A képen szöveg, Betűtípus, képernyőkép, Grafika látható

Automatikusan generált leírás

**Saját célú chat applikációs szoftver**

Készítette:

Illés Péter (KPNJ8D)

Iványi Dávid (BOGUYX)

Vörös Gábor (X6XFNU)

**GKNB\_INTM005**

**2023/2024/1**

Tartalomjegyzék

[Felhasznált technológiák: 3](#_Toc152363428)

[React Native 3](#_Toc152363429)

[Firebase: 4](#_Toc152363430)

[Visual Studio Code (VS Code) 7](#_Toc152363431)

[Expo Go: 8](#_Toc152363432)

[GitHub: 9](#_Toc152363433)

[Adobe XD: 11](#_Toc152363434)

[Funkcionalitások: 13](#_Toc152363435)

[UI/UX Design: 19](#_Toc152363436)

[Alapvető Design Irányelvek 19](#_Toc152363437)

Ez a dokumentum részletes áttekintést nyújt az általunk fejlesztett React Native alapú chat alkalmazásról, amelyet egy cég belső kommunikációjának megkönnyítésére terveztünk és készítettünk az alkalmazottak számára.

Az alkalmazás kialakítását úgy terveztük, hogy hatékonyan támogassa egy adott cég belső kommunikációját. A cég munkatársai saját fiókot tudnak létrehozni az appon belül, ezután be is tudnak jelentkezni oda. A közös, csoportos chaten belül tudnak egymással kommunikálni, ezáltal könnyedén kezelhetik és átláthatóvá tehetik a cégen belüli információk áramlását.

# Felhasznált technológiák:

## ***React Native***

React Native egy nyílt forráskódú keretrendszer, amelyet a Facebook fejlesztett ki, és lehetővé teszi mobilalkalmazások készítését egyetlen kódbázisból. A React Native az eredeti React webes keretrendszer alapelveit használja, de lehetővé teszi a fejlesztők számára, hogy natív mobilalkalmazásokat hozzanak létre JavaScript és React ismereteik felhasználásával.

A React Native alkalmazza a "komponens alapú" fejlesztési módot, ami azt jelenti, hogy az alkalmazás egyes részeit különálló komponensek képezik, amelyek egymásba ágyazhatók és újra felhasználhatók. Az alkalmazás megjelenése a natív API-kon keresztül érhető el, és a JavaScript kód a natív környezetben fut, gyorsítva ezzel a teljesítményt.

Keresztplatformosság és újra felhasználhatóság:

Az egyik legfontosabb előnye a React Native-nek a keresztplatformosság. A fejlesztők egyetlen kódbázisból tudnak Android és iOS alkalmazásokat készíteni, minimalizálva ezzel a fejlesztési erőfeszítéseket. A komponensek többszöri felhasználhatósága tovább növeli a hatékonyságot, mivel a fejlesztők a különböző részeket könnyen újra hasznosíthatják különböző alkalmazásokban.

Fejlesztői Élmény:

A React Native fókuszál a gyors fejlesztésre és a könnyű tanulhatóságra. A Hot Reloading funkció lehetővé teszi a fejlesztők számára, hogy élőben lássák a változtatásokat anélkül, hogy újra kellene indítaniuk az alkalmazást. Ez növeli a fejlesztési ciklusok sebességét és lehetővé teszi a gyors visszajelzést.

Közösség és Támogatás:

A React Native rendelkezik egy hatalmas fejlesztői közösséggel, amely folyamatosan fejleszti és támogatja a keretrendszert. Ez gazdag dokumentációkat, harmadik féltől származó könyvtárakat és megoldásokat biztosít a különböző problémákra. A közösség által nyújtott támogatás segít a fejlesztőknek megoldani az esetlegesen felmerülő kihívásokat.

Közvetlen Hozzáférés a Perifériákhoz:

A React Native lehetővé teszi a fejlesztők számára, hogy közvetlenül hozzáférjenek a mobil eszközök perifériáihoz, mint például a kamera, a GPS vagy a helymeghatározás. Ez lehetővé teszi a fejlesztők számára, hogy teljeskörűen kihasználják a mobilplatformok adottságait.

Hátrányok és Kihívások:

Mint minden keretrendszernek, a React Native-nek is vannak hátrányai. A néhány specifikus natív funkcióhoz való korlátozott hozzáférés, az alkalmazások méretének növekedése és a natív fejlesztéssel szembeni esetleges késés mind olyan kihívások, amelyeket a fejlesztés során figyelembe kellett vennünk.

Összegzés:

A React Native erőteljes eszköz a mobilalkalmazások készítéséhez, különösen olyan projektjekez, amelyeknek fontos a keresztplatformosság és az agilis fejlesztési módszerek alkalmazása. A megfelelő projektek esetén a React Native gyorsabb és hatékonyabb fejlesztést eredményezhet.

## ***Firebase***

Firebase egy olyan teljes körű fejlesztői platform, amely számos szolgáltatást kínál a felhőben történő alkalmazásfejlesztéshez. A Google által üzemeltetett Firebase olyan eszközök és szolgáltatások összessége, amelyek lehetővé teszik a fejlesztők számára az alkalmazások gyors és hatékony fejlesztését, tesztelését és üzemeltetését.

Autentikáció:

A Firebase Autentikáció segítségével könnyedén integrálható a felhasználók hitelesítése. Támogatja az e-mail, jelszó, Google, Facebook, Twitter, GitHub és más hitelesítési módokat.

Firestore (Adatbázis):

Firestore egy NoSQL dokumentumalapú adatbázis, amely lehetővé teszi az alkalmazásoknak a valós idejű adatok tárolását és szinkronizálását a felhőben. A Firestore gyors, skálázható és könnyen használható.

Felhőfunkciók:

A Firebase Felhőfunkciók segítségével kód futtatható a felhőben anélkül, hogy saját szervereket kellene üzemeltetni. Az alkalmazásokban előforduló háttérfolyamatokat és automatizációkat könnyen implementálhatják a fejlesztők.

Firebase Realtime Database:

Az alternatívája a Firestore-nak, egy valós idejű adatbázis, amely lehetővé teszi az alkalmazások számára a valós idejű adatok szinkronizálását a kliens és a szerver között.

Felhőtárhely:

Firebase felhőtárhely biztosít fájlok tárolására, így az alkalmazások könnyen kezelhetnek és szolgáltathatnak felhasználók részére különböző típusú médiafájlokat.

Hosting:

Firebase Hosting lehetőséget biztosít a webalkalmazások könnyű és gyors közzétételét a felhőben. A statikus és dinamikus tartalmak egyszerűen kezelhető, illetve automatikus SSL-támogatást is biztosít.

A/B tesztelés:

A Firebase A/B tesztelési funkcióval könnyen lehetőség nyílik különböző változatok tesztelésére és összehasonlítására, hogy optimalizáljuk az alkalmazás teljesítményét.

Analytics:

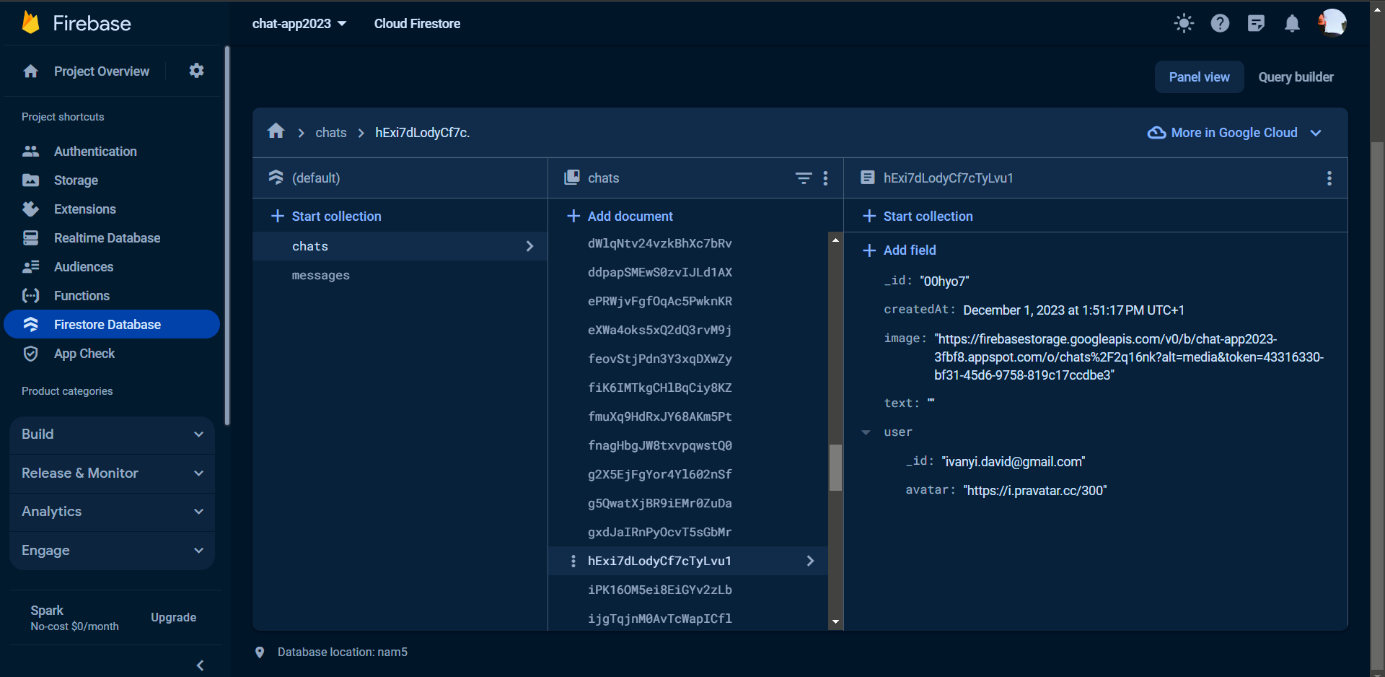
Firebase Analytics segítségével részletes adatokat kaphatunk az alkalmazásunk használatáról, beleértve a felhasználói viselkedést, a konverziókat és más fontos mutatókat.

Biztonság:

Firebase biztonság funkciók biztosítják a felhasználók adatainak védelmét, például hozzáférési ellenőrzések, SSL-titkosítás és egyéb biztonsági intézkedések révén.

Összegzés:

Firebase egy összetett és teljeskörű platform, amely számos különböző funkciót kínál az alkalmazások fejlesztéséhez és üzemeltetéséhez a felhőben. Rugalmassága és könnyű integrálhatósága miatt széles körben alkalmazzák a kis induló projektektől a nagyvállalati alkalmazásokig.



## ***Visual Studio Code***

A Visual Studio Code (VS Code) egy ingyenes, nyílt forráskódú fejlesztői környezet, amelyet a Microsoft készített, és kifejezetten a kódszerkesztéshez, hibakereséshez és az alkalmazások fejlesztéséhez lett tervezve. A VS Code könnyen testreszabható, rendkívül gyors és számos kiterjesztés érhető el, ami lehetővé teszi a fejlesztők számára, hogy az igényeikhez igazítsák.

Keresztplatformosság:

A VS Code minden népszerű operációs rendszeren elérhető, beleértve a Windowsot, a macOS-t és a Linuxot is, így lehetővé téve a fejlesztők számára, hogy ugyanazt az eszközt használják a különböző platformokon.

Nyílt Forráskód és Kiterjeszthetőség:

A VS Code nyílt forráskódú, így bárki hozzáférhet a forráskódhoz, és akár saját kiegészítőket is készíthet. Az alkalmazás kiterjeszthetősége révén a fejlesztők könnyen testreszabhatják a környezetüket a saját igényeikhez.

Intelligens Kódszerkesztés:

Az IntelliSense technológiának köszönhetően a VS Code intelligens kódszerkesztést kínál, amely segít a kódolási folyamat felgyorsításában. A kódkiegészítés, a paraméterek megjelenítése és a hibakeresési segédprogramok a fejlesztők hatékonyabbá teszik a fejlesztési folyamatban.

Beépített Git Integráció:

A VS Code beépített támogatást nyújt a Git verziókezelő rendszerhez. Ez lehetővé teszi a fejlesztők számára, hogy nyomon kövessék a változásokat, kezeljék az ágakat, és közvetlenül az IDE-ből hajtsák végre a Git műveleteket.

Terminál Integráció:

A VS Code-ban beépített terminált használhatunk, így nem szükséges külön parancssort vagy terminált nyitni. Ez lehetővé teszi a kód futtatását, a parancssori műveleteket és egyéb rendszerutasításokat közvetlenül az IDE-ből.

Debugger Támogatás:

A VS Code teljes körű hibakeresési támogatást nyújt, beleértve a beépített hibakeresőt is. Könnyen konfigurálható a különböző nyelvekhez, és lehetővé teszi a változók ellenőrzését, a töréspontok beállítását és a kódsorok hibakeresését.

Témák és Testreszabhatóság:

A VS Code rengeteg témát kínál, amelyek közül választhatunk a kinézet testreszabásához. Továbbá, a felhasználók saját beállításokat, billentyűparancsokat és kiterjesztéseket is hozzáadhatnak a saját preferenciáiknak megfelelően.

Széles Kiterjesztési Képességek:

A Visual Studio Code egy erős kiterjesztési API-val rendelkezik, így a fejlesztők számos kiegészítőt telepíthetnek, amelyek támogatják a különböző programozási nyelveket, keretrendszereket és fejlesztési eszközöket.

Felhőintegráció:

Az integrált felhőszolgáltatásokkal a VS Code lehetővé teszi az alkalmazások könnyű fejlesztését és hozzáférését a felhőben tárolt adatokhoz.

Összegzés:

A Visual Studio Code egy modern, erőteljes fejlesztői környezet, amely ideális választás a kódszerkesztéshez, hibakereséshez és alkalmazásfejlesztéshez. Az aktív fejlesztői közösség, a gyakori frissítések és a testreszabhatóság révén a VS Code népszerű eszközzé vált a fejlesztők körében.

## ***Expo Go***

Az Expo egy nyílt forráskódú platform, amely lehetővé teszi a React Native alkalmazások gyors és könnyű fejlesztését. Az Expo célja, hogy egyszerűsítse az alkalmazásfejlesztést, és biztosítson eszközöket és szolgáltatásokat a fejlesztőknek a hatékony munkavégzéshez. Az Expo támogatja a keresztplatformos fejlesztést, és számos beépített funkciót és szolgáltatást kínál, amelyek segíthetnek az alkalmazások gyorsabb elkészítésében.

Telepítés és Tesztelés:

Az Expo Go alkalmazás telepítése és használata rendkívül egyszerű. Mindössze annyit kell tenni, hogy a fejlesztők a mobil eszközök alkalmazásboltjából letöltik az Expo Go-t, majd a QR-kód beolvasásával könnyedén csatlakozhatnak az Expo-ra.

Keresztplatformos Tesztelés:

Az Expo Go keresztplatformos alkalmazás, így alkalmas mind az Android, mind az iOS eszközökön való tesztelésre. Ez jelentősen megkönnyíti a fejlesztők számára az alkalmazások keresztplatformos kompatibilitásának ellenőrzését.

QR-kód Alapú Kapcsolódás:

A fejlesztők egy QR-kódot generálhatnak az Expo fejlesztői környezetben, amelyet az Expo Go alkalmazás beolvasásával gyorsan és egyszerűen csatlakozhatnak az alkalmazásukhoz, a valós idejű tesztelés céljából.

Eszközbeállítások Szimulálása:

Az Expo Go lehetőséget biztosít arra, hogy a tesztelők szimulálják az eszközbeállításokat, például a különböző kijelzőméreteket és operációs rendszereket, így az alkalmazások viselkedése változatos környezetekben is ellenőrizhető.

Expo Go és Tesztelési Folyamat:

Miután a fejlesztők felkészítették az Expo-ra épülő alkalmazásukat, az Expo Go használatával könnyedén és gyorsan tesztelhetik azt valós környezetben. A QR-kód alapú csatlakozásnak köszönhetően a tesztelők azonnal láthatják az alkalmazás változásait, ami rendkívül hatékony és időtakarékos.

Összegzés:

Az Expo Go széles körű funkcionalitása és egyszerű kezelhetősége miatt ideális eszköz a fejlesztőknek, akik könnyen és gyorsan szeretnék ellenőrizni az alkalmazásaikat a különböző mobilplatformokon.

## ***GitHub:***

GitHub egy webes alapú verziókezelő platform, amely lehetővé teszi a fejlesztők számára, hogy együttműködjenek a kód projektek fejlesztésében. A GitHubot elsősorban a kódmegosztás, a változások kezelése, a kollaboráció és a projektmenedzsment célokra használják. A fejlesztők itt tárolják és nyomon követik a forráskód változásait, illetve támogatja a verziókezelési rendszereket, például a Git-et.

Git Verziókezelés:

A GitHub a Git verziókezelő rendszert használja, amely lehetővé teszi a forráskód változásainak nyomon követését, a különböző verziók közötti navigálást és az esetleges konfliktusok feloldását.

Repository (Tároló):

A GitHubon a projektjeinket tárolókba, vagy más néven repositorykba szervezhetjük. Ezek a tárolók tartalmazzák a projekt összes fájlját, illetve a projekthez tartozó történeti változásokat is.

Ágak (Branches):

Az ágak lehetővé teszik a párhuzamos fejlesztést. Egy ág egy független fejlesztési útvonalat jelent, amelyet a fejlesztők létrehozhatnak és egyesíthetnek a főággal, vagy más ágakkal.

Pull Request és Forgalomkezelés:

A Pull Requestek segítségével a fejlesztők változtatásokat javasolhatnak a kódban. Ezen keresztül történik a kollaboráció és a változtatások elfogadása vagy elutasítása.

Issue Tracker (Probléma Követő):

Az Issue Tracker egy hely, ahol a felhasználók és a fejlesztők problémákat, hibákat, ötleteket és egyéb problémákat jelezhetnek. Ezek a problémák nyomon követhetők és kioszthatók a fejlesztői csapat tagjai között.

GitHub Actions:

GitHub Actions segítségével automatizálhatók a fejlesztési folyamatok, tesztelés, és egyéb CI/CD (folyamatos integráció/folyamatos szállítás) műveletek.

Collaborators és Témakörök:

Collaborators segítségével többen is hozzáférhetnek és módosíthatják egy projekt forráskódját. A témakörök pedig segítik a projekt kategorizálását és könnyebb megtalálását.

GitHub Gist:

A Gist egy egyszerű megosztó szolgáltatás, ahol rövid kódfragmenteket, jegyzeteket vagy más anyagokat lehet megosztani.

GitHub Desktop:

A GitHub Desktop egy asztali alkalmazás, amely egyszerűsíti a Git és GitHub használatát, különösen azoknak, akik kezdőként ismerkednek a verziókezeléssel.

Kollaboráció és Közösség:

A GitHub lehetővé teszi a fejlesztők és fejlesztői csapatok számára a könnyű együttműködést, kódmegosztást és közös projektek fejlesztését.

Nyílt Forráskód:

A GitHub jelentős részét teszi ki a nyílt forráskódú projekt közösségnek. Milliók által létrehozott projekt és kód érhető el a felhasználók számára.

Összegzés:

A GitHub a fejlesztői közösség egyik legfontosabb és leggyakrabban használt platformja, és egyben kulcsfontosságú szerepet játszik a kód megosztásában, együttműködésben és fejlesztési projektek menedzselésében. A platform rendkívül széles körben alkalmazható, és naprakész eszközöket és szolgáltatásokat kínál a modern fejlesztési folyamatok támogatásához.

## ***Adobe XD***

Az Adobe XD egy kiváló eszköz a digitális terméktervezéshez, amelyet az Adobe fejlesztett ki. Ez az alkalmazás lehetővé teszi a felhasználók számára, hogy gyorsan és hatékonyan tervezzenek web- és mobilalkalmazásokat, interaktív prototípusokat készítsenek, és a felhasználói élményt optimalizálják. A Chat App tervezése során az Adobe XD különleges jelentőséggel bírt a felhasználói felületek intuitív kialakításában és az interakciók vizualizálásában.

Gyors Prototípus Készítés:

Az Adobe XD lehetővé teszi a gyors prototípusok készítését, amelyek segítik a tervezőket és fejlesztőket a tervezett alkalmazás funkcionalitásának és áramlatainak vizualizálásában.

Interaktív Elemek és Animációk:

A program kiválóan támogatja az interaktív elemeket és animációkat, így a tervezők valós időben láthatják, hogyan reagálnak a felhasználók a tervezett alkalmazásukra.

Kollaboráció és Megosztás:

Az Adobe XD lehetőséget kínál a csapatoknak a kollaborációra és a tervezési projektek megosztására. Az online tervezési felület lehetővé teszi a csapattagoknak a könnyű együttműködést és visszajelzést.

Komponensek és Stíluslapok:

A tervezők könnyen kezelhetik és frissíthetik az alkalmazás különböző részeit azáltal, hogy komponenseket és stíluslapokat hoznak létre, ami egyszerűsíti a konzisztens és hatékony tervezést.

Összegzés:

Az Adobe XD egyszerűségével és sokoldalúságával egy olyan tervezési folyamatot biztosított, amely lehetővé tette számunkra, hogy gyors és átlátható terveket készítsünk, és optimalizáljuk az alkalmazás felhasználói élményét. Az alkalmazás további funkciói, például a kollaboráció és a tervezések megosztása, segítettek a csapatnak együttműködni és hatékonyan előrehaladni a tervezési folyamatban.

# Funkcionalitások:

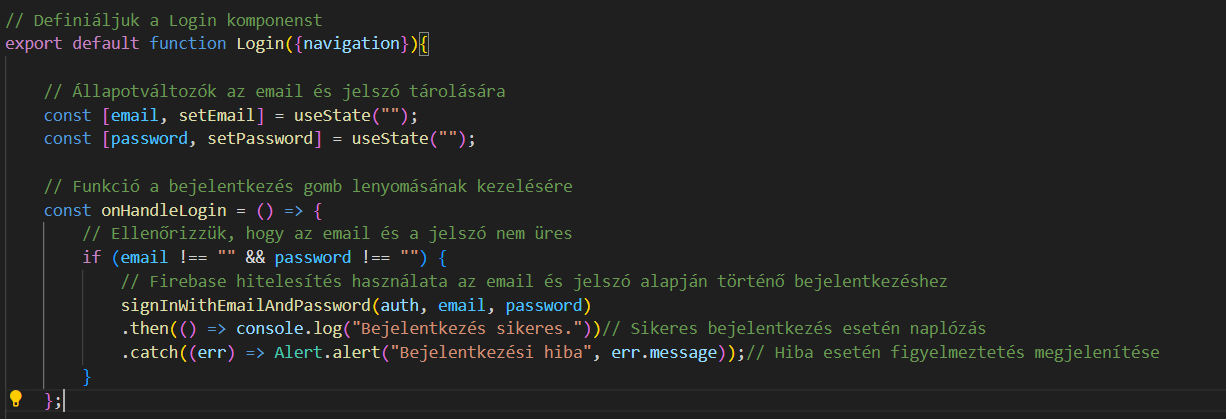
Az applikáció komponensei feladatuk szerint:

Applikáció létrehozása:

Az "Applikáció létrehozása" komponens a projekt kezdeti fázisában található, amikor a fejlesztő elkezdi létrehozni az alkalmazást. Ebben a szakaszban a technológiai választások, például a fejlesztői keretrendszer kiválasztása, és az alkalmazás általános architektúrájának megtervezése történik.

Belépési felület létrehozása:

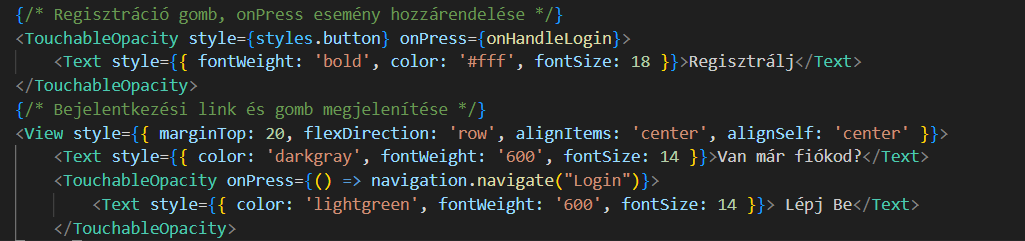
A "Belépési felület létrehozása" komponens az alkalmazás indulásakor az elsődleges interakciós pontot képezi. Itt a felhasználóknak lehetőségük van belépni az alkalmazásba, és ez a rész tartalmazza azokat az elemeket, mint például bejelentkezési űrlapok, adatazonosító mezők és hitelesítő gombok.



Regisztrációs felület létrehozása:

A "Regisztrációs felület létrehozása" komponens a felhasználók számára lehetővé teszi, hogy új fiókot hozzanak létre az alkalmazásban. Ez általában olyan elemeket tartalmaz, mint a regisztrációs űrlapok, a felhasználónév és jelszó beviteléhez szükséges mezők, valamint a regisztrációs gomb.

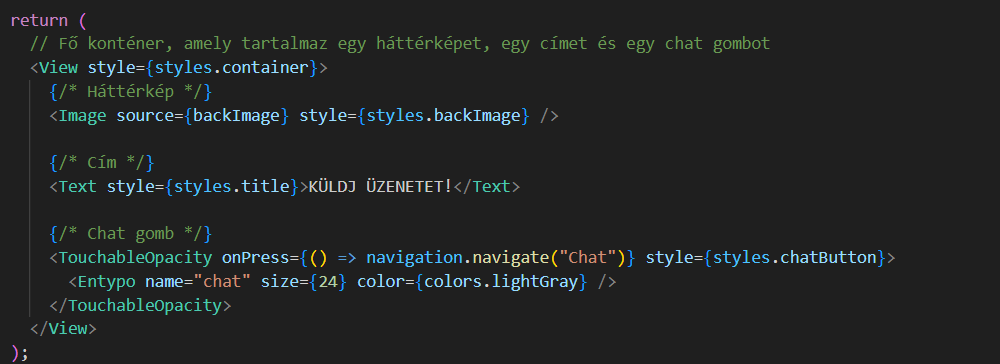




Főoldal megalkotása:

A "Főoldal megalkotása" komponens a felhasználók alapvető kezdőpontja az alkalmazásban. Ide tartozik az alkalmazás főmenüje, navigációs elemek, és azok a tartalmi blokkok, amelyek segítik a felhasználókat a különböző funkciók és szolgáltatások közötti könnyű és intuitív navigációban.



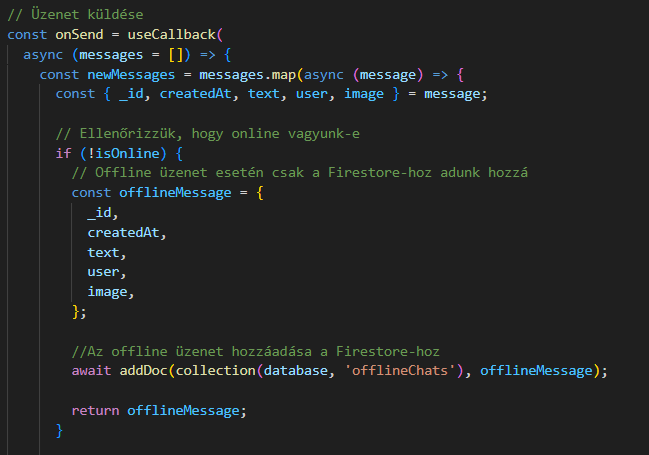


Chat oldal megalkotása:

A "Chat oldal megalkotása" komponens magában foglalja a teljes chat-funkcionalitást, beleértve az egyéni chat-szobák kezelését, a beszélgetési történet megjelenítését, az online/offline státuszokat, az értesítéseket és más olyan elemeket, amelyek lehetővé teszik a résztvevők számára egy teljes körű csevegési élményt.







A képen szöveg, képernyőkép, szoftver, Betűtípus látható

Automatikusan generált leírás

Ezek a komponensek együttműködnek az alkalmazás különböző részeiben, és segítik a felhasználókat a navigációban, a belépésben, a regisztrációban és az interakciókban az alkalmazásban, így egy egységes és funkcionális felhasználói élményt kínálnak.

# UI/UX Design:

Az alkalmazás célja egy könnyen használható és attraktív felhasználói élmény nyújtása, miközben a legfontosabb chat funkciókat támogatja. Az alkalmazásnak mind iOS és Android platformokon egyaránt jól működő dizájnra kell törekednie.

A képen szöveg, képernyőkép, tervezés látható

Automatikusan generált leírás A képen szöveg, képernyőkép, tervezés, sablon látható

Automatikusan generált leírás

## ***Alapvető Design Irányelvek***

Minimalizmus és Tisztaság:

* Egyszerű, letisztult dizájn az egyszerű használatért.
* Kevesebb felesleges információ, a lényegre összpontosítva.

Színválasztás:

* Világos színek a felhasználóbarát és barátságos atmoszféráért.
* Kontrasztos színek a fontos elemek kiemelésére.

Ikonok és Gombok:

* Áttekinthető és egyértelmű ikonok a könnyű navigáció érdekében.
* Nagy, érintésre optimalizált gombok.

A képen szöveg, képernyőkép, zöld látható

Automatikusan generált leírás 