React Native), IOS app (React Native) and web frontend he designed frontend (Angular) and also published mobile apps for stores. In addition to the frontend, he also developed a Node.js push notification server for the project.
- For Sezin Healthcare, he developed a React Native app that integrates E-Work into a backend to receive digital information forms from patients to digitize the entire signature process prior to surgery or other surgery requiring patient consent. With this app, there is no need to sign on paper; instead, patients can sign on a tablet to show approval of a medical procedure. App for IOS and Android

Portfolio



Merhaba

Sezin Onam'a Giriş Yapın

Email

Şifre

[Şifremi Unuttum?](#)

Giriş Yap

Sorun mu yaşıyorsunuz?



Yukarıdaki linke tıklayarak
[Sistem Yöneticisi](#)'ne ulaşabilirsiniz.

2

Hasta Durumu

Buradan geçmiş hastalık ve diğer durumlarınız hakkında aşağıdaki bilgileri doldurabilirsiniz.

Kapalı alan korkunuz var mı?

☐ Evet ☒ Hayır

Şu anda gebeliğiniz / Şüphemiz var mı?

☐ Evet ☒ Hayır

Kalp pili veya beyin içinde uyarıcı bir alet veya insülin pompası taşıyor musunuz ?

☐ Evet ☒ Hayır

Vücudunuza kalp kapakçığı protezi ,stent(metal boru) rota kulak veya iç kulak protezi takıldı mı ?

☐ Evet ☒ Hayır

Vücudunuzda protez veya çıkabilen diş protezinin var mı?

☐ Evet ☒ Hayır

Vücudunuz da özellikle gözünüzde metal var mı?

☐ Evet ☒ Hayır

Daha önce kurşunlama veya saçma, mayın gibi metal yaralanması geçirdiniz mi?

☐ Evet ☒ Hayır

Stent var mı? Varsa ne zaman takıldı?

Form Onayı

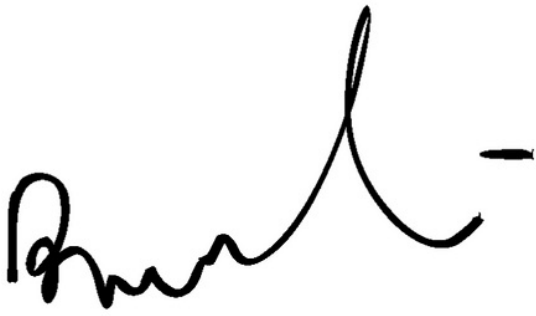
Buradan seçili onam formunu onaylayabilirsiniz.

Sayın hasta /vekili / kanuni temsilcisi;
Sağlık durumunuz ve size önerilen tıbbi, cerrahi, ya da teşhise yönelik her türlü işlemler ve bunların alternatifleri, faydaları, riskleri hatta olabilecek zararları hakkında bilgi sahibi olmak ve tüm bunları ya da bir kısmını reddetme ya da kabul etme veya yapılacak işlemleri herhangi bir aşamasında durdurma hakkına sahipsiniz. Sizden okuyup, anladığınızı istediğimiz bu belge, sizi ürkütmek veya yapılacak tıbbi uygulamalardan uzak tutmak için değil, bu uygulamalara rıza gösterip göstermeyeceğinizi belirlemede, sizi bilgilendirmek ve onayınızı almak için hazırlanmıştır.

YAPILACAK İŞLEM:

MR vücudunuzun görüntülerinin elde edilmesini sağlayan çok gelişmiş bir cihazdır. MR cihazının esas üstünlüğü röntgen ve BT* de kullanılan zararlı ışınlara (X ışınları) ihtiyaç duymamasıdır.

* Buraya 'Okudum, Anladım' yazarak imza atabilirsiniz.



Kaydet

Digital Signature System / Medical (Mobile App)

January 2019 - July 2020

Website: <https://ework.bkbsoft.com>

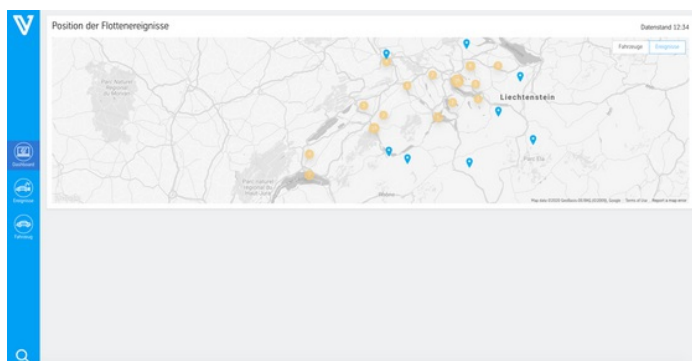
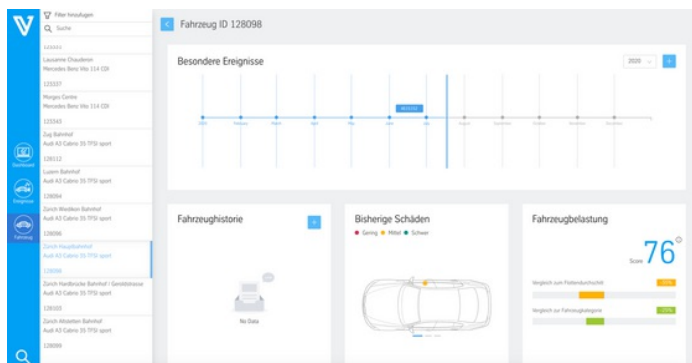
The project was about taking digital signature for patients' consent on having medical imaging scans. The application consisted of a big form getting all answers about the patient's status which is related to medical imaging such as "Does she/he have a metal implant on his/her body?", "Does he/she have claustrophobia?" etc. After reviewing all these related questions; the patient states his/her consent via a digital signature on a tablet device which can be both IOS or Android.

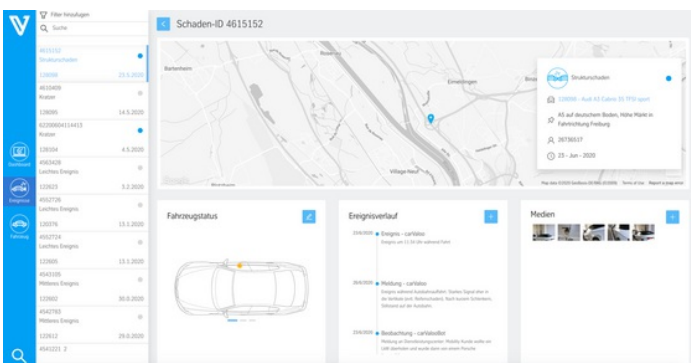
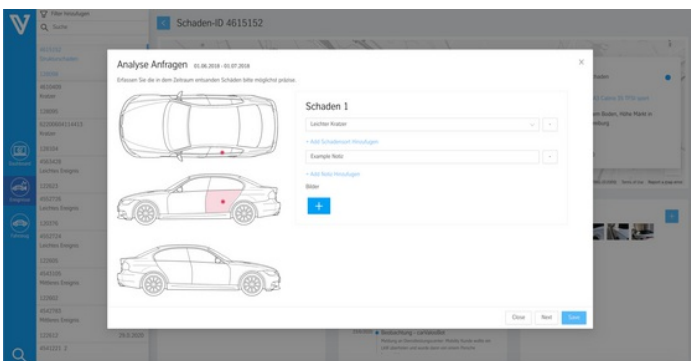
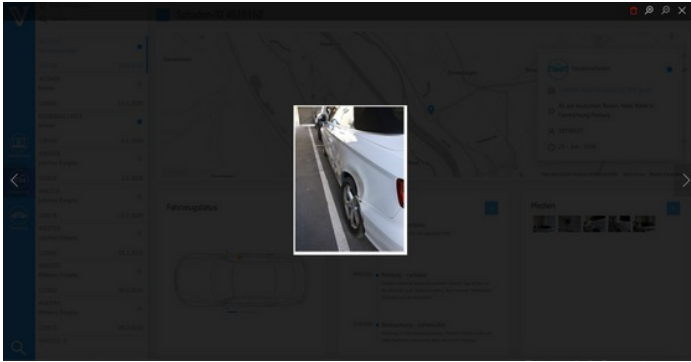
The project was a submodule of e-work project; so the team was the same as the e-work; but the project's hard part was only the frontend part because backend required only two API endpoints which are for creating the form with its data and listing all data with pagination on admin dashboard of e-work.

The hard part was to handle all this big form with stepper form logic in frontend with its signature as a .png file and send all these data to backend after the form is completed. Besides

stepper, the form also required different questions depending on the medical imaging device, because MRI device requires different questions to ask than CTI device. Therefore, I have set up Redux as global store and Axios for http requests to the backend. Users could sign in with the same authentication credentials as they do in e-work; and they would fill up all the form according to what patients tell them about their status. After filling all the form, a signature division gets opened and the patient can sign with his/her finger or tablet pen just like sign in on a real paper. After the signature data is converted into base64 data; all the form data including the signature is sent to the backend with a POST API endpoint; and all the forms can then be viewed by authorized personal in admin dashboard. After the form is sent, all the global store resets to the default and the same process can be repeated again for a new patient. As a result of the project, there is no need for paper for signing anymore; all the consent form process can be both applied and tracked digitally. No paper is lost anymore and no paper is wasted.

Technologies used:
React Native





CarValoo / Car Management IOT Application

January 2019 - July 2020

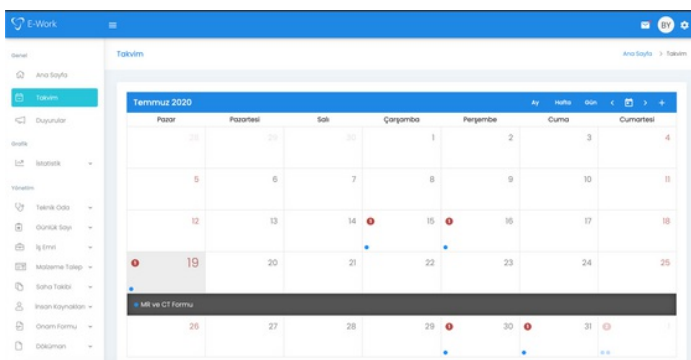
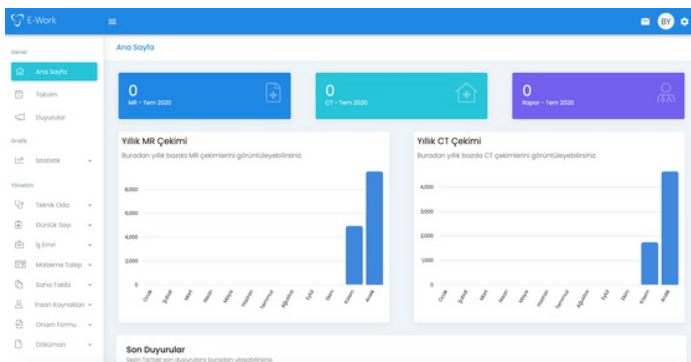
Website: <https://www.thyssenkrupp-automotive-technology.com/en/products-and-services/carvaloo>

The project was about tracking cars from the connected IOT devices which sends signal about any kind of damage. The incoming data is served to the web client which provides engineers with the latest data of any car. In the project, I have developed all React web application with all data coming from ASP.NET Core backend. I have only developed frontend of the application and I only know there are 2 backends in our company. I don't know the size of the team on IOT part of the project. So, we have developed the project with 2 backend and 1 frontend in the company. The database access with the current data was given to our team; and I have coded all frontend according to the UI/UX design that the company sent me. I have coded all CSS of the application

with flex layout and constructed all state management architecture by myself in frontend. The project required so much complex data visualization; therefore I have utilized D3.js for SVG animations and data chart construction. I have even open sourced a chart developed during the project which required zooming in-out in yearly time scale to daily time scale; connecting its state with third party React components and clustering data just like in maps. I have also coded clustering algorithm for the data. For instance, if datas get too close to each other, I needed to prevent their intersections; so a custom clustering logic is implemented by me. Via this clustering logic on client side, the backend sends all the data to frontend; and frontend handles all the clustering and data visualization logic by itself.

In the project, tracking of IOT sensor data is made easy with custom interactive D3.js charts made with SVG. Besides charts, the car is also coded with SVG and the part of the vehicle which sends the sensor data can be viewed easily with this SVG visualization.

Technologies used:
React, D3.js



At first, the project required role based authentication; so every route and some components in

the routes should have been rendered depending on the users' roles. So I have used Redux on React Native and NgRx on Angular to store user roles to decide whether the user has access to specific routes or components to view or not. I have written custom directives in Angular to decide rendering of components and gave some parameters to routes to decide if routes are accessible or not. On React Native, I have coded some higher order components which have been utilizing Redux state to decide if the component should be rendered for the user or not. In both of the applications, the global store saved all the role info of users.

We have implemented human resources management in the project which provided all employees to take day off via both mobile application and web interface. By the multi-tenancy architecture in the project, authorized managers would also be able to approve or reject day off claims from employees and also human resources manager would be able to track every single employees day off statuses with statistical data. Besides human resources, the project also included calendar in which managers could easily assign tasks, which triggers push notification on mobile and web clients, for specific employees and track the latest status of the tasks depending on what employees stated via the web and mobile clients about the assigned tasks.

Technologies used:
Angular, React Native