**Chatex**

**Dokumentáció**

[***Noszlopy Gáspár Közgazdasági Technikum***](http://www.ngkszki.hu/)

**Készítette: Szép Dániel,**

**Kiss Levente,**

**(Szabó Richárd)**

# Tartalomjegyzék

Tartalomjegyzék

[Tartalomjegyzék 2](#_Toc194485482)

[1. Bevezetés 4](#_Toc194485483)

[1.1 A témaválasztás megindoklása 4](#_Toc194485484)

[1.2 Célkitűzés 4](#_Toc194485485)

[1.2.1 Főbb jellemzők: 4](#_Toc194485486)

[1.3 Fejlesztői környezet 5](#_Toc194485487)

[1.3.1 Fejlesztői eszközök 5](#_Toc194485488)

[1.4 Elvárások a feladattal kapcsoltban 6](#_Toc194485489)

[1.4.1 Operációs rendszer, környezet 6](#_Toc194485490)

[1.4.2 Felhasználandó programozási nyelv 6](#_Toc194485491)

[1.5 Github környezet 7](#_Toc194485492)

[1.6 A kialakított adatszerkezet részletes bemutatása 8](#_Toc194485493)

[1.6.1 Az adatbázis táblái 8](#_Toc194485494)

[1.6.2 A Users tábla 8](#_Toc194485495)

[1.6.3 A messages tábla 10](#_Toc194485496)

[1.6.4 A friends tábla 11](#_Toc194485497)

[1.6.5 A friend requests tábla 12](#_Toc194485498)

[1.6.6 Felhasználó regisztrálása 13](#_Toc194485499)

[1.6.7 Felhasználó bejelentkezés 16](#_Toc194485500)

[1.6.8 Jelszó helyreállítás 18](#_Toc194485501)

[1.6.9 Ehhez tartozik még a FORM: 20](#_Toc194485502)

[1.7 Tesztelési Dokumentáció 26](#_Toc194485503)

[1.7.1 Regisztráció tesztek forgatókönyve: 26](#_Toc194485504)

[1.7.2 Tesztek pontosabb ismertetése: 26](#_Toc194485505)

[1.7.3 Bejelentkezési teszt: 28](#_Toc194485506)

[1.7.4 Felhasználó keresés teszt: 28](#_Toc194485507)

[2. Felhasználói dokumentáció 29](#_Toc194485508)

[2.1 Applikáció használatának részletes ismertetése 33](#_Toc194485509)

[2.1.1 Elfelejtett folyamatának ismertetése 34](#_Toc194485510)

[2.1.2 Barát keresése folyamat ismertetése 35](#_Toc194485511)

[2.1.3 Barátjelölések elfogadása 36](#_Toc194485512)

[2.1.4 Beállítások navigálása 37](#_Toc194485513)

[2.1.5 Fiók módosítása 38](#_Toc194485514)

[2.1 További, még nem implementált ötleteink 39](#_Toc194485515)

[3 Összefoglalás 40](#_Toc194485516)

[3.1 Munkamegosztás 40](#_Toc194485517)

[3.2 Főbb feladatok: 40](#_Toc194485518)

[ **Kiss Levente:** 40](#_Toc194485519)

[ **Szép Dániel:** 40](#_Toc194485520)

[ Szabó Richárd: 41](#_Toc194485521)

[4 Köszönetnyilvánítás 42](#_Toc194485522)

# 1. Bevezetés

## 1.1 A témaválasztás megindoklása

Alkalmazásunk a Messenger jelenlegi (v485.2.0.68.111) hibáit javítja ki, köztük a végpontok közötti titkosítástól való teljesítmény problémákat (a felhasználó dönthet a bekapcsolásáról) és a kezelő felületek inkonzisztens megjelenését egy egységes felülettel.

Ez az alkalmazás azoknak a felhasználóknak lesznek, akik ugyanúgy nem kedvelik a Messenger hibáit és felesleges hozzáadásait.

## 1.2 Célkitűzés

A Chatex alkalmazás célja, hogy egy alternatívát nyújtson a híres Messenger helyett, mégpedig úgy, hogy csak az egymás közötti csevegésre fókuszál minden olyan funkció nélkül, ami nem ezt a célt szolgálja. Más szóval, a Chatex használata egy sokkal könnyebb, gyorsabb, és felhasználó barátibb környezetet nyújt, míg ugyanúgy megtartja a játékos funkciókat. **A fejlesztés közben be kell tartani a Clean Code alapszabályait.**

### 1.2.1 Főbb jellemzők:

1. Villámgyors üzenetküldés késleltetés nélkül
2. Biztonságos adatkezelés és titkosítás a felhasználói adatok védelmében
3. Modern, letisztult felület, amely könnyen kezelhető
4. Innovatív funkciók, amelyekkel az élmény még jobb, mint a hagyományos alkalmazásokban

## 1.3 Fejlesztői környezet

A chat applikációnk elkészítéséhez többféle fejlesztőeszközöket használtunk. A kód nagy részét Visual Studio Code-ban írtuk meg.  
Az XAMPP program használatával hajtottuk végre a MySQL adatbáziskezelést és a PHP-t, főleg, hogy lehetővé teszi, hogy lokálisan

fejlesszük és teszteljük applikációnkat. Az adatbázisunk karbantartására a phpMyAdmin segítségét vettük igénybe, utána a MySQL adatbáziskezelő rendszer tárolja és kezeli a felhasználókhoz és a chatekhez tartozó adatokat

### 1.3.1 Fejlesztői eszközök

Androidra való fejlesztéshez szükség van egy IDE-re, mi esetünkben az Visual Studio Code-ra, mellyel nem kapnánk végprogramot és hogy ezt láthassuk egy Emulatorra (lehet fizikai eszköz is, illetve virtuális készülék) van szükség.

Azért választottuk a Visual Studio Code-ot mert sokkal színesebb és átláthatóbb, mint az Android Studio.  
Emellett nagyobb támogatást nyújt az Extension-öknek és ezáltal könnyebb a csoportos munkavégzés (Pl.: Live Share extension).

A Flutter Framework használata úgy indokolható, hogy lehetővé teszi a az egy programnak a több platformon lévő futtatását, illetve remekül kibővíti a Dart nyelvet a Flutter különböző Widgetekkel és hosszútávú Google támogatással.

|  |  |  |  |
| --- | --- | --- | --- |
| Platform | Minimum OS | Fejlesztési eszközök | Támogatott Architektúrák |
| Android | Android 5.0+ | Visual Studio Code | arm32, arm64, x86\_64 |

## 1.4 Elvárások a feladattal kapcsoltban

### 1.4.1 Operációs rendszer, környezet

* Android

### 1.4.2 Felhasználandó programozási nyelv

* Flutter, Dart, PHP(Backend)

## 1.5 Github környezet

A vizsgaremekhez kiírt feltételekhez és munkánk megkönnyítéséhez alakítottunk ki egy Github környezetet applikációnkhoz.

egy közös email cím létrehozásával végeztük el a környezet elkészítését.

A közös emaillel csináltunk egy repository-t Messengeres-vizsgaremek néven és ezután hozzáadtuk egymás fiókjait, hogy mindegyikőnk fejlesztése könnyen nyomkövethető és zökkenőmentes legyen.

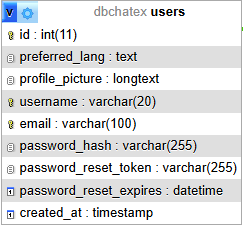
Az elkészült vizsgaremek tartalmazza az adatbázist, programkódot, dokumentációt és az előadást, amit a következő Github repositoryban érhető el: [Chatex vizsgaremek](https://github.com/smolelevent/Messengeres-vizsgaremek). itt található az összes szükséges anyag a projekt teljes megértéséhez.

## 1.6 A kialakított adatszerkezet részletes bemutatása

Chat applikációnk táblái strukturáltak, szervezettek és hatékonyan kezelik, tárolják az adatokat.

### 1.6.1 Az adatbázis táblái

### 1.6.2 A Users tábla

id (fő azonosító): Ez a mező az adott felhasználó azonosítója. Ez az elsődleges kulcs, szóval egyedi minden egyes felhasználónak ezért segít megkülönböztetni különböző felhasználókat a táblában. Ehhez az azonosítóhoz kapcsolódik az összes többi azonosító, mivel ez a fő azonosító.

preferred\_lang (kiválasztott nyelv): Ez a mező a felhasználó által kiválasztott nyelv adatait tárolja, Egyelőre még csak magyar és angol nyelv elérhető.

profile\_picture (profilkép): Ez a mező az adott felhasználó profilképe, a felhasználó állítja be egy JPEG, PNG, GIF, Animated GIF, WebP, Animated WebP, BMP, and WBMP formátumu képpel, ha nem állít be profilképet, az applikáció az alapértelmezettet adja a felhasználónak.

username (felhasználónév): Ez a mező tartalmazza az adott felhasználó által megadott felhasználónevet, ezzel a névvel tudnak a többi felhasználók rákeresni erre a személyre, ez a név jelenik meg a chatekben.

email (email cím): Ez a mező tartalmazza az adott felhasználó által megadott email címet, erre az email címre fogja megkapni a felhasználó az összes üzenetet a rendszertől (pl.: új jelszó kérés).

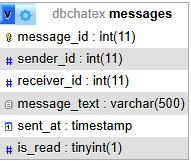
password\_hash (jelszó): Ez a mező tartalmazza az adott felhasználó által megadott jelszót, ezzel tud csak a felhasználó bejelentkezni fiókjába, és he elfelejti, új jelszót kell kérnie mert a Chatex nem fogja tudni megadni, mert az adatbázisban csak egy hash code-ként van elmentve.

password\_reset\_token (jelszó helyreállítása): Ez a mező tartalmazza az adott felhasználónak éppen függőbe lévő jelszó helyreállítási kérését, ha nincs, akkor üres, ha van, akkor az adatbázis mutatja a tokent.

password\_reset\_expires (jelszó helyreállításának élettartama): Ez a mező tartalmazza az adott felhasználónak a jelszó helyreállításának élettartamát (15 perc), ha a felhasználó nem végezte el a dolgát ebben az időtartamban, a jelszó helyreállításának tokenje automatikusan törölve lesz az adatbázisból és új kérést kell nyitnia a felhasználónak.

created\_at (fiók készült ekkor): Ez a mező tartalmazza azt, hogy adott felhasználó pontosan mikor regisztrálta fiókját.

### 1.6.3 A messages tábla

message id (üzenet azonosító): Ez a mező tartalmazza minden egyes üzenetnek az azonosítóját, hogy miután a felhasználó kilép az applikációból, ne tűnjenek el az üzenet mikor visszalép, ez a táblának az elsődleges kulcsa, a mező INT típusú és maximum 11 karaktert fogad be.

sender\_id (adó azonosító): Ez a mező tartalmazza az üzenetet küldő felhasználónak az azonosítóját, ezzel biztosítjuk, hogy az üzenet biztos ez a kettő személy között lesz elmentve, a mező INT típusú és maximum 11 karaktert fogad be.

receiver\_id (vevő azonosító): Ez a mező tartalmazza az üzenetet vevő felhasználónak az azonosítóját, ezzel biztosítjuk, hogy az üzenet biztos ez a kettő személy között lesz elmentve, a mező INT típusú és maximum 11 karaktert fogad be.

message\_text (üzenet): Ez a mező tartalmazza az üzenetek tartalmát, amihez hozzá van adva az üzenet azonosító, ezzel a kettővel nem fog elveszni se maga az üzenet, se az üzenet tartalma, a mező VARCHAR típusú és maximum 500 karaktert fogad be.

sent\_at (elküldve ekkor): Ez a mező tartalmazza azt az időpontot, amikor az adó felhasználó által elküldött üzenetet a vevő felhasználó megkapta, a mező TIMESTAMP típusú.

is\_read (láttam): Ez a mező tartalmazza azt, hogy az elküldött üzenetet a vevő felhasználó látta-e vagy nem, a mező TINYINIT típusú és csak egyetlen egy karaktert fogad be.

### 1.6.4 A friends tábla

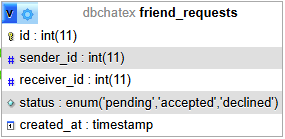
id (azonosító): Ez a mező tartalmazza a barátlistán lévő felhasználók azonosítóját, ez az azonosító az elsődleges kulcs és INT típusú, maximum 11 karaktert fogad be.

user\_id (felhasználó azonosító): Ez a mező tartalmazza a felhasználónak az azonosítóját, ez az egyik rész a barát azonosító megalkotásához, a mező INT típusú és maximum 11 karaktert fogad be.

friend\_id(barát azonosító): Ez a mező tartalmazza a két felhasználó azonosítóját, amit kombinálva megalkotja a barát azonosítót (pl.:1-27), a mező INT típusú és maximum 11 karaktert fogad be

created\_at (barátlistához hozzáadva ekkor): Ez a mező tartalmazza azt az időpontot, amikor a két felhasználó hozzá adta egymást a barátlistájukhoz, a mező TIMESTAMP típusú.

### 1.6.5 A friend requests tábla

id (azonosító): Ez a mező tartalmazza maga a barátkérelemnek az azonosítóját, hogy az adatbázis feljegyezze a két felhasználó között lehetséges több barátkérelmet is, a mező INT típusú és maximum 11 karaktert fogad be.

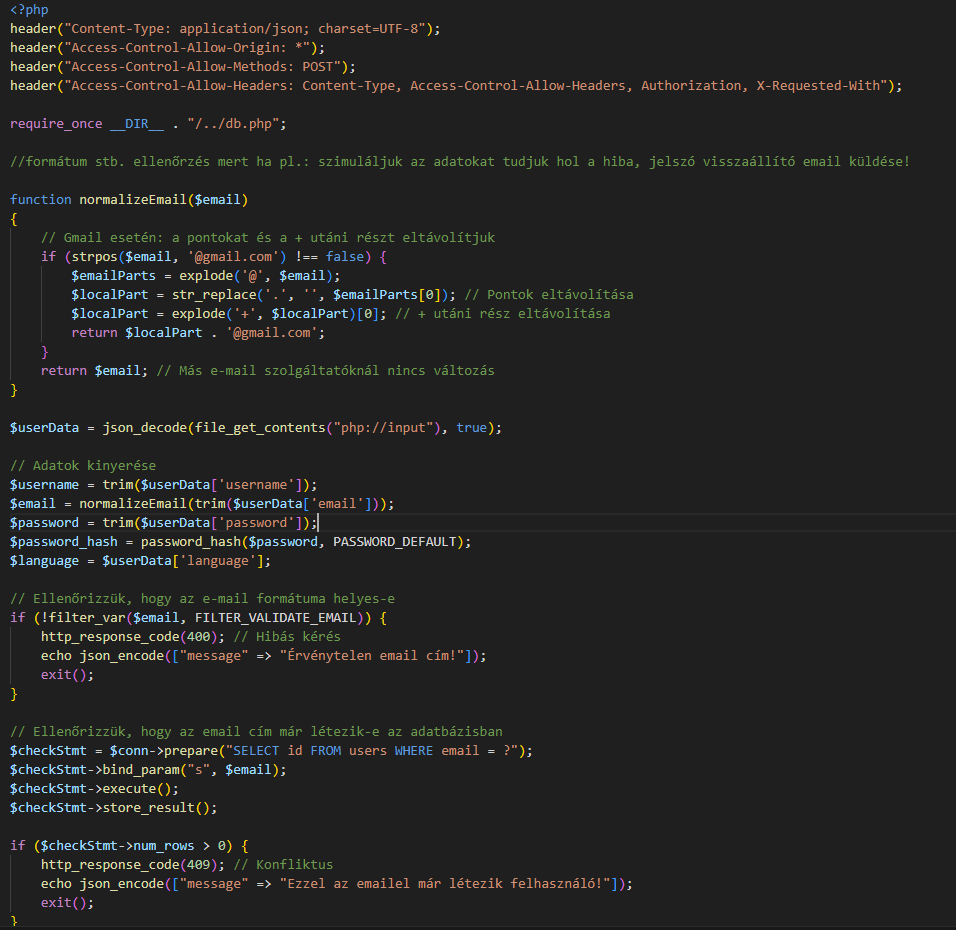
sender\_id (barátkérelmet küldő azonosító): Ez a mező tartalmazza a barátkérelmet elküldő felhasználónak az azonosítóját, a mező INT típusú és maximum 11 karaktert fogad be.

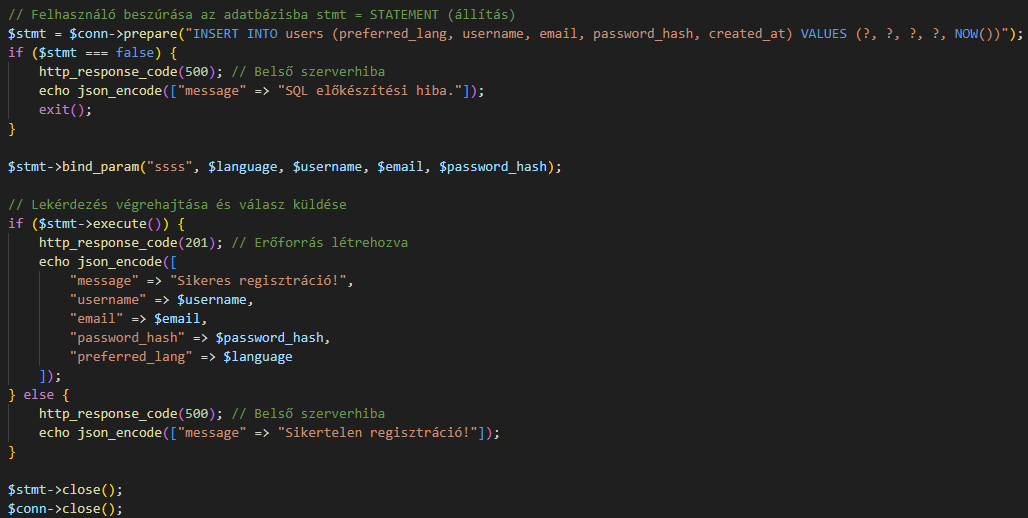
receiver\_id (barátkérelmet kapó azonosító): Ez a mező tartalmazza annak a felhasználónak az azonosítóját, aki a barátkérelmet kapta egy másik felhasználótól, aki még nincs a barátlistáján, a mező INT típusú és maximum 11 karaktert fogad be.

status (barátkérelem állapota): Ez a mező tartalmazza a barátkérelem állapotát, egyből az elküldés után a „pending” (függőben) attribútumot kapja, ha a vevő felhasználó elutasítja akkor a „declined” (elutasítva) attribútumot kapja, ha elfogadja akkor meg „accepted” (elfogadva) attribútumot kapja, a mező ENUM típusú, és csak „pending”, „accepted” és „decline” -al térhet vissza.

created\_at (barátkérelem elküldésének időpontja): Ez a mező tartalmazza azt az időpontot, amikor a barátkérelmet elküldték a felhasználónak.

### 1.6.6 Felhasználó regisztrálása

  
Ez a PHP szkript egy API végpontot biztosít a felhasználók regisztrációjához. Az API JSON formátumban fogadja a bemenetet, ellenőrzi az adatokat, normalizálja a Gmail-es e-mail címeket, és elmenti az adatokat egy MySQL adatbázisba.

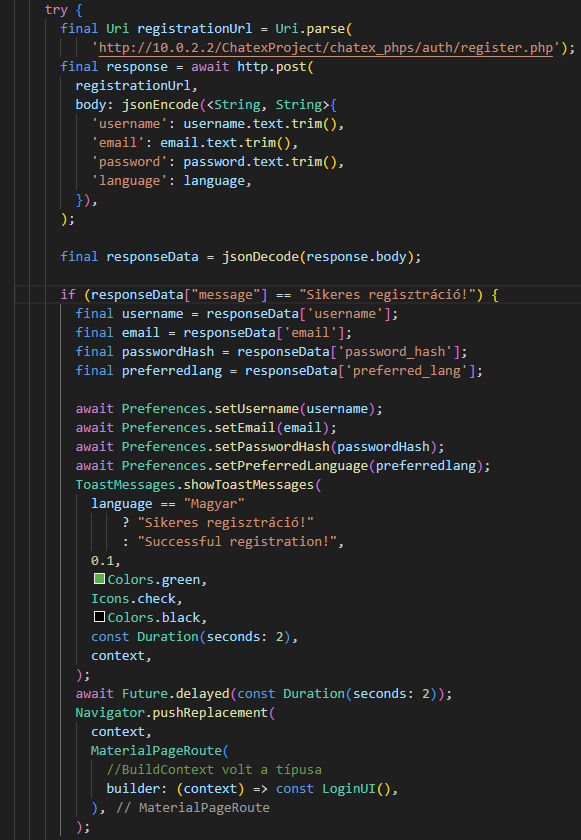
A header beállításoknál előszőr beállítjuk, hogy A válasz JSON formátumban érkezzen, bárhonnan engedélyezett a hozzáférés, Az API kizárólag **POST** metódussal hívható és az API-hoz szükséges fejléceket engedélyezi.

A normalizeEmail funkció használatával a különböző használatú emaileket egy formátummá teszi, így nem kell tökölni az összes lehetséges emailt lecsekkolásával.

Átveszi az adatokat és be is olvassa a tárolójukba:

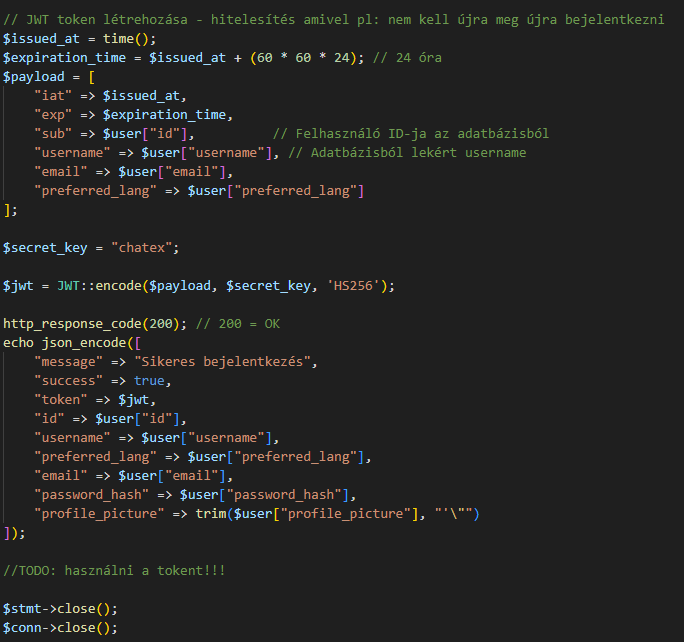
Utána ellenőrzi az email formátumát, és ha helyes, akkor tovább megy és ellenőrzi, hogy létezik-e már az email az adatbázisban, utána beszúrja az adatbázisba, ha minden jól megy és kész a regisztráció.

**Ezt a kódot az Auth.dart-ban így használjuk fel:**

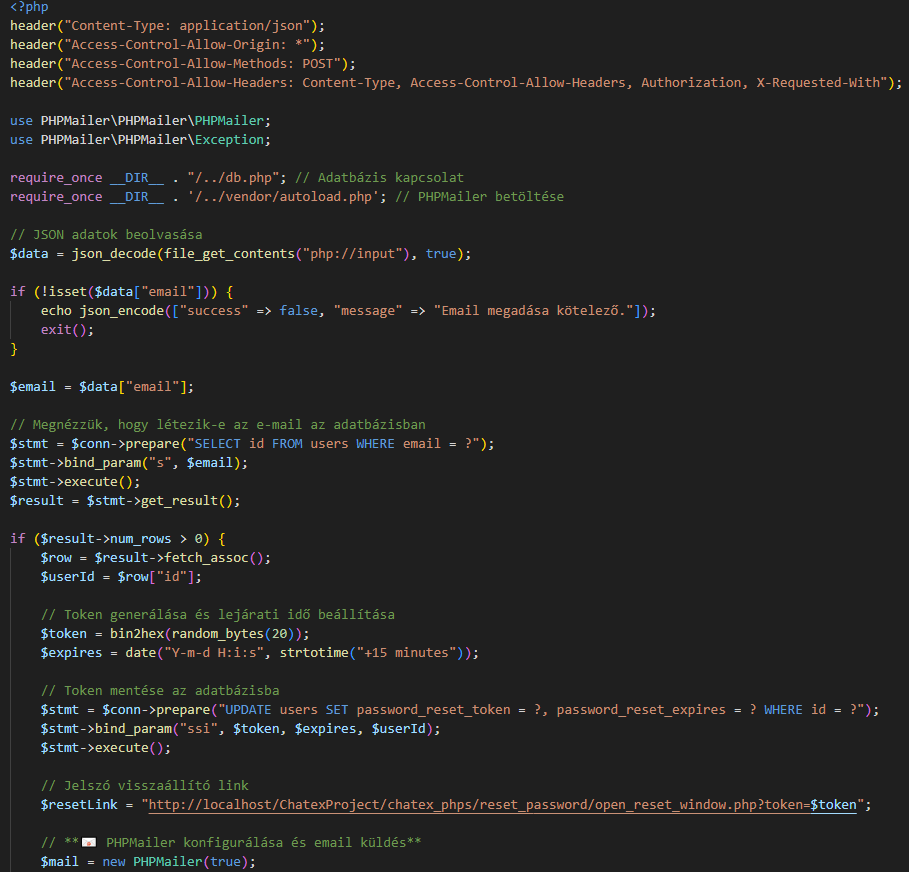


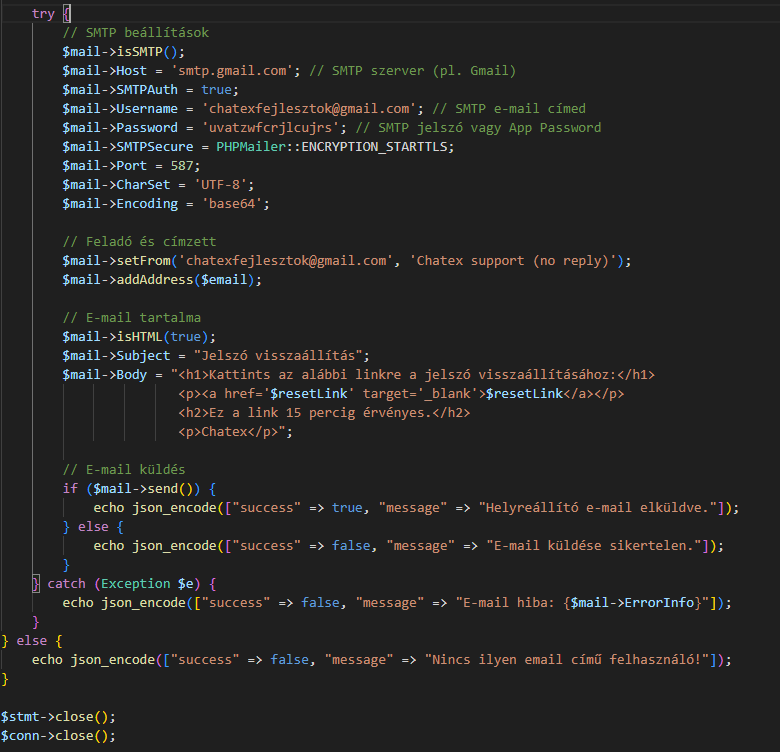
### 1.6.7 Felhasználó bejelentkezés

Ez a PHP szkript egy bejelentkezési API végpontot valósít meg JSON Web Token (JWT) használatával. A felhasználó e-mail címe és jelszava alapján hitelesíti a bejelentkezést, és sikeres azonosítás esetén egy JWT tokent generál.

Adatok fogadása: A bejövő JSON kérés tartalmát a **php://input** segítségével olvassuk be és dekódoljuk.  
Adatok ellenőrzése: Az e-mail cím és a jelszó tisztítása (trim()) után az adatbázisban ellenőrizzük a felhasználót.  
Felhasználó keresése: SQL lekérdezéssel próbáljuk megkeresni az e-mail címhez tartozó felhasználót.  
Jelszó ellenőrzése: Ha a felhasználó létezik, a password\_verify() függvénnyel ellenőrizzük a jelszót.  
JWT token generálása: Sikeres bejelentkezés esetén egy 24 órán keresztül érvényes JWT tokent hozunk létre.  
Válasz küldése: A válasz tartalmazza a JWT tokent és a felhasználó adatait.  
Hibakezelés: Hibás bejelentkezési adatok esetén 401-es státuszkódot küldünk vissza.

### 1.6.8 Jelszó helyreállítás

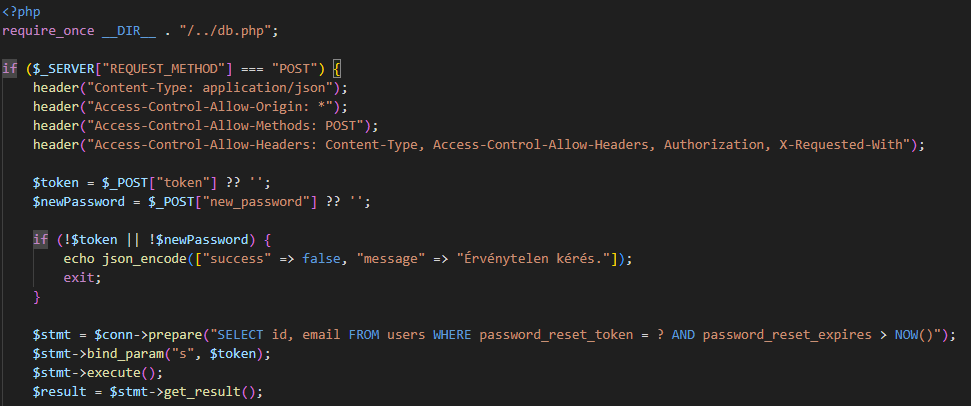


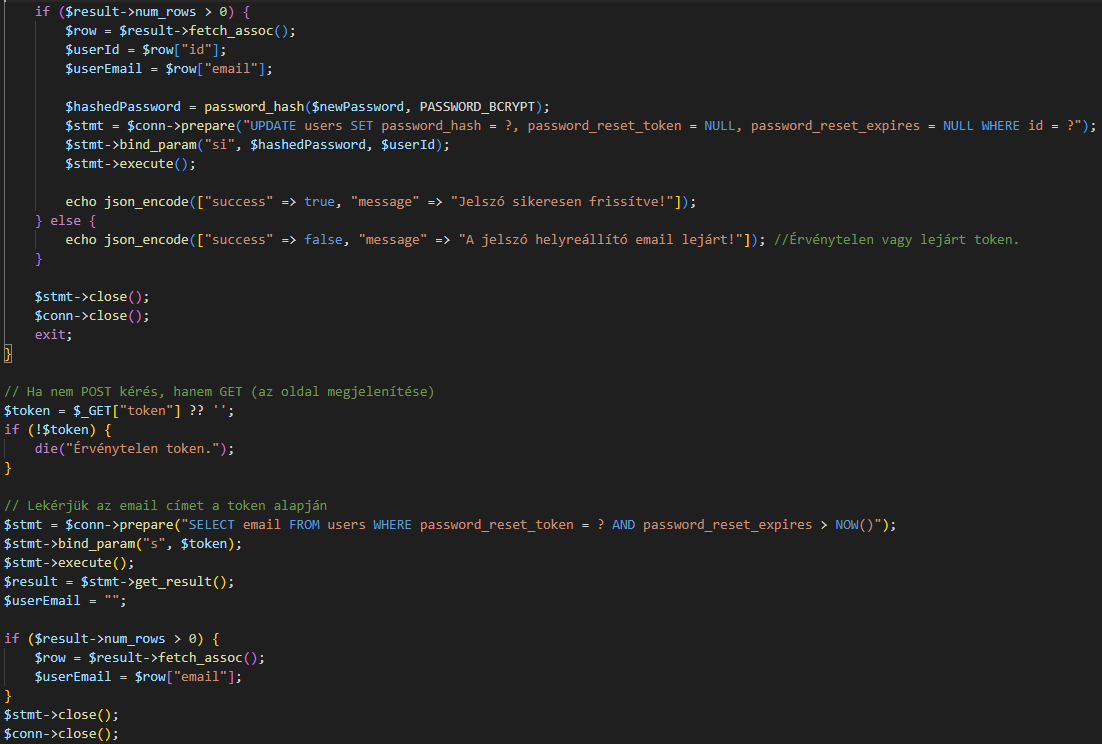


Ez egy PHP alapú rendszer, amely segít a felhasználóknak visszaállítani az elfelejtett jelszavukat. Ha valaki elfelejti a jelszavát, megadhatja az e-mail címét, és kap egy linket, amelyen keresztül új jelszót állíthat be.

A szkript fogad egy HTTP POST kérést, amely tartalmazza a felhasználó e-mail címét.  
Ellenőrzi, hogy az e-mail szerepel-e az adatbázisban.  
Ha igen, akkor generál egy véletlenszerű token-t, amelyet eltárol az adatbázisban egy lejárati idővel együtt.  
Létrehoz egy jelszó visszaállító linket, amely ezt a tokent használja.  
Az e-mailt elküldi a felhasználónak PHPMailer segítségével.  
Ha az e-mail küldése sikeres, visszatér egy JSON válasszal.

### 1.6.9 Ehhez tartozik még a FORM:





Kétféle módon működik:

POST kérés: Ha a felhasználó már kapott egy visszaállító e-mailt és beírja az új jelszavát.  
GET kérés: Ha valaki rákattint a jelszó-visszaállító linkre, megnyílik az oldal, ahol beírhatja az új jelszót.

A rendszer ellenőrzi, hogy a felhasználó valóban jogosult-e a jelszó megváltoztatására.

**1. Ha a felhasználó elküldi az új jelszót (POST kérés)**

A program ellenőrzi, hogy mindkét adat meg van-e adva.  
Megkeresi az adatbázisban, hogy a token érvényes-e.  
Ha a token rendben van:

* + Az új jelszót titkosítva menti el.
  + A token törlésre kerül, hogy ne lehessen újra felhasználni.
  + Sikeres választ küld vissza.

Ha a token hibás vagy lejárt, hibaüzenetet küld vissza.

**2.** **Ha a felhasználó megnyitja a jelszó-visszaállító oldalt (GET kérés)**

A program ellenőrzi, hogy van-e token az URL-ben.  
Ha igen, megkeresi az adatbázisban az e-mail címhez tartozó adatokat.  
Ha a token érvényes:

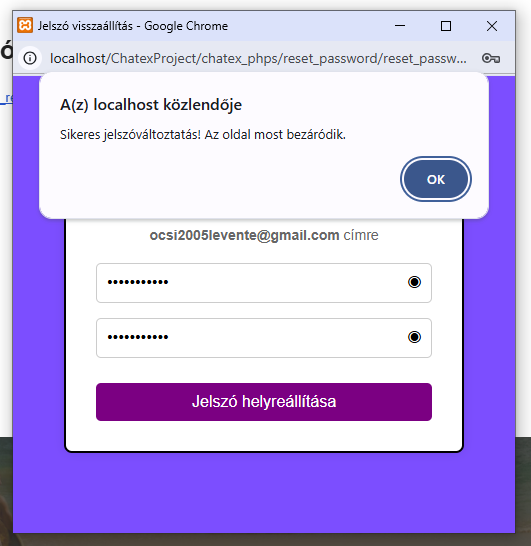
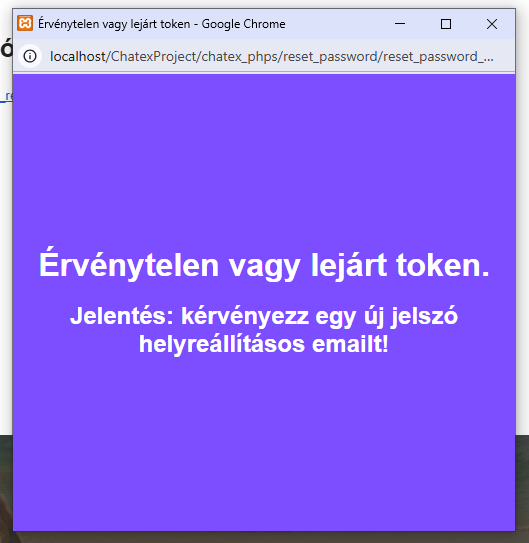
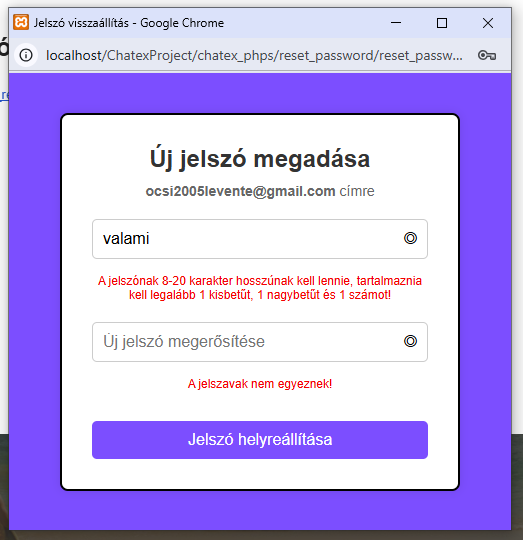
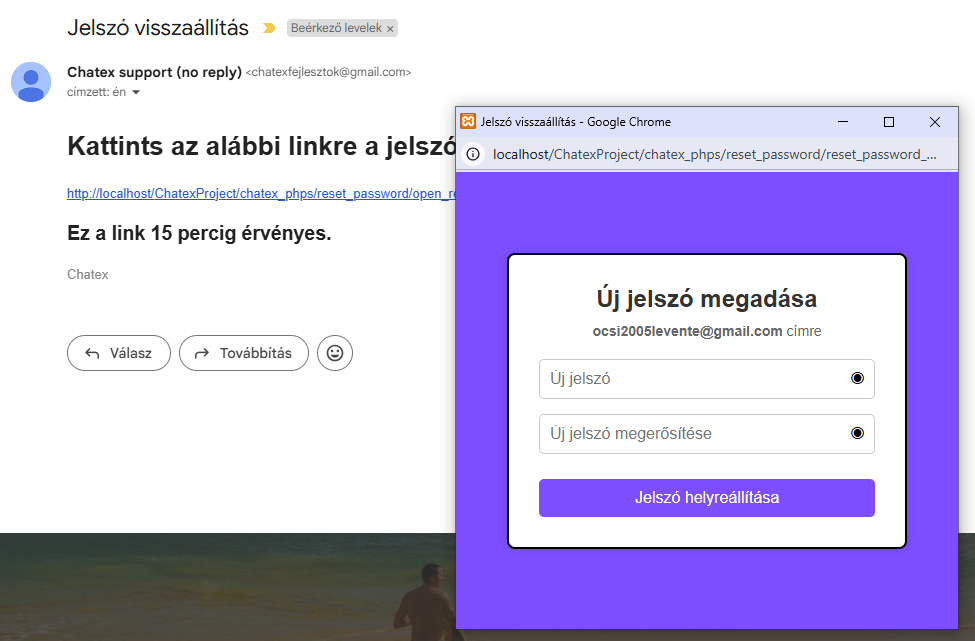
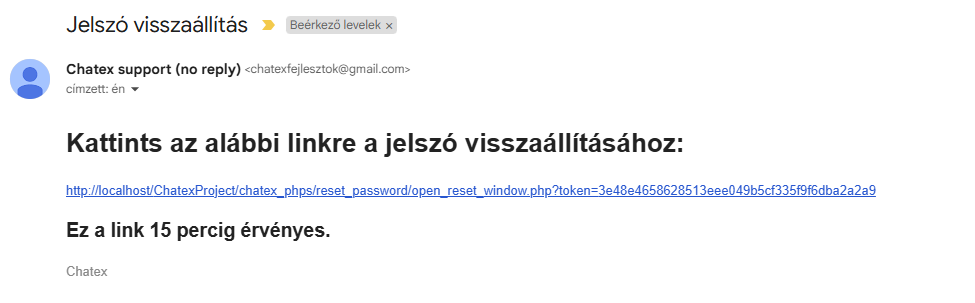
* + Megnyílik egy oldal, ahol a felhasználó beírhatja az új jelszavát.

Ha a token lejárt vagy hibás:

* + Egy hibaüzenetes oldal jelenik meg.

A weboldal tartalmaz:

* Két jelszó mezőt (az új jelszó és annak megerősítése).
* Ellenőrzést, hogy a jelszó megfelelő-e:
  + Legalább 8 karakter hosszú legyen.
  + Legyen benne kis- és nagybetű.
  + Tartalmazzon legalább egy számot.
* Hibaüzeneteket, ha valami nem stimmel.



## 1.7 Tesztelési Dokumentáció

A tesztelés célja az, hogy biztosra menjünk, hogy az applikációnk különböző funkciói (pl.: regisztráció, bejelentkezés, jelszó helyreállítás, ismerősök hozzáadása, stb.) helyesen működjenek.

* Ennek két fő része a pozitív és a negatív tesztelés:
  + A pozitív teszteléssel helyes adatokat megadva vizsgáljuk, hogyan veszi át az applikáció és mit csinál vele.
  + Negatív teszteléssel rossz adatot megadva vizsgáljuk azt, hogyan viselkedik és reagál erre az applikáció

### 1.7.1 Regisztráció tesztek forgatókönyve:

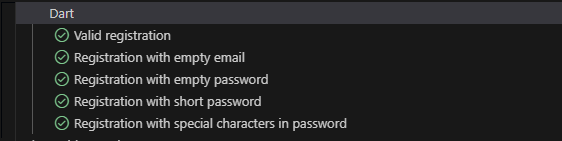
* Test 1: Regisztráció helyes adatokkal.
* Test 2: Regisztráció üres emaillel.
* Test 3: Regisztráció üres jelszóval.
* Test 4: Regisztráció rövid jelszóval.
* Test 5: Regisztráció különleges karakterekkel.

A teszt helyes adatok esetén sikeresen regisztrál és az adatok megtalálhatók lesznek az adatbázisban.

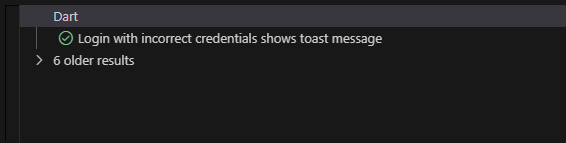
Helytelen adatok esetén megjelenik a hibát jelző toastmessage.

### 1.7.2 Tesztek pontosabb ismertetése:

* **Test 1:**
  + Azonosító: Valid registration
  + Elvárt adatok:
    - Email: validEmail
    - Jelszó: validPassword123
  + Eredmény: sikeresen regisztrál és nem ad ki hibaüzenetet.
* **Test 2:**
  + Azonosító: Registration with empty email
  + Elvárt adatok:
    - Email:
    - Jelszó: validPassword123
  + Eredmény: Sikertelen regisztrálás, toastmessage megjelenik, hogy nem lehet üres emaillel regisztrálni.
* **Test 3:**
  + Azonosító: Registration with empty password
  + Elvárt adatok:
    - Email: validEmail
    - Jelszó:
  + Eredmény: Sikertelen regisztrálás, toastmessage megjelenik, hogy hibás a jelszó.
* **Test 4:**
  + Azonosító: Registration with short password
  + Elvárt adatok:
    - Email: validEmail
    - Jelszó: 123
  + Eredmény: Sikertelen regisztrálás, toastmessage kiírja, hogy túl rövid a jelszó.
* **Test 5:**
  + Azonosító: Registration with special characters in password
  + Elvárt adatok:
    - Email: validEmail
    - Jelszó: @nval!dPassword123)
  + Eredmény: Sikeres regisztrálás és bekerül az adatbázisba



### 1.7.3 Bejelentkezési teszt:

* Rossz adatok beírásával teszt
* Azonosító: Login with incorrect credentials shows toastmessage
* Elvárt adatok:
  + Email: [wrongemail@example.com](mailto:wrongemail@example.com)
  + Jelszó: wrongpassword
* Eredmény: A bejelentkezés gomb rányomására megjelenik a toastmessage ami azt írja, hogy „Hibás Email vagy jelszó!” pirosban.

### 1.7.4 Felhasználó keresés teszt:

* Az applikáció felületén rányom az ismerősök gombra a jobb alsó sarokban, ami átviszi az ismerősök felületre, az ott lévő írható felületre beírja azt, hogy „valaki2”, a felület kimutatja a felhasználó profilját, és rányom a barát hozzáadás gombra.
* Azonosító: FindValaki test
* Eredmény: Sikeresen megtalálja a valaki2 felhasználót és sikeresen elküldi az ismerősnek jelölést

# Felhasználói dokumentáció

Elérhetőségünk: chatexfejlesztok@gmail.com

A Chatex egy chat applikáció, amit arra szeretnénk fejleszteni, hogy majd jobb legyen, mint a Messenger, mert szerintünk azért, mert szétpakolnak egy chat applikációt sok haszontalan képességgel, nem lesz attól jobb, csak nehezebb lesz rajta kiigazodni, és mi ezen szeretnénk javítani.

Hardver követelmények: Modern érintőképernyős eszköz

Operációs rendszer: Legalább Android 5.0-ás verzió

Szoftver követelmények: kell, hogy az applikációnak tudja telefonja futtatni.

Internetkapcsolat: legalább 5 Mbps

Minimum képernyőfelbontás: 360x640 pixel.

Mivel az applikációnk egy chat alkalmazás ezért minden egyes funkcióhoz sajnos kell regisztráció, ahoz, hogy hogyan kell regisztrálni, a következő oldalon megtalálod

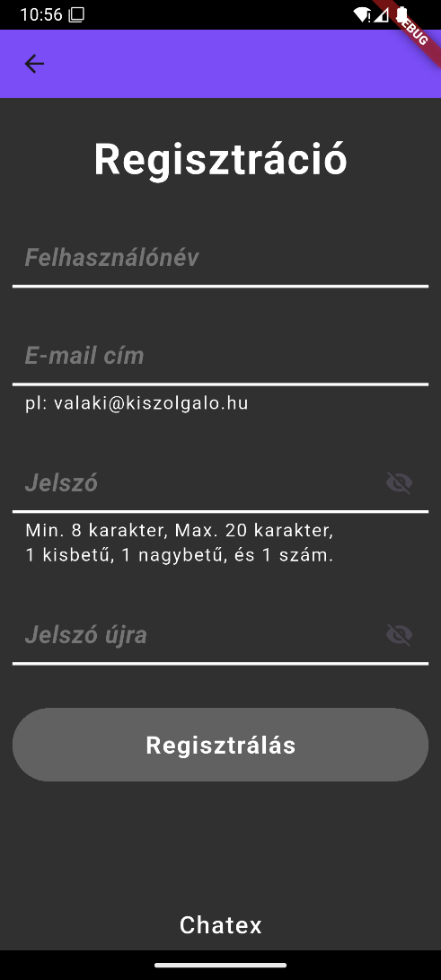
* Főoldal

Az alkalmazás/weboldal megnyitása után megjelenő felület (ha a felhasználó nincs bejelentkezve).

A felületen található:

Regisztrációs felület, ami tartalmaz:

* Felhasználónév
* Email cím
* Jelszó
* Jelszó megerősítése
* Regisztrálás gomb
* Jelszó megjelenítése
* Megváltozik a mező színe ha nem   
  helyes adatot adnak meg
* Toastmessage-el jelzi, hogy sikeres-e a regisztráció
* A felhasználónév rész elvár legalább három karaktert és nem mehet túl 20 karakteren
* Az email cím rész elvár egy rendes email formátumot, szóval @-jel, az email nevezője (Pl.: gmail) és a domain név (Pl.: .com).
* A jelszó rész elvár minimum 8 karaktert, amiben kell lennie 1 kisbetűnek, 1 nagybetűnek és egy számot, maximum 20 karakter
* A képen szöveg, képernyőkép, Betűtípus, szoftver látható

  Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A jelszó megerősítésnél ugyan az, mint a jelszónál, csak még meg kell egyeznie a jelszóval is.

Bejelentkezési felület, ami tartalmaz:

* Email cím
* Jelszó
* Bejelentkezés gomb
* Jelszót megjelenítése
* elfelejtett jelszó mező
* Nyelv megváltoztatása
* Toastmessage-el jelzi, hogy sikeres-e a bejelentkezés

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A fő képernyőn fér hozzá a felhasználó az alkalmazásunk fő részeihez  
mint például maga a chatekhez, a csoportokhoz, ahol egyszerre több személy tud

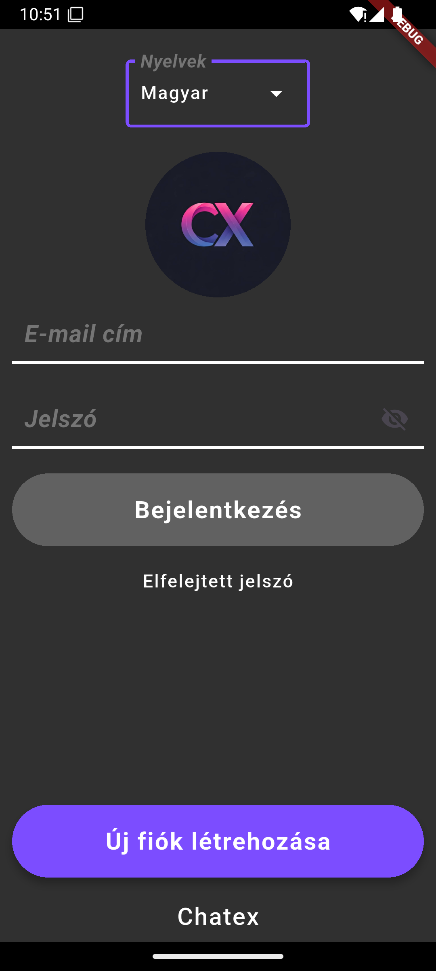
egy helyen chatelni egymással, beállítások menü, ahol minden fontosat beállíthat, Ismerősök kezelése és jelölése.

## 2.1 Applikáció használatának részletes ismertetése

Itt az applikáció bejelentkezési felületét látod, innét tudsz bejelentkezni a fiókodba, hogy el tudj kezdeni chatelni, regisztrálni, ha nem lenne még fiókod, megváltoztatni az alkalmazás által használt nyelvet és a jelszavadat helyreállítani, ha elfelejtetted, vagy csak újat akarsz csináltatni.

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.Ha van fiókod akkor az emailed és jelszavad beírásával be tudsz jelentkezni, ha elfelejtetted a jelszavad, akkor nyomj az elfelejtett jelszó gombra és ha még nincs fiókod akkor készíthetsz egyet az Új fiók létrehozása gombal.



### 2.1.1 Elfelejtett folyamatának ismertetése

A képen szöveg, képernyőkép, Betűtípus, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A képen szöveg, képernyőkép, szoftver, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.Elfelejtett jelszó felület:

1. Megadja a felhasználó az email címét
2. Megnyomja a Jelszó helyreállítása gombot

Ha helyes email címet adott meg, akkor az email fiókjában találja az üzenetet az új jelszó kéréséről.

### 2.1.2 Barát keresése folyamat ismertetése

A képen szöveg, képernyőkép, multimédia, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A fő felületen rányomsz a „Ismerősök (Friends)” oldalra, ami előhozza a bal oldali képen látható felületet, oda kell beírnod egy fióknak a felhasználónevét, ha van hasonló találat vagy teljes egyezés, akkor kimutatja azt a felhasználót, utána már csak az „Jelölés (Add)” gombra kell rányomnod és el is küldted a felhasználónak a barátkérelmed.

### 2.1.3 Barátjelölések elfogadása

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.Ha kaptál egy barátkérelmet, akkor bármikor amikor az Ismerősök felületre lépsz, kimutatva lesz piros számmal a „Barát jelölések” oldal, rákattíntva kifogja mutatni azokat a felhasználókat, akiknek a barátkérelmeit nem fogadtad vagy utasítottad el, ha el szeretnéd fogadni, csak a zöld pipára kell nyomnod, és ha el akarod utasítani, akkor a piros X-re.

2.1.4 Beállítások navigálása

A beállítások felületre nyomva előhúzza a bal oldali képernyőt, itt meg tudod változtatni az applikáció nyelvét, ki és be tudod kapcsolni az értesítéseket, a fiók adatait meg tudod nézni és a jelszavadat módosítani, még a Nyelv menüt bővíteni fogjuk több nyelvel.

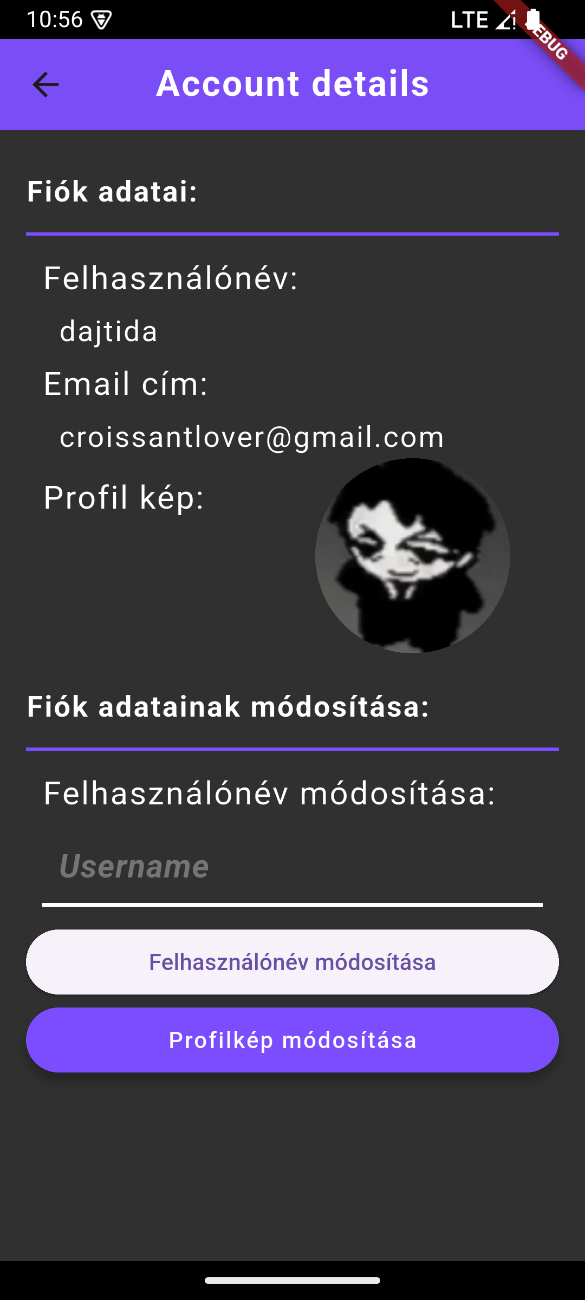
A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A képen képernyőkép, szöveg, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

2.1.5 Fiók módosítása

A fiók felületen meg tudod tekinteni az adataidat és néhány dolgot még módosítani is tudsz, mint például a profilképedet megváltoztatni akármilyen png, jpeg fájlal és a felhasználónevedet is megváltoztathatod.

* 1. További, még nem implementált ötleteink

Még biztosan tovább lehetne fejleszteni ezt az applikációt, már van pár ötletünk is, Például:  
Kétlépcsős hitelesítés, hogy biztonságosabb legyen a felhasználók fiókja.

Bármikor elérhető szerver kapcsolat

1. Összefoglalás

## 3.1 Munkamegosztás

* A munka során az együttműködés Discordon történő hívásokon át és a Github verziókezelő segítségével lett biztosítva.
* **Ötlettervező, projektvezető és programozó (Kiss Levente):** A Projekt témájának kitalálója, feladatok elosztása, A többi csapattárs közötti kapcsolattartás.
* **Fejlesztők (Kiss Levente, Szép Dániel):** A chat applikáció Frontend és Backend fejlesztése, tesztek megírása, adatbáziskezelés, funkciók megírása.
* **Designer (Kiss Levente, Szép Dániel):** A chat applikációnk kinézetének tervezése, megírása, az összes felületen lévő egységesítése.
* **Dokumentáció megírása (Szép Dániel, Kiss Levente):** Az applikációt teljes fokát átölelő dokumentáció megírása, végén utolsó korrektúraolvasás.

## 3.2 Főbb feladatok:

* **Kiss Levente:** 
  + Applikáció programjainak megírása
  + **XAMPP szerver konfigurálása és fenntartása:** PhpMyAdmin, MySQL, Apache szerver a chat applikáció teszteléséhez.
  + Az applikáció fő dizájnjának elkészítése, amit a végén egységessé teszünk az applikáción keresztül.
  + Alkalmazásnak naprakészen tartása.
  + Az alkalmazásnak különböző méreteken való futtatása.
  + Dokumentáció korrektúraolvasása.
* **Szép Dániel:**
  + Tesztesetek megírása.
  + Dokumentáció és prezentáció megalkotása
  + Program nyelvhelyességének átnézése.

### Szabó Richárd:

Sajnos Szabó Richárd osztálytársunk nem bírta követelményeinket teljesíteni, mert el volt foglalva a vizsgán kívüli dolgokkal (Pl.: Sorozatok nézésével).

1. Köszönetnyilvánítás