**Debreceni Szakképzési Centrum**

**Baross Gábor Középiskolája és Kollégiuma**

**Debrecen, 4030**

**Budai Ézsaiás u. 8/A.**

**A szakképesítés azonosító Száma:54 213 05**

**Szakképesítés megnevezése: Szoftverfejlesztő**

**Záródolgozat címe:**

**Jegyzetelő applikáció telefonra**

**Záródolgozat készítője:**

**Gazdag Ádám**

**Konzulens: Dr. Várbíróné Nahaji Anikó**

**Konzultációs időpontok:**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Debrecen, 2024.**

Tartalomjegyzék

[Bevezetés 3](#_Toc163205943)

[Felhasználói dokumentáció 4](#_Toc163205944)

[Feladatspecifikáció 4](#_Toc163205945)

[Rendszerkövetelmények 4](#_Toc163205946)

[Program telepítése, elérése 4](#_Toc163205947)

[Program használatának leírása 5](#_Toc163205948)

[Fejlesztői dokumentáció 10](#_Toc163205949)

[Témaválasztás indoklása 10](#_Toc163205950)

[Alkalmazott fejlesztői eszközök 10](#_Toc163205951)

[Rendszerterv 11](#_Toc163205952)

[Adatmodell leírása 11](#_Toc163205953)

[Program leírása 13](#_Toc163205954)

[Részletes feladatspecifikáció, algoritmusok 13](#_Toc163205955)

[Tesztelés 20](#_Toc163205956)

[Továbbfejlesztési lehetőségek 20](#_Toc163205957)

[Összegzés 21](#_Toc163205958)

[Irodalomjegyzék 21](#_Toc163205959)

[Eredetiségnyilatkozat 22](#_Toc163205960)

# Bevezetés

A dolgozat témája egy jegyzetelő applikáció, amiben fel tudunk jegyezni bármit, amit szeretnénk. Mikor ezt az applikációt terveztem, azt kerestem, hogy miben lehet több a többi jegyzetelő applikációval szemben. Mivel nem jutott eszembe egy sem, úgy döntöttem, hogy elképzelek egy általam ideálisnak képzelt applikációt, ami tartalmaz minden olyan funkciót, ami esetleg számomra előnyös lehet.

Mindig is vágytam egy egyszerű jegyzetelő applikációra, mert manapság már nem talál az ember ingyenesen olyat, ami egy tartós eszköz lenne számára anélkül, hogy kiadná érte a pénztárcája teljes tartalmát havonta. Ekkor kezdtem bele ebbe a projektbe, és habár eleinte nem volt semmi ötletem, egész szépre, egyszerűre, és használhatóra sikerült elkészítenem. Mielőtt rátérnék a tervezési választásaimra, szeretném egy kicsit összehozni a szenvedélyeim a projekttel. Számomra elsősorban fontos, hogy hatékonyan, és gyorsan tudjak dolgozni, és ennek hála mindig találok olyan megoldásokat, illetve eszközöket, amik elősegítik ezt. Volt egy ilyen nap is, amikor otthon dolgoztam egy saját projekten, és a billentyűzetem komplexitása frusztrációt okozott munka közben. Ez azt jelenti, hogy gyorsabban jár az agyam, mint amilyen gyorsan dolgozom. Ekkor elkezdtem keresni egy megoldást, hogy hogyan tudnék gyorsabban dolgozni. Ezt követően rátaláltam a Vim nevezetű szövegszerkesztőre, és társaira. Tetszett a felépítése, mivel több rétegből állt fel. Az első réteg az, amikor megvizsgálhatjuk a dokumentumot, és esetlegesen szerkeszthetjük. Illetve egy másik réteg, ahol teljes eszközkészletet kapunk, és billentyűkombinációkkal tudunk gyorsabban navigálni, és egyéb akciókat végrehajtani. Ennek a szerkesztőprogramnak a felépítése nagyon tetszett, és szerettem volna belevinni az alkalmazásba. A programban megtalálható ez a réteges felépítés, a jegyzetek szerkesztésénél. Amikor rákattintunk egy jegyzetre, akkor láthatjuk a tartalmát, és egyéb olyan akciókat tudunk végrehajtani, amik ennél a folyamatnál relevánsak.

Ezek a lehetőségek viszont csak regisztrált, vagy bejelentkezett felhasználóknak elérhetőek. Gyakorlatilag tehetetlen a felhasználó, míg be nem lép a saját profiljába. Mivel az applikáció szerveroldali, és adatbázissal kommunikál, azok, akik hozzáférnek az adatbázishoz, csak mi vagyunk, a készítők. Fontosnak tartom megjegyezni így zárásként, hogy milyen képességeink vannak az applikációnk fölött, félreértések elkerülése érdekében. Képesek vagyunk jegyzeteket, és felhasználókat törölni kérésre. Láthatjuk a felhasználók nevét, email címét és jegyzeteit. A jelszók titkosítva vannak, és nem láthatóak. Mi sosem törlünk, vagy változtatunk a jegyzeteken, illetve felhasználókon kérés nélkül.

# Felhasználói dokumentáció

## Feladatspecifikáció

A program legkiemelkedőbb tulajdonsága, hogy képes felismerni a markdown szintaxist a jegyzetek tartalmában, ezáltal a felhasználó széleskörűbben tudja személyre szabni, formázni, illetve szervezni a jegyzeteit. A program felépítését, a bevezetésben említett Vim szövegkezelő program inspirálta. Manapság sok weboldalfelület-tervező könyvet, illetve blogot olvasok, ami segített a felület kidolgozásában. Az applikáció nem tartalmaz adminisztrációs felületet, illetve profilokat. Ehhez köthető feladatok elvégzésére, közvetlenül az adatbázis parancsai állnak rendelkezésre.

A program fő funkciói:

* Felhasználók autentikálása
* Jegyzetek megjelenítése adatbázisból
* Jegyzetek szerkesztése, kapcsolódó szükséges tartalmak frissítése
* Jegyzetek elrendezése adatbázisból nyert adat alapján
* Markdown szintaxisok renderelése, és megjelenítése
* Jegyzetek törlése, és készítése

## Rendszerkövetelmények

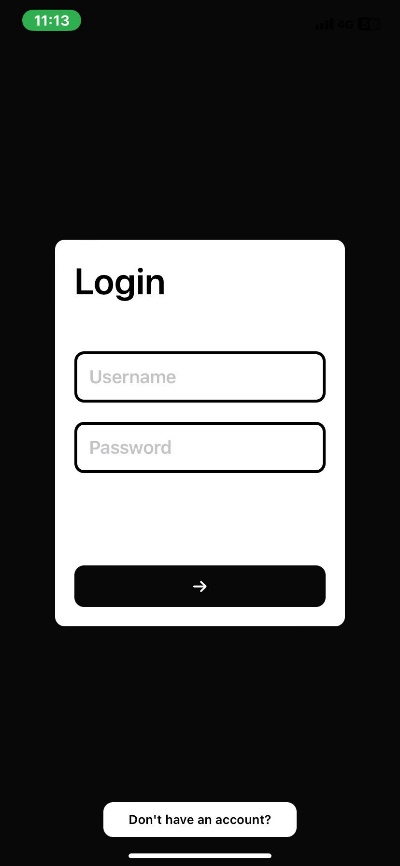
A szoftver alacsony hardverigényű, képes elfutni alacsonyabb teljesítményű eszközökön is. A felület szándékosan reszponzív kialakítású, és az eddigi tesztek alapján megtartja ezt a tulajdonságát. Az alkalmazás kisebb-nagyobb esztétikai változtatásokon keresztül az iOS és Android operációs rendszereken is egyaránt megfelelően működik. Az applikáció internet elérést igényel.

## Program telepítése, elérése

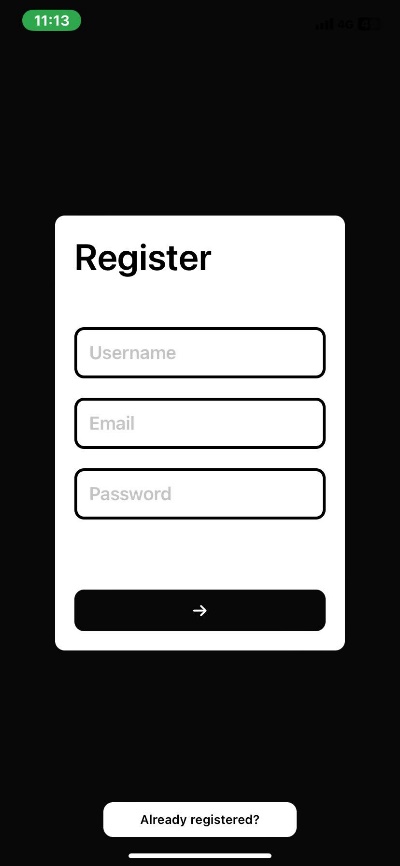
A telepítés egy letölthető APK fájllal történik. A letöltés után telepítés szükséges. Internetes elérés esetén az applikáció lehetőséget nyújt a bejelentkezésre, illetve regisztrációra. Sikeres regisztráció, és internet elérés feltételében használható az applikáció rendeltetésszerűen.

## Program használatának leírása

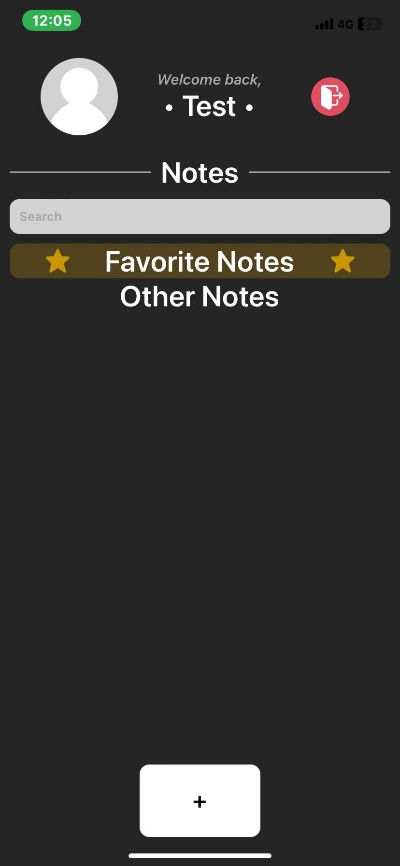
A program elindításakor a következő kijelzőt látjuk:



Ezen a kijelzőn be tudunk lépni a fiókunkba, he van olyan. Két mezőt láthatunk melyek az „Username”, és a „Password” címkékkel vannak ellátva. Az „Username” szövegdobozba a felhasználónevünket kell írni, míg a „Password” címkével ellátott szövegdobozba, a felhasználó által, regisztrációkor beállított jelszót kell megadni. Ha a felhasználónév, és a jelszó helyes, a program beenged minket a fiókunkba. Ha nincs még fiókunk, a lent található „Don’t have an account?” azaz „Nincs még fiókja?” gombra kattintva, átdob minket egy regisztrációs lehetőségre:



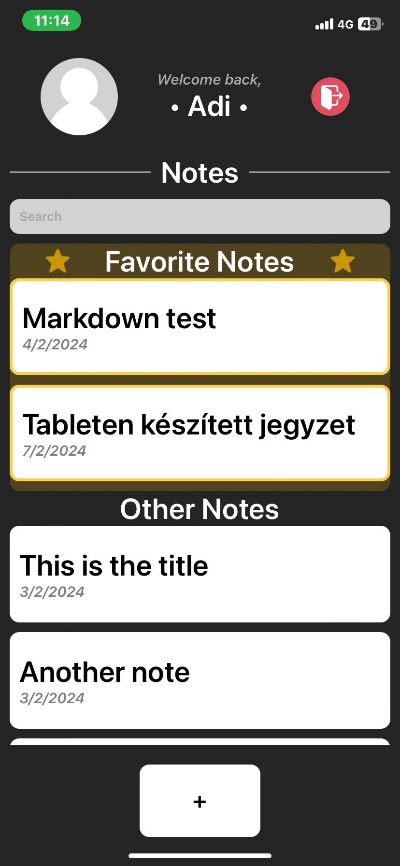
Ezen a felületen szintúgy meg kell adnunk a felhasználónevünket („Username”), és jelszavunkat („Password”), viszont itt az email cím („Email”) megadása is szükséges. Az email cím arra szükséges, hogy egy felhasználónak ne lehessen több fiókja. Így egy email címhez, csak egy darab fiók tartozhat. Az email címet másra nem használjuk. Sikeres regisztráció esetén, a program automatikusan beléptet minket a regisztrált fiókunkba. Ekkor a következő látvány fogad minket:



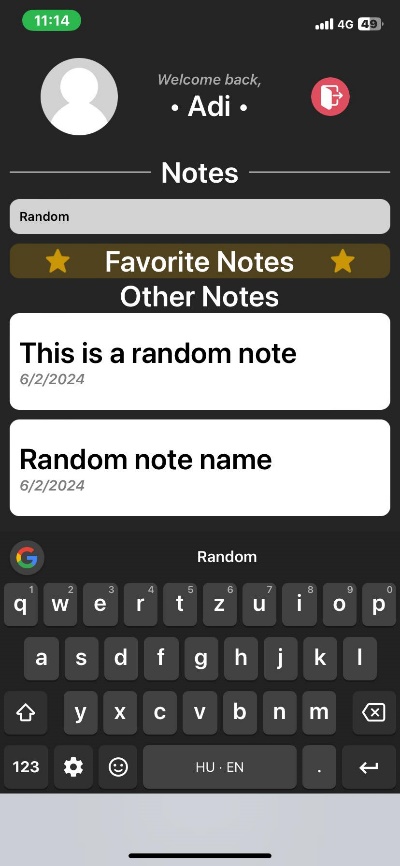
A képernyő felső részén láthatjuk a felhasználónevünket, és a piros kijelentkezés gombot. A kijelentkezés gombra kattintva, az applikáció kijelentkeztet, és visszavisz a bejelentkező felületre. A „Notes” cím alatt találhatjuk a keresősávot, a kedvencnek jelölt jegyzeteket, illetve a többi besorolatlan jegyzetet. A fenti képen jelenleg egy teljesen új fiókot látunk jegyzetek nélkül. Ahhoz, hogy jegyzeteket hozzunk létre, a lenti „+” gombra kell kattintani. Ekkor az applikáció átvisz minket egy másik képernyőre, ahol megírhatjuk az úgy jegyzetünket. Ez a következőképpen néz ki:



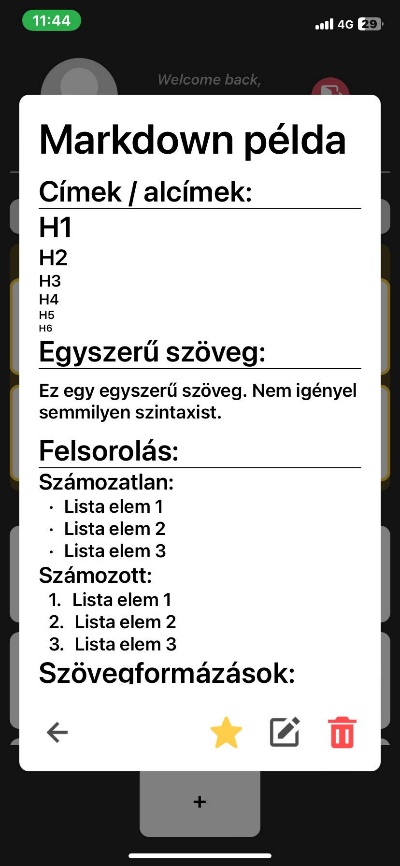
A „Title” helyére írhatjuk a jegyzetünk címét, a tartalmát pedig a „Your note starts here…” címkére kattintva kezdhetjük meg írni. A lenti sávban láthatunk egy nyilat, ami **mentés nélkül** visszavisz minket a fő oldalra. Mellette található egy szem, ami akkor használatos, ha markdown szintaxist használunk a jegyzetben, és elő szeretnénk nézni. Erről egy kicsit később bővebben. A legutolsó zöld mentés ikon, pedig elmenti a jegyzetet, és visszavisz minket a fő oldalra. Mentés utáni visszatéréskor látnunk kell a jegyzetet. Ha mégsem így van, frissítsünk rá a jegyzetekre, felfele húzással, szükség esetén többször. Előfordulhat, hogy ez sem válik be, akkor jelentkezzünk ki, és újra be. A következő példákban egy olyan fiókot fogok mutatni, ami már több jegyzetet is tartalmaz.



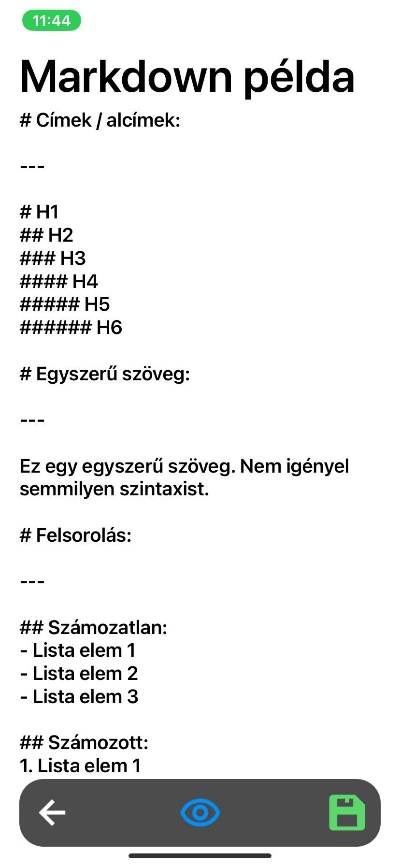
Ez egy olyan fiók, ami számos jegyzetet tartalmaz, többek között olyat is, amik kedvenceknek lettek jelölve. A kedvencnek jelölt jegyzetek priorizáltan, felül jelennek meg a többi jegyzethez képest. A „Search”-el jelölt keresőben tudunk keresni a jegyzetek címében, illetve tartalmában:



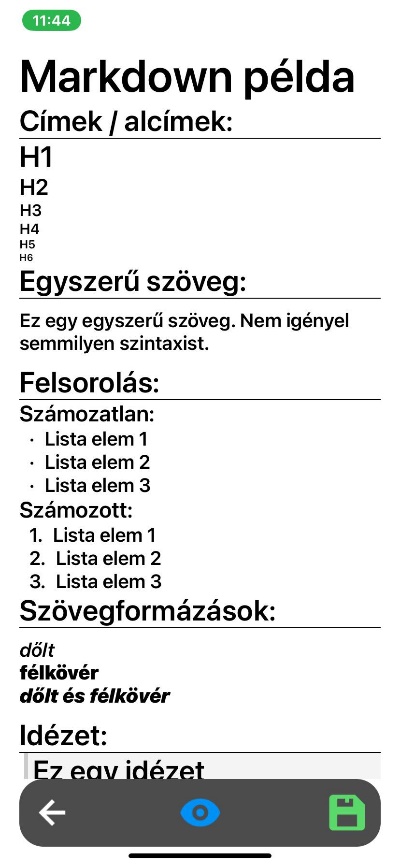
Ha egy jegyzetre rákattintunk, az előhozza a kiválasztott jegyzet gyorsnézetét:



Ezen a képen, legfelül, a jegyzet címét, alatta a tartalmát láthatjuk. Ezt követően alatta, pedig az előnézethez relevánsnak talált funkciókat, mint például a vissza nyilat, amivel visszaléphetünk, a csillagot, amire rákattintva betehetjük az adott jegyzetet a kedvencek közé, a szerkesztés gombot, amivel a szerkesztő módba válthatunk, és módosíthatjuk a jegyzetet, végezetül pedig a törlés gomb amire rákattintva egy megerősítési kérelmet kapunk, amit elfogadva törölhetjük a jegyzetet. Szerkesztéskor a következőt láthatjuk:



Felül szintúgy a cím látható, alatta a jegyzet tartalmával. Ez a mód ugyan úgy működik, mint az új jegyzeteknél dokumentált felület. Ez a jegyzet viszont egy tökéletes példa arra, hogy megmutassam a markdown szintaxis használatát, ami még többet ki tud hozni a jegyzetekből. A példán látható jegyzet tartalma ezt a szintaxist használva mutatja be ezen használatának a potenciálját. Az utóbb említett kék szemre kattintva előnézhetjük a markdown formázását:



# Fejlesztői dokumentáció

## Témaválasztás indoklása

A programmal jegyzeteket tudunk írni, illetve elmenteni a felhőben. Azért választottam ezt a témát, mert szerettem volna egy személyre szabott, saját igényeknek megfelelő jegyzetelő applikációt, amit nem kér el a szolgáltatásáért több tízezer forintot havonta.

## Alkalmazott fejlesztői eszközök

Az applikációt React Native Expo segítségével hoztam létre, és készítettem el. A Visual Studio Code fejlesztői környezetet használtam. Az adatbázist a Xampp használatával hostoltam lokálisan, tesztelésre, később publikusan, használatra. MySQL-t használtam az adatbázis kialakítására, illetve elemeinek lekérdezésére. Az applikáció létrejöttét továbbiakban a következő moduloknak köszönhető:

* **bcryptjs** – A jelszavak titkosítására, és visszafejtésére
* **cors** – A felhasználó-backed kapcsolat biztonságára
* **express** – A backend útvonalainak kiépítésére, adatbázissal kapcsolatos felhasználói műveletek lebonyolítására, és a backend hostolására
* **react navigation** – Az applikáción belüli navigáció leegyszerűsítésére
* **mysql** – Adatbázis, és backend közötti kapcsolatra
* **nodemon** – Effektív tesztelésre

Továbbá az applikációban használt ikonok és elemek a következő weboldalakról származnak:

* <https://www.flaticon.com/>
* <https://boxicons.com/>

Minden médiaelem 144px-es felbontással lett letöltve, és használva. Bizonyos ikonok színezve, és méretezve lettek igény szerint.

## Rendszerterv

### Adatmodell leírása

Az applikáció egy egyszerű adatbázissal van összekötve, ami tárolja a szükséges adatokat. Az adatbázis a következő táblákkal és felépítéssel rendelkezik:

A képen szöveg, képernyőkép, Betűtípus, diagram látható

Automatikusan generált leírás

Van egy „note”, illetve egy „user” tábla. A „note” tábla tartalmazza a felhasználók jegyzeteit, míg az „user” tábla a felhasználók adatait.

Minden felhasználó rendelkezik:

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

* UserID – Egyedi azonosító
* Username – Egyedi felhasználónév
* Email – Egyedi email cím
* Password – Titkosított jelszó

A felhasználók az egyedi azonosítóik alapján tudjuk beazonosítani. Az egyedi azonosítóik emellett meghatározzák, hogy melyik jegyzetek tartoznak hozzájuk. Minden jegyzetnek van:

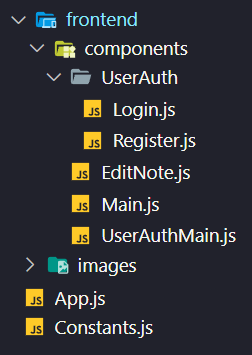
A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

* NoteID – Egyedi jegyzet azonosító
* UserID – Jegyzethez tartozó gazda
* Title – Jegyzet címe
* Content – Jegyzet tartalma
* IsFav – A jegyzet kedvenc státusza (igaz, hamis)
* CD – A jegyzet készítési dátuma
* LMD – A jegyzet utolsó módosításának dátuma

### Program leírása

A program felépítése a következő:



A fájlszerkezetet a frontend mappa tárolja. A components mappa tárolja a Main.js-hez szükséges js componenteket. Ez a mappa tartalmazza az UserAuth mappát, amiben megtalálható a Register.js, és a Login.js. A components mappa tartalmazza még az EditNote.js-t, a Main.js-t, és az UserAuthMain.js-t. Ezen felül van egy images, azaz képek mappa, ami nyilvánvalóan a képeket, ikonokat tartalmazza. Az applikáció gyökere pedig az App.js, amiben lebonyolítom az applikáció fő folyását. Végleg pedig a Constants.js tárolja azokat az API útvonalakat, amikre tudok hivatkozni HTTP kérelmek küldésekor.

## Részletes feladatspecifikáció, algoritmusok

Elsősorban a bejelentkezést írtam meg. A bejelentkezést az UserAuth mappán belüli komponensek, és az UserAuthMain.js komponens bonyolítja le.

**UserAuth/Login.js**

A képen szöveg, képernyőkép látható

Automatikusan generált leírás

A Login.js komponens tartalma. Ez a bejelentkezési felület, ahol 2 TextInput, és Pressable található. Mind a két TextInput szöveges értékeket fogad, amiket a Pressable megnyomásával felhasznál, a handleLogin metódusban:

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

A handleLogin, ami nevéből következtethetően kezeli a bejelentkezést, elküld egy POST requestet az autentikációs végpontra, ahol a backend ezt lekezeli. Ha bármilyen hiba merül fel folyamat közben, azt a hozzá tartozó HTTP hibakóddal, és üzenettel együtt visszaküld a felhasználónak. A regisztrációs folyamat hasonlóan zajlik. Gyakorlatilag csak a Login.js van kibővítve egy TextInput-tal az email miatt, illetve más végpontra küld POST kérelmet. A két komponenst az UserAuthMain.js komponens köti össze, ami hozzáad egy Pressable gombot a képernyő aljához. Ezt a gombot megnyomva tudunk váltani bejelentkezés és regisztráció között.

**components/UserAuthMain.js**

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

A komponens tartalmaz egy isLogin változót, ami csak igaz, vagy hamis értéket tud magában tárolni. Az alul hozzáadott gomb megnyomásával ez a változó negálja saját magát, egy kapcsolót létrehozva. Ennek segítségével beleraktam a Login, és a Register komponenst egy kondicionális feltételbe, ami az isLogin változót veszi alapul. Ha a feltétel igaz, a Login komponenst látjuk, viszont, ha hamis akkor a Register-t. Ezek mellett mind a két komponensbe belevezetek egy singIn, azaz bejelentkezési metódust, amit szintúgy bele van vezetve az UserAuthMain komponensbe. Ez azért szükséges, hogy máshol tudjam lekezelni a bejelentkezési logikát. Ez a logika az applikáció fő gyökerében található, az App.js-ben:

**App.js**

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Mivel az adatokat a bejelentkezéshez a Login, illetve Register komponensből szerzem, ezért el kell, hogy juttassam hozzájuk, hogy a beléjük írt értékeket le tudjam kezelni a signIn metóduson belül. Ezen a metóduson belül el is mentem a felhasználót a telefon memóriájába, így nem kell minden alkalomkor bejelentkeznie a felhasználónak, ha belép az applikációba. Ezt követően beállítom a bejelentkezett felhasználót a setUser metódussal. Ez azért szükséges, mert az alkalmazás fő része nem az App.js-ben van. Valójában a Main.js-ben található, aminek argumentumként átpasszolom az user-t. Így az megkapja a szükséges adatokat ahhoz, hogy a bejelentkezett felhasználóhoz tartozó jegyzeteket betöltse. Az App.js-ben is hozom létre a React Navigation navigációs konténerét, amihez hozzáadok minden olyan szükséges komponenst, ahova navigálhatok majd az app működése során:

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

Ahogy a fenti képen is látható, a bejelentkezés, és az applikáció fő funkcionalitása külön van szedve. Így ha véletlenül navigálni akarnék az autentikációs kijelzőkre, bejelentkezett felhasználó esetében, az nem működne. Az applikáció fő része (Main.js) egy hosszú, és összetett komponens, ami az EditNote komponenssel együtt működik. A Main komponens betöltésekor és inicializálásakor betölti a bejelentkezett felhasználó jegyzeteit:

A képen szöveg, Betűtípus, képernyőkép, Grafika látható

Automatikusan generált leírás

A jegyzetek a következőképpen néznek ki:

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Egy Pressable, ami tartalmaz 2 Text-et. Az egyik a címét, míg a másik a CreationDate-t írja ki. A jegyzetre nyomáskor egy handleNotePress metódust futtat le a rányomott jegyzett azonosítójával.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Ez a metódus beállít egy csomó, feldolgozáshoz szükséges változót, többek között azt a változót is, amelyik az előnézeti modal megjelenéséért felelős. A modal tartalmazza a jegyzet címét, tartalmát (markdown-al renderelve), és a releváns funkciókat, mint például a törlést, módosítást, és a kedvencekbe rakást. Ezeket ikonok jelzik, amik változhatnak állapottól függően. A funkciókkal szemközti oldalon van egy nyíl, ami elbocsájtja a modal-t, visszadobva a jegyzetekhez. Visszatérve a funkciókhoz, a törlés gomb a következő metódust futtatja le:

A képen szöveg, képernyőkép látható

Automatikusan generált leírás

Először a konfirmációs ablakot hoz be, amit ha megerősítünk, elküld egy DELETE kérelmet az arra megalkotott végpontra, és elbocsájtja a modal-t. A metódus, ami a kedvencekbe rakást kezeli, a következőképpen néz ki:

A képen szöveg, képernyőkép, Betűtípus, szoftver látható

Automatikusan generált leírás

Ez az egyszerű metódus elküld egy POST kérelmet az arra megalkotott végpontra, ami az adatbázisban megváltoztatja a kiválasztott jegyzet azon tulajdonságát, ami azt jelöli, hogy az adott jegyzet a kedvencek között szerepel, vagy sem. Ezt követően frissíti a jegyzeteket. Végezetül, a szerkesztés gomb a következőt futtatja le:

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Ez a metódus elbocsájtja a modal-t, és egy új képernyőre küldi a felhasználót a React Navigation segítségével. Így rátérhetünk végre az EditNote komponensre, ami a fent látható kép alapján kikövetkeztethetően bekért adatok alapján dolgozik. Az EditNote komponens 2 funkcióra használatos: a jegyzetek szerkesztésére, illetve új jegyzetek létrehozására. Ezt az isNew attribútum határozza meg, ami eldönti, hogy a komponens melyik végpontra küldje el a POST, vagy PUT kérelmet. Betöltéskor elénk tárul egy teljese képernyőt befedő szerkesztői felület, ahol tudjuk módosítani, illetve létrehozni a jegyzeteket. attól függően, hogy melyik gombra nyomtunk. A szerkesztői felület alján 3 gomb található. Egy vissza gomb, egy előnézeti gomb, és egy mentés gomb. Ebből a vissza gomb és a mentés gomb magáért beszél. Az előnézeti gomb megnyomásakor pedig szintúgy, mint a bejelentkezési felületnél, átváltja a jegyzet tartalmát kijelző TextInput-ot egy Markdown renderelő komponensre, amit az isPreview változó határoz meg.

## Tesztelés

Az applikáció csak mobilon működik. Az applikációhoz legalább 1334 × 750 felbontású telefon szükséges. A program szinte minden funkciója interneten keresztül végez valamilyen kérést, vagy műveletet, azért a hibákat legtöbbször a backend-ben kezeljük. Hiba esetén a hibához tartozó http hibakóddal, és üzenettel kap választ a felhasználó egy alert-en leresztül.

## Továbbfejlesztési lehetőségek

Úgy gondolom, hogy az applikációt a következő ötletek sokkal funkcionálisabbá, és elérhetőbbé tennék:

* Jegyzetek egyéni kategóriába rendezése
* Jegyzetek egyéni színezése
* Jegyzetek megosztása más felhasználókkal
* Jegyzetek részletesebb szűrése
* Jegyzetek letöltése / lokálissá tétele
* Emlékeztetők hozzáadása
* Jegyzetek helymeghatározáshoz kötése, ami előre hozza a releváns jegyzeteket a felhasználónak, ha egy megadott hely közelében tartózkodik

# Összegzés

Úgy gondolom, hogy az applikáció hasznos lehet bárki számára. Leginkább fejlesztőknek, vagy technológiailag jártasabb embereknek készült, de javára szolgálhat egy laikus felhasználónak is. Az app képes eltárolni mindent, ami szöveg, kiegészítve képekkel, formázásokkal, táblázatokkal, ezáltal képes egy nagyobb igénykészletet kiszolgálni. Ennek köszönhetően, szélesebb körben válhat hasznára bárkinek, akinek szüksége van egy jegyzetelő applikációra.

Őszintén szólva, az applikáció frontend szegmense nem sikerült úgy ahogy terveztem, de sokat tanultam a hibáimból. Gyakran elakadtam bizonyos megvalósításoknál, voltam kilátástalan is bizonyos szituációkban, de sikeresen megléptem ezeket a nehézségeket. Így remélem, hogy az applikáció sok ember javára szolgál majd.

# Irodalomjegyzék

<https://boxicons.com/>

<https://www.w3schools.com/>

<https://developer.mozilla.org/en-US/>

<https://www.youtube.com/>

# Eredetiségnyilatkozat

Alulírott Gazdag Ádám nyilatkozom, hogy záró dolgozatom a saját szellemi termékem.

Debrecen, 2024. márc. 22

.

aláírás