**Projekt Labor**

**Alkalmazásfejlesztés web alapú mobil platformra**

Baumgartner János Tancsics László

témavezető Németh Ádám

Abafalvi Balázs

csapattagok

Tartalom

[1. Felhasználói dokumentáció 3](#_Toc52405657)

[2. Fejlesztői dokumentáció 4](#_Toc52405658)

[2.1 Feladat leírása 4](#_Toc52405659)

[2.2 Az alkalmazott fejlesztői eszközök 5](#_Toc52405660)

[2.2.1 React 6](#_Toc52405661)

[2.2.2 Ionic 6](#_Toc52405662)

[2.3 Adatmodell leírása 7](#_Toc52405663)

[2.3.1 Firebase 7](#_Toc52405664)

[2.3.2 Adatbázis felépítése 7](#_Toc52405665)

[2.4 Algoritmusok 8](#_Toc52405666)

# 1. Felhasználói dokumentáció

# 2. Fejlesztői dokumentáció

## 2.1 Feladat leírása

A projekt során egy webes alkalmazás fejlesztése a cél, amely megkönnyíti, valamint hatékonyabbá teszi a kis- és középvállalkozások (KKV) számára az egyes szolgáltatások használatát

Az alkalmazás segítségével lehetőség nyílik a felhasználók számára az árajánlatok kérése időponttal együtt. Az admin felhasználó pedig ezeket a kéréseket el tudja fogadni, utasítani, esetlegesen „talonba” helyezni. Továbbá lehetőség nyílik admin oldalról a statikus oldalak tartalmának szerkesztése.

Manapság a BYOD (Bring Your Own Device) trend hatalmas népszerűségnek örvend. Mindenki a saját eszközein próbálja intézni az ügyeit. Ezen eszközök közül a legnépszerűbb kétségtelenül az okostelefon, hiszen könnyen hordozható, és szinte mindent meglehet vele csinálni, amit egy asztali számítógéppel is. Így nem is volt kérdés, hogy ennek a webalkalmazásnak mobilapplikációként is működnie kell.

## 2.2 Az alkalmazott fejlesztői eszközök

**Fejlesztői környezet**

* Atom Editor

**Programozási nyelvek**

* HTML5
* CSS
* Javascript

**Adatbázis rendszer**

* Firebase

**Javascript könyvtár**

* React

**Keretrendszer**

* Ionic

**Szövegszerkesztő**

* Word

**Képszerkesztő**

* GIMP 2

### 2.2.1 React

A **React** egy Facebook által kifejlesztett nyílt forráskódú JavaScript könyvtár, amit felhasználói felületek programozására használunk. Az MVC (Model View Controller) programramtervezési mintából a React a *„V-t”* képezi. Tehát csak React-et használva nem lehet web alkalmazást fejleszteni!

Komponensekből tevődik össze. Ezek JavaScript függvények, tetszőleges számú inputtal és egy outputtal. JSX (JavaScript XML) szintaxissal íródnak. A komponensek fejlesztése ES6 szabvány szerint történik, amely a következőket határozza meg:

* A komponensek tetszőleges számú bemenetet fogadhatnak és ezeket az úgynevezett „*props”* nevű objektumba tárolják.
* Mindig kell lenni-e egy *„render”* metódusnak. Ez a metódus határozza meg, hogy mi legyen megjelenítve, amikor a komponens meghívódik.
* Minden komponens egy belső állapotot menedzsel. Amikor az állapot változik akkor az adott komponens *„render”* függvénye meghívódik.

React könyvtár segítségével íródott alkalmazások teljesítménye kivételesen magas. Ennek az oka nagyon egyszerű. A render-elés Virutal DOM (Document Object Model) segítségével történik. Tegyük fel, hogy sok kis *„View”* változást szeretnénk eszközölni. Eredetileg ezek a változások azonnal, közvetlenül a DOM-on hajtódnak végre. Emiatt a DOM folyamatosan frissülni fog🡪 az alkalmazás teljesítménye romlik. React megoldása: Ezek a *„view”* módosítások tárolódjanak el a VDOM-ban. Ezután jön egy algoritmus, amely először összehasonlítja a a Virtual DOM jelenlegi és előző állapotát, majd végrehajtja azt a minimum számú módosítást az eredeti DOM-on, ami ténylegesen szükséges, a kívánt állapot eléréséhez. Felhasználói interakciónál is ez történik!

### 2.2.2 Ionic

## 2.3 Adatmodell leírása

### 2.3.1 Firebase

A **Firebase** egy szolgáltatáscsomag a Google-től. Segítségével különböző backend feladatok oldhatóak meg könnyedén. Ilyen például a felhasználók beléptetése akár Google vagy Facebook azonosítással, adatok felhős tárolása, vagy éppen lehetőséget nyújt a felhasználók számára, hogy kedvencek közé rakhassanak tartalmakat.

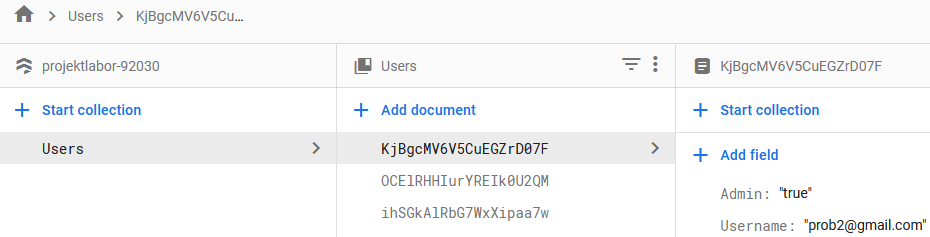
Firebase szolgáltatásai nem csak a fejlesztésre koncentrálnak, számos lehetőséget nyújt arra, hogy mobil applikációnk mérhető növekedést érjen el:

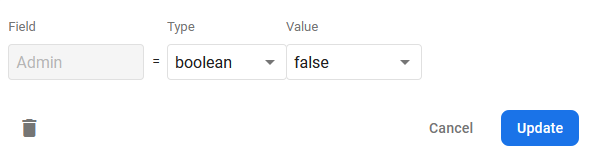
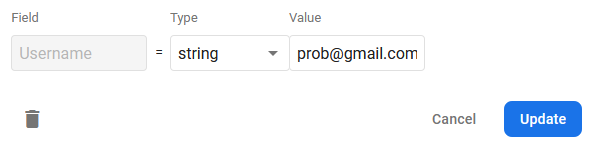
* Mérhetők/elemezhetők a felhasználók interakciói
* Az appunkat aktívabban használó közönség felé kommunikálhatunk új alkalmazás verziók kiadása nélkül.
* Fizetett kampányokkal szerezhetünk további felhasználókat

Ez egy „Realtime Database”, tehát az adatok szinkronizálása valós időben történik. NoSQL felhőalapú adatbázist használ. Itt tárolódnak az adatok. JSON alapú, így könnyen belátható, hogy egyszerű adatszerkezetek tárolására alkalmas ez a rendszer. **Használjunk SQL alapú rendszert bonyolultabb adatszerkezetek tárolása esetén!**

### 2.3.2 Adatbázis felépítése

A projekt során három alapvető authorizációs adatot tárolunk, amelyek a következők: Username (Email cím), jelszó, valamint a felhasználó jogosultsági szintje (Admin).



A Username és a password string, valamint az Admin mező boolean típusú.

## 2.4 Algoritmusok