Jegyzőkönyv

Mobil programozás alapjai

Todo App

Készítette: **Lőrinc Ákos**

Neptunkód: **XVG56E**

Dátum: 2025. január 29.

**Miskolc, 2025**

Tartalomjegyzék

[Bevezetés 3](#_Toc189088738)

[Használt technológiák és eszközök 3](#_Toc189088739)

[Struktúra szerkezete 3](#_Toc189088740)

[Fájlok részletezése 4](#_Toc189088741)

[App.js 4](#_Toc189088742)

[TodoList.js 4](#_Toc189088743)

[TodoModal.js 5](#_Toc189088744)

[AddListModal.js 6](#_Toc189088745)

[Felhasználói felület 7](#_Toc189088746)

[Kezdő oldal 7](#_Toc189088747)

[Új lista felvétele 7](#_Toc189088748)

[Teendők kezelése 8](#_Toc189088749)

# Bevezetés

Ez a projekt egy React Native Expo alkalmazás, amely a Firebase-t használja a felvett adatok tárolására.

## Használt technológiák és eszközök

1. **React Native és Expo:**

* A React Native biztosítja a mobil alkalmazások fejlesztésének alapjait.
* Az Expo egy fejlesztői környezet, amely egyszerűsíti a React Native alkalmazások létrehozását és futtatását.

1. **Firebase:**

* **Firestore Database**: A Firebase felhő alapú adatbázisa, amelyet a jegyzetek tárolására használunk.

## Struktúra szerkezete

* **App.js:** ez az alkalmazás fő komponense
* **Color.js:** ez a fájl tárolja az alkalmazásban használt színeket
* **Fire.js:** ez a fájl felelős az adatbázishoz való csatlakozásért
* **components/AddListModal:** ezt a komponenst a fő oldalról tudjuk megnyitni és ezáltal tudunk felvenni új listákat
* **components/TodoList:** ez a komponens felelős a listák megjelenítéséért a fő oldalon
* **components/TodoModal:** egy lista kiválasztása után ez a komponens jelenik meg, a listában található teendőket jeleníti meg és lehetőséget biztosít újak felvételére
* **package.json:** Tartalmazza a projekt függőségeit és alapvető konfigurációit

# Fájlok részletezése

## App.js

1. **Állapotkezelés (state):**

Az alkalmazás state változója az alábbi értékekkel indul:

* addTodoVisible: Az AddListModal láthatóságát irányítja.
* lists: A listák tárolása, amelyeket a Firebase adatbázisból tölt be.
* user: A felhasználói adatok tárolása.
* loading: A betöltési állapot, amely kezdetben igaz.

1. **ComponentDidMount függvény:**

A componentDidMount függvényben inicializálódik a Firebase kapcsolat, és a getLists metódus segítségével lekérjük a listákat az adatbázisból. Miután a lista betöltődött, a loading állapotot hamisra állítjuk, jelezve a felhasználónak, hogy az alkalmazás készen áll.

1. **ComponentWillUnmount függvény:**

A komponens eltávolításakor (ha a felhasználó elhagyja az oldalt) a Firebase kapcsolatot bontjuk.

1. **AddListModal** **ablak vezérlés:**

Az AddListModal ablak, amely lehetővé teszi új listák hozzáadását, láthatósága az addTodoVisible változótól függ. Az ablakot a toggleAddTodoModal metódus segítségével kapcsolhatjuk be és ki,

ami az addTodoVisible értékét változtatja.

1. **Listák kezelése:**

* **renderList**: A TodoList komponenst használja, amely megjeleníti a listákat és lehetővé teszi annak módosítását.
* **addList**: Az új lista hozzáadására szolgáló metódus, amely a Firebase adatbázisba küldi az új listát.

## TodoList.js

1. **Állapotkezelés (state):**

A state tartalmazza a showListVisible változót, amely a lista modal ablakának láthatóságát szabályozza. Kezdetben az értéke false.

**2. TodoModal ablak vezérlés:**

Azablakot a toggleListModal metódus kezeli, amely a showListVisible állapotot váltogatja. Az ablak akkor jelenik meg, ha a felhasználó rákattint a lista elemére.

**3. Feladatok állapota:**

* **completedCount**: A lista feladatain belül lévő elvégzett feladatok számát jeleníti meg.
* **remainingCount**: A lista hátralévő feladatait jeleníti meg.

## TodoModal.js

**1. Állapotkezelés (state):**

* A komponens egyetlen állapotváltozót (newTodo) használ, amely az újonnan beírt teendő szövegét tárolja.
* Az állapotot a TextInput mezőhöz kötött onChangeText metódus frissíti.

**2. Feladat állapotának módosítása (toggleTodoCompleted):**

* Az adott indexű feladat completed értéke megváltozik (true -false).
* A változást az updateList metódus továbbítja a fő komponensnek, ami továbbítja a Fire.js-nek és frissíti az adatbázist.

**3. Új feladat hozzáadása (addTodo):**

* Ha a beírt feladat címe még nem szerepel a listában, akkor hozzáadja.
* Az új elem alapértelmezett állapota: false.
* A setState({ newTodo: "" }) gondoskodik arról, hogy a beviteli mező kiürüljön.
* A Keyboard.dismiss() elrejti a billentyűzetet.

**4. Feladat törlése (deleteTodo):**

* Az adott indexű feladatot törli a listából a splice metódussal.

**5. Feladatok listázása (renderTodo):**

* Minden feladat egy View konténerbe kerül.
* A TouchableOpacity gomb segítségével lehet a teendő állapotát módosítani.
* A textDecorationLine: "line-through" áthúzza az elvégzett feladatokat.
* Egy másik TouchableOpacity elem szolgál egy teendő törlésre (kuka ikon).

## AddListModal.js

**1.Állapotkezelés (state):**

* name: Az új lista neve, amit a TextInput frissít.
* color: Az új lista színe. Alapértelmezetten a backgroundColors tömb első elemét veszi fel.

**2. Új lista létrehozása (createTodo):**

* Létrehozza az új lista objektumot { name, color }.
* Meghívja a this.props.addList(list) metódust, amely a fő alkalmazásba továbbítja az új listát, ami továbbítja a Fire.js-nek és frissíti az adatbázist.
* Bezárja a modalt (this.props.closeModal()).

**3. Színválasztó (renderColors):**

* Végigmegy a backgroundColors tömb elemein, és minden színhez létrehoz egy TouchableOpacity elemet.
* Ha a felhasználó egy színre kattint, az lesz az új lista színe.

# Felhasználói felület

## Kezdő oldal

Az alkalmázás kezdő oldala, itt látható az Add List gomb, ami megnyomására ugrik fel az AddListModal komponens. A gomb alatt pedig a létező listák jelennek meg, amik között oldalirányosan görgethetünk.

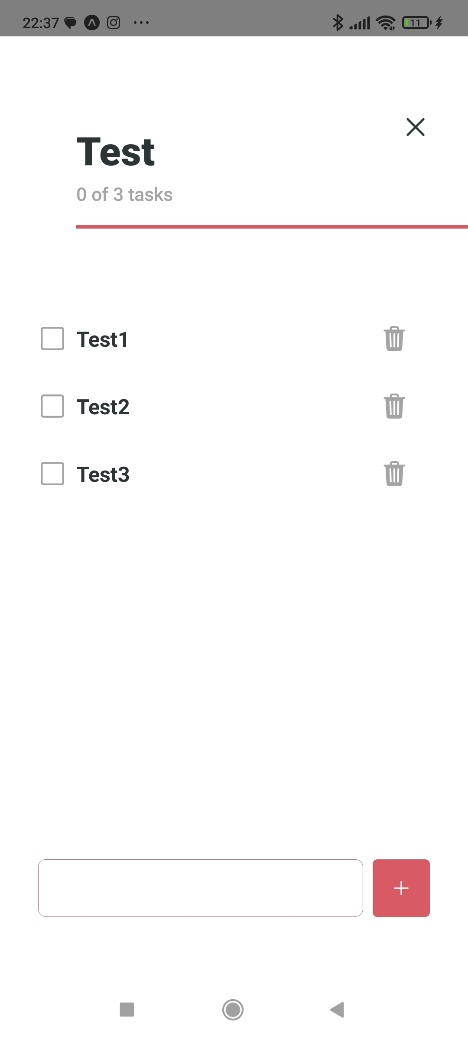
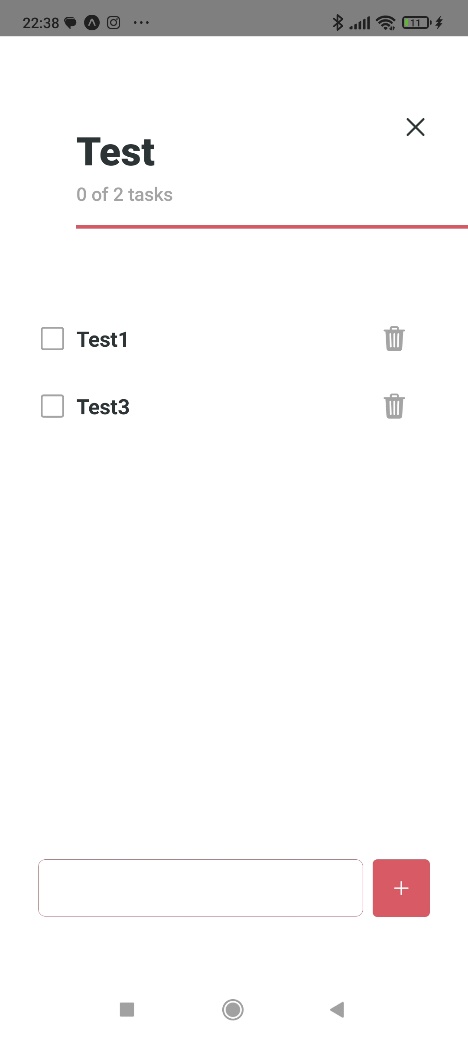
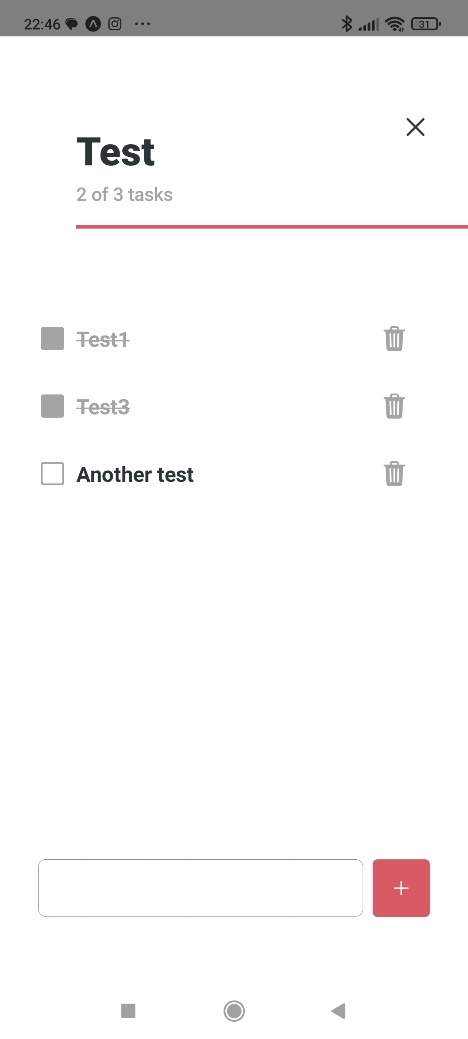
1. ábra Kezdő képernyő

## Új lista felvétele

Itt tudunk új listákat létrehozni és kiválasztani azok színét.

2. ábra Új lista hozzáadása

## Teendők kezelése

 Egy listára kattintva erre az oldalra jutunk, itt tudjuk a listában található teendőket kezelni, újat létrehozni, megjelölni elvégzettként és kitörölni őket.

5. ábra

4. ábra

3. ábra

* 3. ábra: itt három darab felvett teendőt láthatunk
* 4. ábra: itt a Test2 névvel rendelkező teendő el lett távolítva a listából
* 5. ábra: itt a Test1 és Test3 teendő meg lett jelölve készként és egy új fel lett véve