**Chatex**

**Dokumentáció**

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

**Készítette:**

**Szép Dániel,**

**Kiss Levente,**

**(Szabó Richárd)**

# Tartalomjegyzék

Tartalomjegyzék

[Tartalomjegyzék 2](#_Toc194994268)

[1. Bevezetés 5](#_Toc194994269)

[1.1 A témaválasztás megindoklása 5](#_Toc194994270)

[1.2 Célkitűzés 5](#_Toc194994271)

[1.2.1 Főbb jellemzők: 5](#_Toc194994272)

[1.3 Fejlesztői környezet 5](#_Toc194994273)

[1.3.1 Fejlesztői eszközök 6](#_Toc194994274)

[1.4 Elvárások a feladattal kapcsoltban 7](#_Toc194994275)

[1.4.1 Frontend Technológia: 7](#_Toc194994276)

[Flutter 7](#_Toc194994277)

[HTML (Hypertext Markup Language) 8](#_Toc194994278)

[1.4.2 Backend Technológia: 8](#_Toc194994279)

[PHP (Hypertext Preprocessor) 8](#_Toc194994280)

[MySQL 9](#_Toc194994281)

[Dart 9](#_Toc194994282)

[2. Fejlesztői dokumentáció 11](#_Toc194994283)

[2.1 Github környezet 11](#_Toc194994284)

[2.2 A kialakított adatszerkezet részletes bemutatása 12](#_Toc194994285)

[2.2.1 Az adatbázis táblái 12](#_Toc194994286)

[2.2.2 A Users tábla 13](#_Toc194994287)

[2.2.3 A messages tábla 14](#_Toc194994288)

[2.2.4 A friends tábla 15](#_Toc194994289)

[2.2.5 A friend requests tábla 16](#_Toc194994290)

[2.2.6 Felhasználó regisztrálása 17](#_Toc194994291)

[2.2.7 Felhasználó bejelentkezés 20](#_Toc194994292)

[2.2.8 Jelszó helyreállítás 22](#_Toc194994293)

[2.2.9 Ehhez tartozik még a FORM: 24](#_Toc194994294)

[2.3 Fájlstruktúra 30](#_Toc194994295)

[A PHP fájlok funkciói: 30](#_Toc194994296)

[**1.** **Auth** 30](#_Toc194994297)

[**2.** **Chat** 30](#_Toc194994298)

[**3.** **Friends** 30](#_Toc194994299)

[**4.** **Reset password** 31](#_Toc194994300)

[**5.** **Settings** 31](#_Toc194994301)

[2.4 Tesztelési Dokumentáció 33](#_Toc194994302)

[2.4.1 Regisztráció tesztek forgatókönyve: 33](#_Toc194994303)

[2.4.2 Tesztek pontosabb ismertetése: 33](#_Toc194994304)

[2.4.3 Bejelentkezési teszt: 35](#_Toc194994305)

[2.4.4 Felhasználó keresés teszt: 35](#_Toc194994306)

[3. Felhasználói dokumentáció 37](#_Toc194994307)

[3.1 Applikáció használatának részletes ismertetése 41](#_Toc194994308)

[3.1.1 Elfelejtett folyamatának ismertetése 42](#_Toc194994309)

[3.1.2 Barát keresése folyamat ismertetése 43](#_Toc194994310)

[3.1.3 Barátjelölések elfogadása 44](#_Toc194994311)

[3.1.4 Beállítások navigálása 45](#_Toc194994312)

[3.1.5 Fiók módosítása 46](#_Toc194994313)

[3.1.6 További, még nem implementált ötleteink 47](#_Toc194994314)

[4. Összefoglalás 48](#_Toc194994315)

[4.1 Munkamegosztás 48](#_Toc194994316)

[4.2 Főbb feladatok: 48](#_Toc194994317)

* [**Kiss Levente:** 48](#_Toc194994318)
* [**Szép Dániel:** 49](#_Toc194994319)
* [Szabó Richárd: 49](#_Toc194994320)

[5. Köszönetnyilvánítás 50](#_Toc194994321)

# 1. Bevezetés

## 1.1 A témaválasztás megindoklása

Alkalmazásunk a Messenger jelenlegi verziójának (a készítéskor: v485.2.0.68.111) ad egy letisztult, egyszerű, fölösleges funkciók nélküli, konzisztens dizájnnal rendelkező alternatívát!

Ez az alkalmazás azoknak az embereknek szól, akik ugyanúgy nem kedvelik a Messenger felsorolt hibáit (akár még többet is) és felesleges, nem használt funkcióit!

## 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, miközben ugyanúgy megtartja a játékos, személyiséget kifejező (kép küldés, fájl küldés) funkciókat.

### 1.2.1 Főbb jellemzők:

1. Villámgyors üzenetküldés késleltetés nélkül.
2. Biztonságos adatkezelés és titkosítás a felhasználók adatainak védelméért.
3. Modern, letisztult felület, amely könnyen kezelhető.
4. Ellenben a Messengerrel a Chatex modern kódra épül, ami egy megbízható és remekül kezelhető Framework köré épül (Flutter) így a közel jövőben kiadott verziók tökéletesen fogják fedni a kitűzött célokat!

## 1.3 Fejlesztői környezet

A chat applikációnk elkészítéséhez többféle fejlesztőeszközt használtunk. A kód nagy részét Visual Studio Code-ban írtuk meg, de használtunk Android Studio-t is a fejlesztéshez. Később viszont áttértünk véglegesen a VS Code-ra mivel a nagy támogatás az extensionök-re kulcs fontosságú volt a csoportmunka és egyéb dolgok kivitelezésére (Pl.: Live Share extension).

A Flutter Framework használata úgy indokolható, hogy lehetővé teszi, hogy a programot több platformon is tudjuk futtatni egyetlen egy Dart nyelven íródott kóddal! Így remekül kibővíti az alapból backend-re szánt Dart nyelvet a Flutter különböző kiegészítései (csomagjai) amit mi is használtunk bizonyos funkciók elérése érdekében, amik a [pub.dev](https://pub.dev/) oldal segítségével nem sikerültek volna. A hab a tortán az hogy a Google által fejlesztett ez a Framework ami azt jelenti, hogy remek támogatással rendelkezik, és tökéletesen implementálható bármilyen szolgáltatás, amit a Google fejlesztett pl.: Firebase. A következő ábrán látszik hogy az Android-ra fejlesztett alkalmazáshoz milyen rendszerkövetelmények szükségesek:

|  |  |  |
| --- | --- | --- |
| Platform | Minimum OS | Támogatott Architektúrák |
| Android | Android 5.0+ | arm32, arm64, x86\_64 |

### 1.3.1 Fejlesztői eszközök

Mind VS Code-ba, mind Android Studio-nál szükség volt a fejlesztéshez egy Android-ot futtató emulátorra (lehetett volna fizikai eszközt is használni, de mi esetünkben a lokális futtatás maradt az egyetlen járható út. Emiatt is folyamodtunk ahhoz, hogy egy jól kezelhető, átlátható szerverünk legyen, ahol jól tudjuk majd megírni a backend-et! Ez volt az XAMPP, ahol a webszerveren keresztül értük el az adatbáziskezelőnket (Apache webszerver és phpMyAdmin). Innét kiindulva a backend programozásunk PHP-on keresztül ment végbe, ahol MySQL kéréseket hajtottunk végre, amit a Chatex kezdeményezett a felhasználói bemenetekre.

## 1.4 Elvárások a feladattal kapcsoltban

A célkitűzésben leírtakat sikeresen implementáltuk a fejlesztés során, de akadtak olyan részek is, amik nem mindig működtek úgy ahogy vártuk (legtöbbször ez a Flutter-Dart widgetek és a csomagok által használt kiegészítések működésével volt probléma). Sajnos az olyan eseteknél rögtönöznünk kellett, néhol még teljes funkciókat kellett feladnunk pl.: csoportok, videó küldés implementálása!

A jó hír viszont az, hogy ezek a kivételek csak ritkán jelentek meg és azokat leszámítva sikeresen, elvárt módon készült el minden funkció a Chatex 1.0-ba! Ezek között volt pl.: a sikeres „fél” valós idejű chat kommunikáció fénykép és fájl küldéssel, reszponzív felhasználói felület, ami alkalmazkodik különböző mobil eszközökhöz, és még sorolhatnám…

A chat applikációnk funkcióinak megfelelő működését, mind backend technológiák és frontend technológiák együttes működésével tudtuk biztosítani!

### 1.4.1 Frontend Technológiák:

**Frontend**-nek számít minden olyan technológia, amivel a felhasználó interakcióba képes lépni, tehát minden olyan elem, ami dizájn-t, szöveget, gombokat, beviteli mezőket, ikonokat épít fel az ide tartozik! Ilyen technológia a:

### C:\Users\User\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9DB981C1.tmpFlutter

**A Flutter egy szoftverfejlesztő készlet (SDK),** ami ráadásul a Google által készített nyílt forráskódú Frameworknek számít. Használata megkönnyíti a natív mobil- (Android, iOS), web- és asztali alkalmazások (Windows, Linux) létrehozását egyetlen kódbázisból (Dart nyelv). Fő előnye, hogy **lehetővé teszi:**

* **Platformok közötti fejlesztést:** Lehetővé teszi Android, iOS, Windows, macOS, Linux és webes alkalmazások létrehozását egyetlen kódbázissal.
* **Hot restart/Hot reload újratöltés:** A fejlesztők valós időben láthatják a kódváltozásokat anélkül, hogy a teljes alkalmazást újra kellene fordítaniuk.
* **Rugalmas felhasználói felület:** Használata testre szabható widgeteken alapszik, amelyek lehetővé teszik, hogy minden platformon natív megjelenésű és teljesítményű alkalmazásokat tervezzen.
* **Optimalizált teljesítmény:** A natív kódösszeállításnak köszönhetően az alkalmazások gyorsak és gördülékenyek minden platformon, a megfelelő eszközökkel (ha eléri persze a minimum rendszerkövetelményt, különben gyengén fog futni).
* **Több IDE támogatása:** Az alkalmazások a Flutter segítségével fejleszthetők olyan szerkesztőkben, mint az Android Studio, a Visual Studio Code, és még az IntelliJ is!

### C:\Users\User\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\858A72FB.tmpHTML (Hypertext Markup Language)

Leíró nyelv, melyet a weboldalak készítéséhez fejlesztettek ki, és mára már internetes szabvánnyá vált. Munkánkban elég kicsi mennyiségben használtunk HTML-t, de amikor kellett (jelszó helyreállító email küldése és maga az emailben elküldött jelszó helyreállító űrlap) akkor nagyon könnyen tudtuk kezelni! Mihez szokták használni:

* A HTML-lel egy webhely szerkezetét, illetve tartalmát szokás meghatározni. Létrehozhatók vele például bekezdések, címsorok, táblázatok, elhelyezhetők képek, és így tovább.
* Összes böngésző számára egy alapkövetelmény lett, hogy felismerjék (erre épül az Internet).
* Önmagában nem tud sokat alkotni, de JavaScript-el és PHP használatával egy nagyon erős leíró nyelv tud lenni, ami látszik is a jelszó helyreállítási funkciónkból (és a phpk-ből)

### JavaScript

### A JavaScript használata lehetővé tette számunkra, hogy dinamikusan építsük fel a jelszó helyreállítási formot pl.: speciális ablakméret, tulajdonságok különféle böngészőkhöz, és a PHP-val való adatátvitel és kommunikáció! Előnyei, amik megkönnyítették a munkánkat:

### Minden modern böngésző megfelelően tudja értelmezni a JavaScript kódot a beépített motorjával, ami nagyon hatékonnyá teszi a kód értelmezést. Talán mi esetünkben ez annyira nem mérvadó mert tényleg nagyon minimális a webes rész, de akár a későbbiekben nagyon hasznunkra lesz az, hogy ilyen magas szintű nyelvről beszélünk!

### Dinamikus változó deklarálás és kezelés így nem kell bajlódni azzal, hogy milyen típusokat fognak kezelni a változók értékadáskor. A mi esetünkben ez a funkció hasznos volt a képernyő méret és egyéb változók pl.: helyreállítási link értelmezése.

### Remek kommunikáció, adat értelmezés PHP-val, mert ha php változókat kezelünk, amik lehetnek elég komplexek akkor is probléma nélkül tudja őket értelmezni!

### 1.4.2 Backend Technológiák:

A **Backend** a háttérben futó folyamatokkal foglalkozik, pl.: szerveroldali programozással, űrlapon beküldött adatok feldolgozásával. Az alkalmazásunk kulcsfontosságú része, és a téma megválasztásából fakadó egyik legfontosabb kérdés, hogy milyen technológiákat kell használnunk a „fél” valós idejű chat alkalmazás megvalósításáért! Végül az XAMPP és a Flutter Frameworkből adódóan ezek a technológiák voltak kulcs fontosságúak:

### C:\Users\User\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\A4EA993F.tmpPHP (Hypertext Preprocessor)

* A **PHP** egy általános szerveroldali szkriptnyelv elsősorban dinamikus weblapok készítésére (lásd jelszó helyreállító email). Az első szkriptnyelvek egyike.
* Sok különböző platform használja különféle feladatokra, ami esetünkben relációs adatbázisok (MySQL) kezelésére szolgál SQL kérésekkel, amiket maga a Chatex küld a frontend/backend adataival.
* Remekül működik csomagkezelőkkel pl.: a Composer, ami nagy hasznunkra vált a fejlesztés során!

### C:\Users\User\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\5CD334A5.tmpMySQL

* A MySQL az egyik legelterjedtebb adatbázis-kezelő rendszer, aminek egyik fő oka az lehet, hogy a teljesen nyílt forráskódú LAMP (Linux–Apache–MySQL–PHP) összeállítás részeként költséghatékony és egyszerűen beállítható megoldást ad dinamikus webhelyek szolgáltatására, vagy egyéb adatbázist igénylő projektekre.
* MySQLi segítségével lehet integrálni a PHP-val (MySQL Improved csomag)
* PhpMyAdmin segítségével könnyebb az adatbáziskezelés, mivel támogatja a GUI használatát egy jól megírt lokálisan futó weboldalként, ahol minden funkcióját elérjük pl.: táblák létrehozását, módosítását, importálását és exportálását.

### C:\Users\User\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\5E75EF49.tmpDart

A Dart célkitűzése a webböngészők fő szkriptnyelvének, a JavaScriptnek a lecserélése (kitalálásakor egy backend nyelvnek szánták). Kísérletet tesznek a JavaScript problémáinak megoldására, miközben a nyelv jobb teljesítményt nyújt, mint a JS. Könnyebben lehet fejlesztőeszközöket alkalmazni a nagyobb szabású projektekhez, és egyben biztonságosabb is.

* Típusos nyelv: a változóknak meg kell adni a típusát a létrehozásuk során (Flutter Frameworkkel nem muszáj). Ez jó, mert így minden változóról pontosan tudjuk, hogy milyen értéket vár és a Dart is, hogy milyen jellegű adatként bánjon vele. Sőt dinamikusan típusos, ami azt jelenti, hogy akár saját típusokat is kreálhatunk és használhatunk a Dart programunkon belül.
* Null-biztos: ami megköveteli, hogy minden változónak legyen értéke. Ez megelőzi az olyan jellegű hibákat, hogy bármely változó véletlenül null értéket kapjon (sok null kezelés van az alkalmazásunkban). Az alkalmazásunkban például gyakori eset volt, hogy amikor egy értéket vettünk át az adatbázisból php-n keresztül akkor null-ként értelmezte, ami exceptionöket adott (pl.: profilkép értelmezése).
* Rengeteg különböző könyvtárral (library), beépített típussal rendelkezik.
* A kódot kétféleképpen futtathatjuk: natív platformon, ami a mobil és asztali eszközöket célozza meg. Web platformon, ami azt jelenti, hogy a Dart nyelv segítségével webalkalmazásokat is készíthetünk, melynek során a Dart a JavaScript-re fordítódik át.

# 2. Fejlesztői dokumentáció

## 2.1 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.

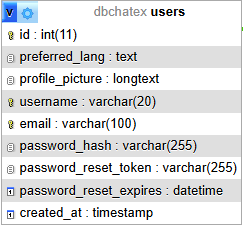
Az elkészült vizsgaremek tartalmazza az adatbázist, programkódot, dokumentációt és az előadást, amit a következő Github repositoryban érhető el: [Chatex vizsgaremek](https://github.com/smolelevent/Messengeres-vizsgaremek). itt található az összes szükséges anyag a projekt teljes megértéséhez.

## 2.2 A kialakított adatszerkezet részletes bemutatása

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

### 2.2.1 Az adatbázis táblái

### 2.2.2 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).

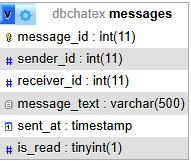
password\_hash (jelszó): Ez a mező tartalmazza az adott felhasználó által megadott jelszót, ezzel tud csak a felhasználó bejelentkezni fiókjába, és he elfelejti, új jelszót kell kérnie mert a Chatex nem fogja tudni megadni, mert az adatbázisban csak egy hash code-ként van elmentve.

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.

### 2.2.3 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.

### 2.2.4 A friends tábla

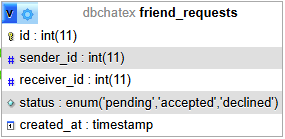
id (azonosító): Ez a mező tartalmazza a barátlistán lévő felhasználók azonosítóját, ez az azonosító az elsődleges kulcs és INT típusú, maximum 11 karaktert fogad be.

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.

### 2.2.5 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.

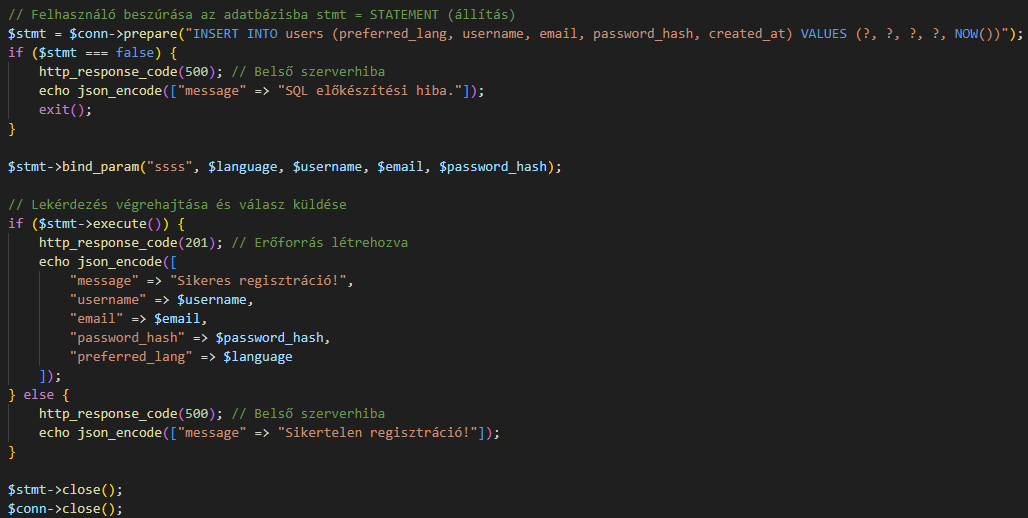
receiver\_id (barátkérelmet kapó azonosító): Ez a mező tartalmazza annak a felhasználónak az azonosítóját, aki a barátkérelmet kapta egy másik felhasználótól, aki még nincs a barátlistáján, a mező INT típusú és maximum 11 karaktert fogad be.

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.

### 2.2.6 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.

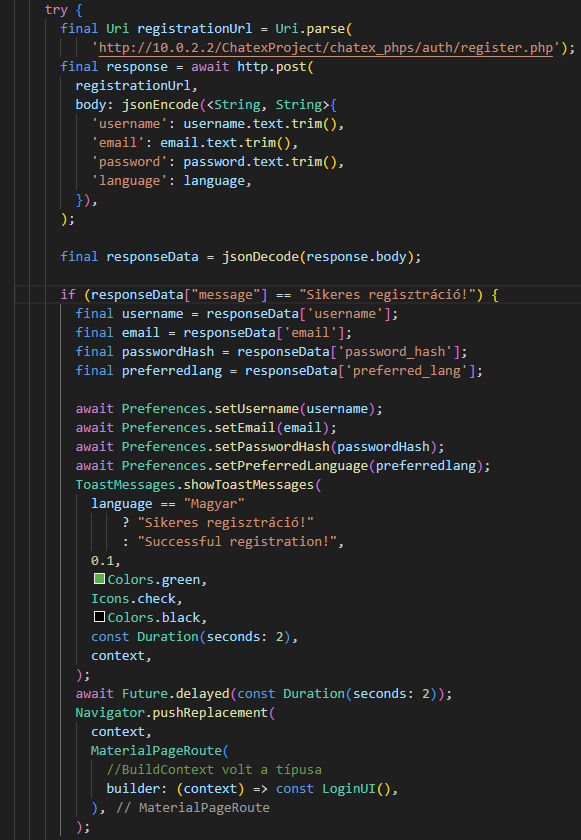
A header beállításoknál előszőr beállítjuk, hogy A válasz JSON formátumban érkezzen, bárhonnan engedélyezett a hozzáférés, Az API kizárólag **POST** metódussal hívható és az API-hoz szükséges fejléceket engedélyezi.

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

Átveszi az adatokat és be is olvassa a tárolójukba:

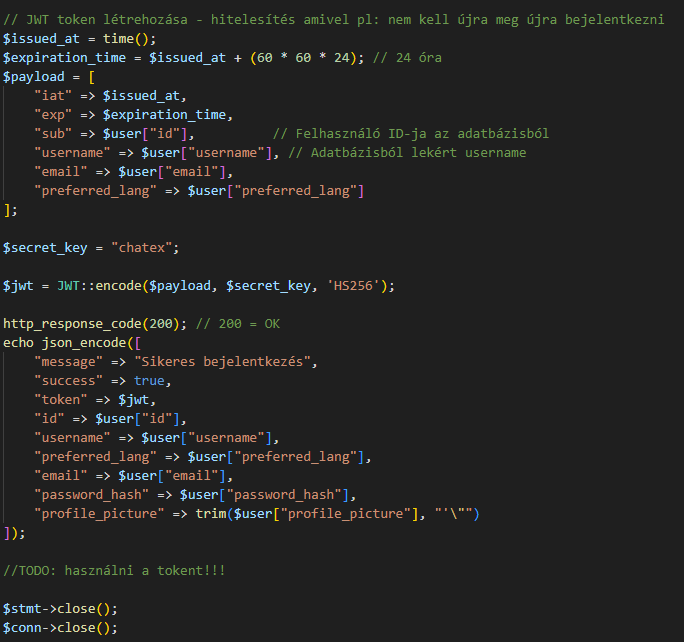
Utána ellenőrzi az email formátumát, és ha helyes, akkor tovább megy és ellenőrzi, hogy létezik-e már az email az adatbázisban, utána beszúrja az adatbázisba, ha minden jól megy és kész a regisztráció.

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

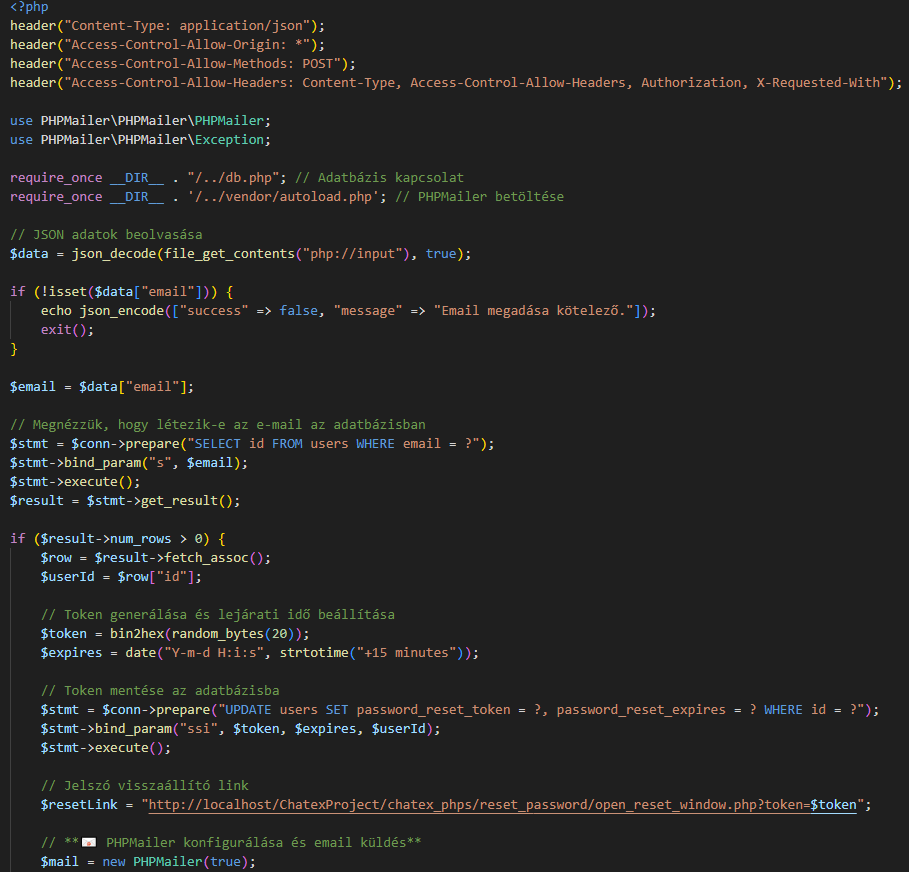


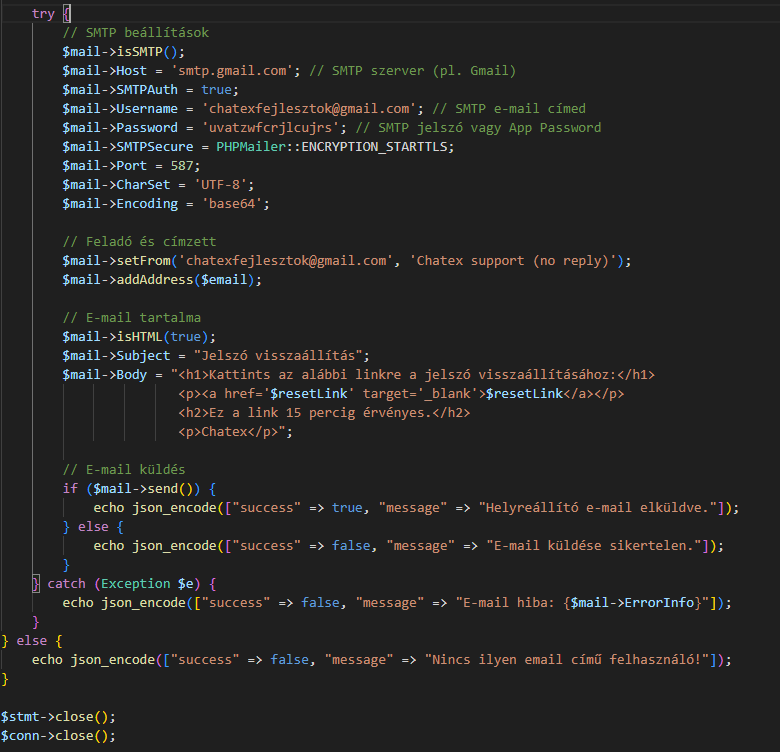
### 2.2.7 Felhasználó bejelentkezés

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.

### 2.2.8 Jelszó helyreállítás

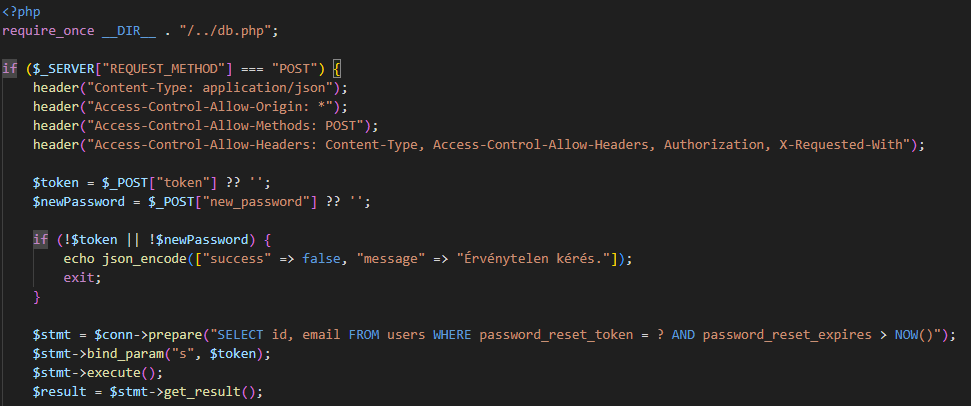


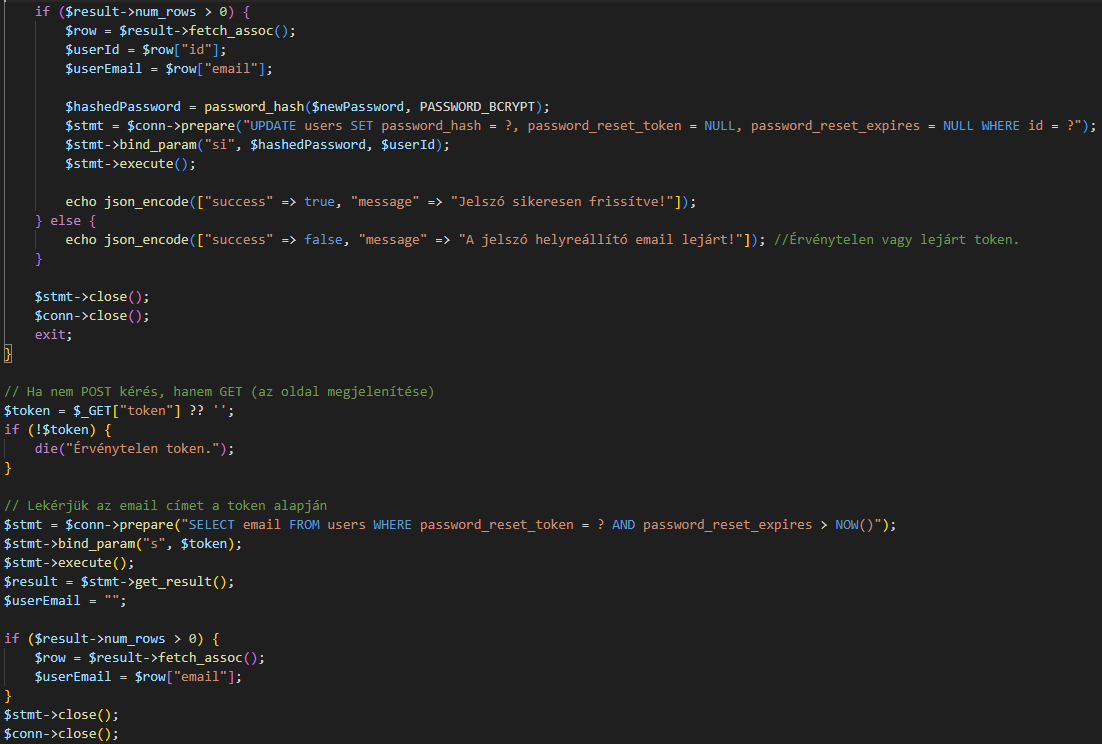


Ez egy PHP alapú rendszer, amely segít a felhasználóknak visszaállítani az elfelejtett jelszavukat. Ha valaki elfelejti a jelszavát, megadhatja az e-mail címét, és kap egy linket, amelyen keresztül új jelszót állíthat be.

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.

### 2.2.9 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:

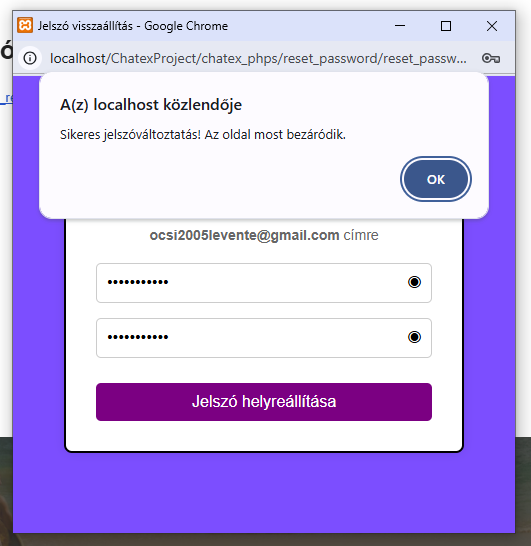
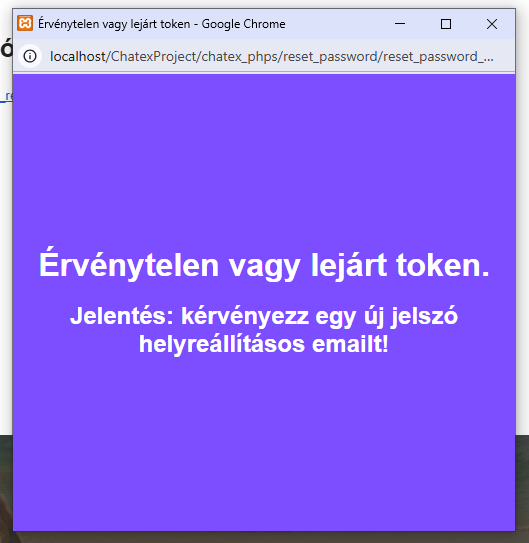
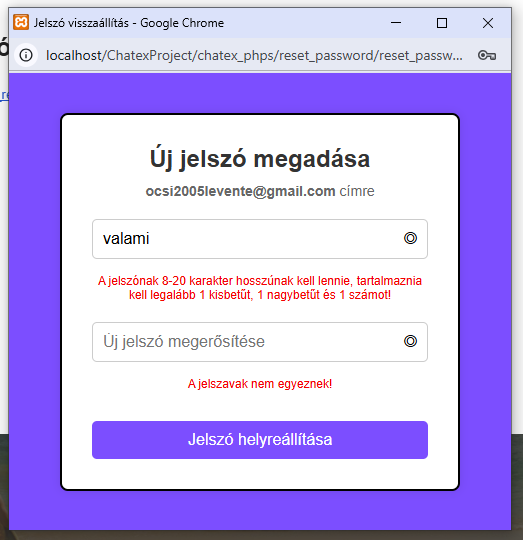
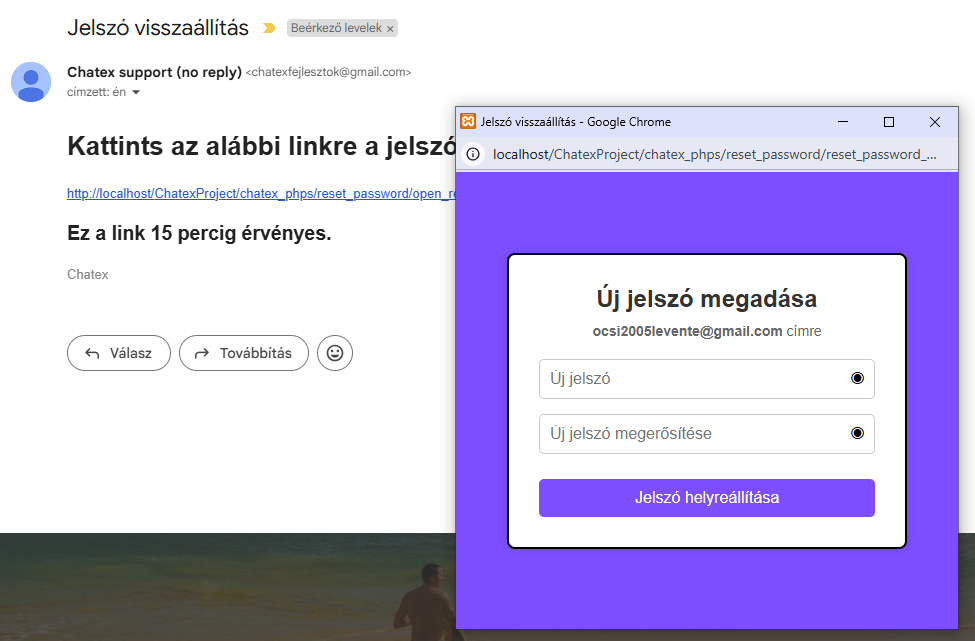
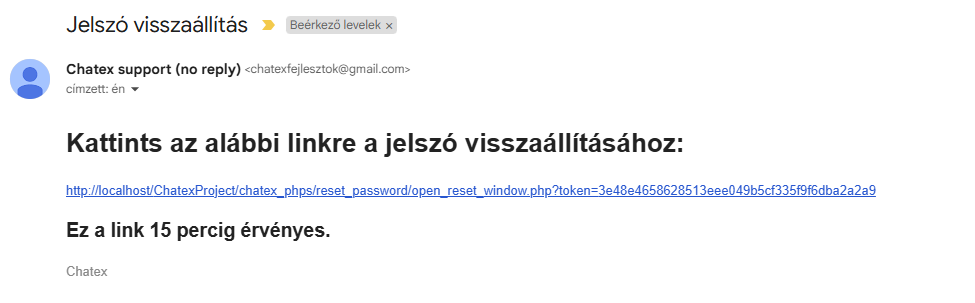
* + Megnyílik egy oldal, ahol a felhasználó beírhatja az új jelszavát.

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.



## 2.3 Fájlstruktúra

A Chatex applikáció fájlendszerének felépítése, ami az alkalmazás működéséhez szükséges.

## A PHP fájlok funkciói:

1. **Auth**

* Tartalmazza az applikációnknak az összes authentikációs funkcióját
* login.php: Bejelentkezési funkció
* Register.php: Regisztrációs funkció
* logout.php: Kijelentkezési funkció
* A képen szöveg, képernyőkép, szoftver, multimédia látható

  Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.update\_status.php: A felhasználó státuszának ellenőrzése (ha nyitva van a mobilon az applikáció akkor Online, különben offline)
* validate\_token.php: Megnézi, hogy a token még érvényes-e vagy már elavult

1. **Chat**

* Tartalmazza a chathez szükséges funkciókat
* get\_chats.php: Kimutatja a felhasználónak a jelenlegi csevegéseit
* get\_friend\_list.php: Kimutatja a barátlistát.
* get\_group\_chats.php: Kimutatja a felhasználó által felvett csoportokat
* get\_messages.php: Kimutatja a felhasználó és egy másik személy közötti üzeneteket
* start\_chat.php: Csevegés elindítása egy ismerőssel

1. **Friends**

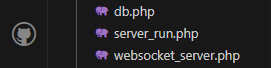
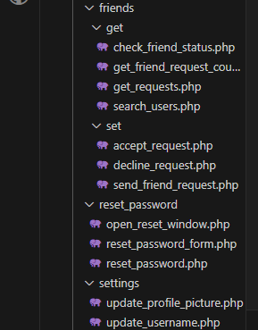
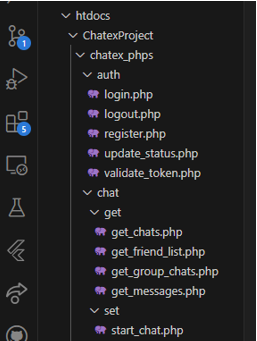
* check\_friend\_status.php: Kimutatja az ismerőseidnek a státuszát (Online – Offline)
* get\_friend\_request\_count.php: Az összes még el nem fogadott barátkérelmek számát mutatja ki
* get\_requests.php: Kapott barátkérelmek kimutatása
* search\_users.php: Felhasználók keresése
* accept\_request.php: barátkérelmek elfogadása
* decline\_request.php: barátkérelmek elutasítása
* send\_friend\_request.php: barátkérelmek elküldése

1. **Reset password**

* open\_reset\_window.php: Megnyitja a jelszó helyreállításhoz szükséges ablakot
* reset\_password\_form.php: Egy lekicsinyített ablakban megjelenő Form a jelszó helyreállításához
* reset\_password.php: Jelszó helyreállítása

1. **Settings**

* update\_profile\_picture.php: Új profilkép beszúrása a felhasználónak
* update\_username.php: Új felhasználónév készítése



## 2.4 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

### 2.4.1 Regisztráció tesztek forgatókönyve:

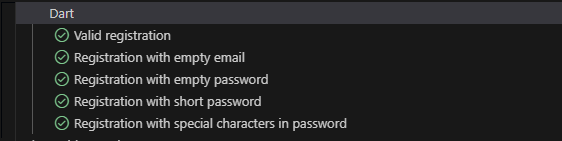
* Test 1: Regisztráció helyes adatokkal.
* Test 2: Regisztráció üres emaillel.
* Test 3: Regisztráció üres jelszóval.
* Test 4: Regisztráció rövid jelszóval.
* Test 5: Regisztráció különleges karakterekkel.
* Test 6: Bejelentkezés rossz adatokkal.
* Test 8: Felhasználó „valaki2” megkeresése.

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

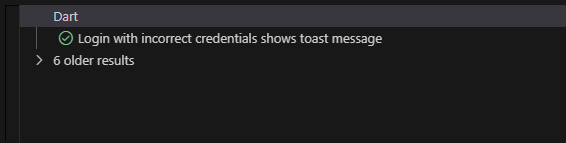
Helytelen adatok esetén megjelenik a hibát jelző toastmessage.

### 2.4.2 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



### 2.4.3 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.

### 2.4.4 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

# 3. Felhasználói dokumentáció

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, ahhoz, 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 címA képen szöveg, képernyőkép, szoftver, Betűtípus 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.
* 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 jelszó megerősítésnél ugyan az, mint a jelszónál, csak még meg kell egyeznie a jelszóval is.

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.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 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, Ismerősök kezelése és jelölése beállítások és menü, ahol minden fontosat beállítha.

A képen szöveg, képernyőkép, szoftver, multimédia látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

## 3.1 Applikáció használatának részletes ismertetése

Itt az applikáció bejelentkezési felületét lehet látni, innét tud a felhasználó bejelentkezni a fiókjába, hogy el tudjon kezdeni chatelni, regisztrálni, ha nem lenne még fiókja, megváltoztatni az alkalmazás által használt nyelvet és a jelszavát helyreállítani, ha elfelejtette, vagy csak újat akar csináltatni.

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.Ha van fiókja akkor az emailje és jelszava beírásával be tud jelentkezni, ha elfelejtette a jelszavát, akkor nyomjon az elfelejtett jelszó gombra és ha még nincs fiókja akkor készíthet egyet az Új fiók létrehozása gombal.

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

### 3.1.1 Elfelejtett folyamatának ismertetése

A képen szöveg, képernyőkép, szoftver, Betűtípus 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.

### 3.1.2 Barát keresése folyamat ismertetése

A fő felületen rányom a felhasználó az „Ismerősök (Friends)” oldalra, ami előhozza a bal oldali képen látható felületet, oda kell beírnia egy fióknak a felhasználónevét, ha van hasonló találat vagy teljes egyezés, akkor kimutatja azt a felhasználót, utána már csak az „Jelölés (Add)” gombra kell rányomni és el is küldte a felhasználónak a barátkérelmet.

A képen szöveg, képernyőkép, szoftver, multimédia látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

### 3.1.3 Barátjelölések elfogadása

Ha kapott a felhasználó egy barátkérelmet, akkor bármikor amikor az Ismerősök felületre lép, ki lesz mutatva piros számmal a „Barát jelölések” oldal, erre rányomva kifogja mutatni azokat a felhasználókat, akiknek a barátkérelmeit nem fogadta vagy utasította el a felhasználó, ha el szeretné fogadni, csak a zöld pipára kell nyomnia, és ha el akarja utasítani, akkor a piros X-re.

A képen szöveg, képernyőkép, szoftver, multimédia látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.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.

**3.1.4 Beállítások navigálása**

A képen képernyőkép, szöveg, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A beállítások felületre nyomva előhúzza a bal oldali képernyőt, itt meg tudja a felhasználó változtatni az applikáció által kimutatott nyelvet, ki és be tudja kapcsolni az értesítéseket, a fiók adatait meg tudja nézni és a jelszavát 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, multimédia, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

3.1.5 Fiók módosítása

A fiók felületen meg tudja a felhasználó tekinteni az adatait és néhány dolgot még módosítani is tud, mint például a profilképét megváltoztatni akármilyen png, jpeg fájlal és a felhasználónevét is megváltoztathatja, vagy akár még törölheti is.

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.

* + 1. További, még nem implementált ötleteink

Még biztosan tovább lehetne fejleszteni ezt az applikációt, már van pár ötletünk is, Például:

1. Kétlépcsős hitelesítés, hogy biztonságosabb legyen a felhasználók fiókja.
2. Bármikor elérhető szerver kapcsolat.
3. Több testreszabhatósági lehetőség profilok és chatekhez az egyéniség kedvéért.
4. Még több nyelv hozzáadása, hogy még több felhasználó tudja élvezni applikációnkat
5. Egyszerű bejelentkezés Google fiók segítségével
6. Híváson át történő csevegés

**4. Összefoglalás**

## 4.1 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.
* **Mesteri munkakerülés (Szabó Richárd):** Hívásokba nem megjelenés, hamar lelépés más és más kifogásokkal, kapcsolattartás leszarása.

## 4.2 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.
* **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).

**4.3 Elérhetőségeink:**

**Email cím:** [chatexfejlesztok@gmail.com](mailto:chatexfejlesztok@gmail.com)

**Telefon:** +36 20 542 6746

1. Köszönetnyilvánítás

Ezúton szeretnénk megköszönni **Bloch Tamás** tanárúrnak, a folyamatos támogatást, szakmai iránymutatást és türelmet, amellyel végigkísérte a munkáink elkészítését. Nélküle ez a vizsgaremek nem jöhetett volna létre ilyen formában.

Külön köszönettel tartozunk **Pradalits Tibor** igazgató úrnak, aki lehetővé tette a projekt megvalósítását, valamint biztosította a szükséges feltételeket és hátteret a munkához.

Hálásak vagyunk minden segítségért és biztatásért, amit a folyamat során kaptunk.