BKSZC Weiss Manfréd Technikum, Szakképző Iskola és Kollégium

VIZSGAREMEK

Témavezető:   
Kovács László

Készítette:   
Csonka Dominik,   
Mohai Olivér

BUDAPEST  
2025

# Bevezetés

## Témaválasztás indoklása

Mindenképp szerettünk volna egy olyan weboldalt létrehozni, amely egy valós problémára ad gyakorlati megoldást. Emellett fontos szempont volt, hogy a projekt kihívás elé állítson bennünket, és lehetőséget adjon a már meglévő ismereteink fejlesztésére. Kezdetben többféle elképzelés is felmerült, azonban ezek többsége nem igényelt volna többfős együttműködést, így a projekt témájának kiválasztása nehézséget okozott. Tapasztalataink szerint, az iskolánk tanulói nem mindig kapták meg időben a diákönkormányzati gyűlésen elhangzott információkat. Ez alapján kaptunk egy olyan témát, amely ennek a kommunikációs problémának az áthidalását célozza. Bár kezdetben attól tartottunk, hogy ebben a projektben is nehéz lenne a feladatokat úgy megosztani, hogy közben ne zavarjuk egymás munkáját, végül sikerült egy olyan ötletet találni, amely ezt áthidalja. Az eredeti ötletet továbbgondolva, a felmerült igények figyelembevételével született meg a gólyakereső, valamint a privát és csoportos csevegés ötlete.

## Célkitűzés

A cél egy fórumszerű oldal, valamint egy csevegő felület létrehozása volt, amely az intézmény diákjai között hoz létre egy kétirányú kommunikáció lehetőséget, legyen az egyik fél diákönkormányzati tag, vagy egy szimpla diák. A *közösségi média* és *csevegő felület* ugyan nagy részben független egymástól, de mégis találtunk megoldást arra, hogy egybekössük a kettőt a következőképp:

- A felhasználók nevére vagy profilképére kattintva az oldal gyors elérést biztosít a két felhasználó közti csevegési felülethez.

- A diákönkormányzati tagok által közzétett bejegyzésekben lehetőség van osztályok megjelölésére. Ezek a megjelölések a csevegőfelületen „rendszerüzenetként” jelennek meg, jelezve az érintett diákok számára, hogy osztályuk említve lett egy bejegyzésben.

## Kiknek szánjuk a weboldalt

A weboldalt elsősorban a Weiss Manfréd Technikum diákjainak szánjuk – legyenek azok frissen belépő „gólyák”, jelenlegi tanulók, vagy már végzett diákok. A cél, hogy az iskolai közösség tagjai könnyebben elérjék az információkat és kapcsolatba léphessenek egymással.

# Fejlesztői dokumentáció

## Fejlesztői környezet

A weboldal fejlesztése során több különböző eszközt és környezetet alkalmaztunk, amelyek segítették a munka hatékonyságát és átláthatóságát.

A **Visual Studio Code** egy könnyen kezelhető, széles körben használt kódszerkesztő, amely lehetővé tette a HTML, CSS, JavaScript és PHP kód gyors és hatékony írását.

A **XAMPP** egy olyan komplett fejlesztői csomag, amely tartalmazza az Apache webszervert, a PHP értelmezőt, valamint a MariaDB-t (MySQL-kompatibilis adatbázis-kezelő). Ennek segítségével helyileg is futtathattuk és tesztelhettük weboldalunkat.

A **phpMyAdmin** segítségével könnyedén kezelhetjük és karbantarthatjuk az adatbázisunkat, illetve webalapú felületet biztosít a MySQL/MariaDB adatbázisok kezelésére.

A **MySQL** strukturált adatbázis-kezelő rendszer, amely biztosítja az adatok tárolását, visszakeresését és manipulációját.

Ezeket a fejlesztőeszközöket választottuk, mivel korábbi tanulmányaink során már megismertük őket, széles körben elterjedtek, ingyenesen elérhetők, és hatékonyan támogatják a weboldal fejlesztését és tesztelését.   
A projekt előrehaladtával kényelmi szempontból áttértünk egy online fájlkezelő szerkesztőre is, amely lehetővé tette, hogy bármilyen eszközről, külön letöltés nélkül elérjük és módosítsuk a forrásfájlokat.

## Domain és tárhely

Ahhoz, hogy a weboldal online is elérhető legyen, szükség volt egy domain név és egy megfelelő tárhely kiválasztására. Választásunk az [InfinityFree](http://www.infinityfree.com)-re esett, mert a legtöbb igényünket ki tudta elégíteni. Az egyetlen dolog, ami nem volt benne az ingyenes csomagba, az az email kiküldés értesítés, vagy megerősítés szempontjából.

Ettől eltekintve végül itt hoztuk létre a [weissesvagyok.infy.uk](http://weissesvagyok.infy.uk) weboldalt, amely az általunk véglegesített verziója a [weissesvagyok.hu](https://weissesvagyok.hu) oldalon is elérhető (lesz).

## Miért InfinityFree?

Fejlesztésünk első szakaszában a Nethely szolgáltatását próbáltuk ki. Azonban már az induláskor adatbázis-kezelési hibákba ütköztünk, és ezek elhárítására nem találtunk működő megoldást. Emiatt praktikusabbnak bizonyult új szolgáltató után nézni, és így esett a választás az InfinityFree-re, amely azóta is stabil működést biztosít.

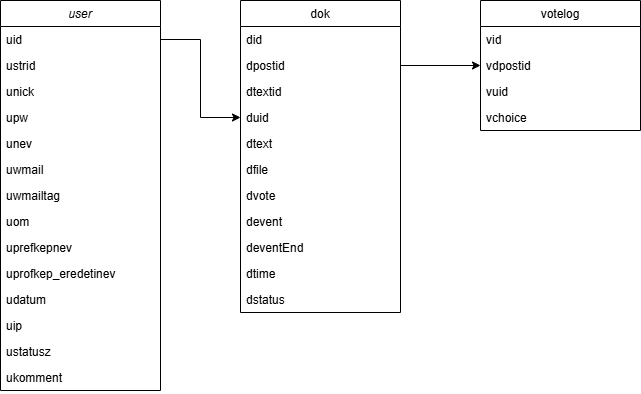
# Fejlesztői dokumentáció: Közösségi média

## Fejlesztéshez használt hardver

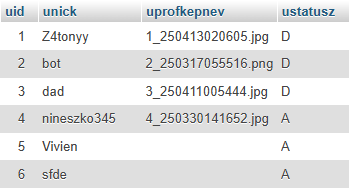
Fejlesztés főként asztali számítógépen

* Processzor: AMD Ryzen 5 7600X3D
* Videókártya: ASUS ROG STRIX GeForce RTX 4070 SUPER
* Memória mérete: 32 GB
* Operációs rendszer: Windows 11, 64 bit

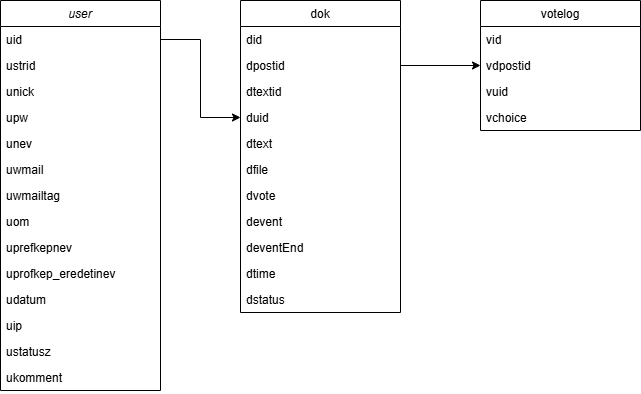
## Adatbázis táblák

A weboldal közösségi funkcióinak alapját képezik azok az adatbázistáblák, amelyek a felhasználók, bejegyzések, szavazások és egyéb kapcsolódó adatok strukturált tárolását teszik lehetővé. Az adatok elrendezése és a mezők kialakítása során fontos szempont volt az átláthatóság, a kompaktság, valamint az, hogy későbbi bővítés esetén is egyszerűen módosíthatók legyenek az összefüggések.

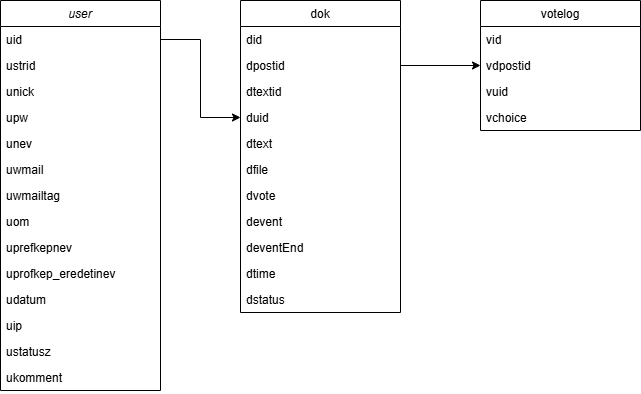
### *’user’ tábla (felhasznált részei)*

* uid – Ez a tábla a felhasználó azonosítója, és a tábla elsődleges kulcsa.
* unick – Ez tartalmazza a felhasználó nevét, amelyet módosíthat is. Minden felhasználó neve legfeljebb 16 karakter hosszú lehet.
* uprofkepnev – A felhasználó profilképének egyedileg elmentett neve. A típusa VARCHAR.
* ustatusz – Ez alapján határozható meg a felhasználó jogosultsága.

### *’dok’ tábla*

* did – Ez a tábla elsődleges kulcsa.
* dpostid – Ez a közzétett bejegyzés azonosítója.
* dtextid – Ez a bejegyzésekhez tartozó komment azonosítója. A 0. azonosítójú „komment” az a bejegyzés önmaga.
* dtext – Ez az adatbázisba feltöltött szöveg.
* dfile – Ez a feltöltött fájlnevek mezője, több esetén   
  ; karakterrel elválasztva.
* dvote – Ezek a szavazati opciók, több esetén   
  ; karakterrel elválasztva
* devent – Ez az eldöntés, hogy kiírásnál feltüntesse-e az események között: Igen („I”) / Nem („N”)
* deventEnd – Ez az esemény lejárati dátuma.   
  „N” devent tulajdonság esetén az értéke NULL.
* dtime – Ez a közzététel ideje.
* dstatus – Ez a bejegyzés állapota Aktív („A”) / Inaktív („I”)

### *’votelog’ tábla*

* vid – Ez a tábla elsődleges kulcsa.
* vdpostid – Ez annak a bejegyzésnek az azonosítója, amelynél a felhasználó éppen szavazott.
* vuid – Ez a szavazott felhasználó azonosítója.
* vchoice – Ez a választott szavazat, szövegként elmentve.

## Algoritmusok a weblapon

A közösségi felület működése során több kulcsfontosságú algoritmus is fut a háttérben. Ezek célja az adatok validálása, a hibák megelőzése, valamint a funkciók gördülékeny működtetése.

### *Jogosultságellenőrzés*

A weboldal biztonsága és személyre szabott működése érdekében minden oldalbetöltés előtt ellenőrizzük a felhasználó jogosultságait:

**Bejelentkezés ellenőrzése**: Ha a felhasználó nem jelentkezett be, átirányítjuk a bejelentkező oldalra.

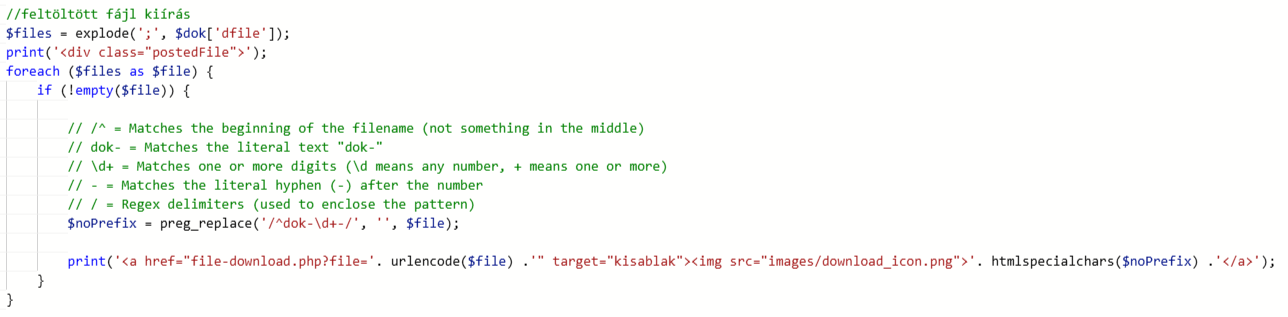
**Osztály beállításának ellenőrzése**: Ha a classid mező nincs kitöltve, a rendszer a profiloldalra navigál.

**DÖK jogosultság ellenőrzése**: A ustatusz = 'D' mező alapján korlátozott funkciókhoz (pl. bejegyzés létrehozás, szavazás indítása) csak DÖK tag férhet hozzá.

**Biztonság**: A jogosultságokat mindig PHP oldalon, szerveroldalon ellenőrizzük, nem csak JavaScript-tel a kliensoldalon.

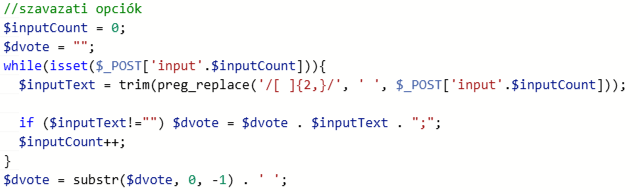
### *Fájlkezelés*

A felhasználók képesek fájlokat csatolni a bejegyzésekhez (nem hozzászólásokhoz).

* **Feltöltés**: A fájlokat a dok-files/ mappába mentjük. A fájlnév ütközések elkerülése érdekében egyedi fájlnevet generálunk, és ezt mentjük az adatbázisba. A zárt hozzáférés végett nincs megszabva fájlformátum, mindent fájl típus engedélyezve van, de csak maximum csak 2 megabájtos fájlméret (fájlonként). 
* **Letöltés**: A fájlokat URL-en keresztül lehet letölteni, egy PHP script közvetíti a lekérést. Ez biztosítja, hogy csak létező és jogosult fájlokat lehessen elérni. 
* **Megjelenítés**: A fájl eredeti neve jelenik meg, nem a generált technikai név. 

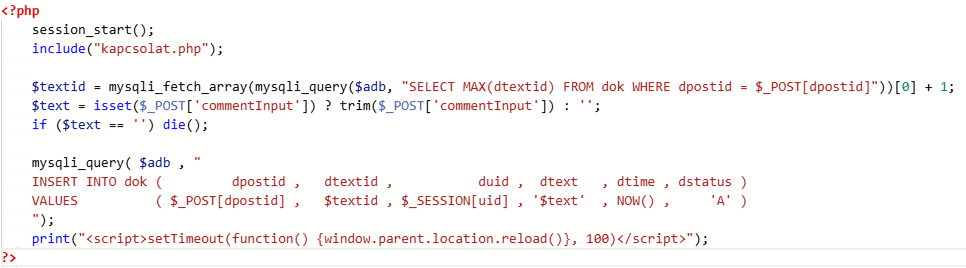
### Szavazási opciók kezelése

A DÖK tagok bejegyzéshez szavazást is mellékelhetnek.

* **Adatrögzítés**: A dvote mező tartalmazza az opciókat ; karakterrel elválasztva. 
* **Megjelenítés**: A szavazási opciókat explode(';') függvénnyel tömbbé alakítjuk, majd a tömböt végig járva, minden elemet rádiógombként jelenítünk meg.
* **Szavazat leadása**: A votelog táblába új rekord kerül, amely tartalmazza a felhasználó azonosítóját, a szavazott bejegyzés dpostid mezőjét, és a választott opció szövegét.
* **Újraszavazás**: Az előző szavazat módosítható, az új érték felülírja a régit. 

### Bejegyzések megkülönböztetése

A rendszer képes kezelni a bejegyzések és hozzászólások hierarchiáját.

* **Bejegyzés**: dtextid = 0, dpostid egyedi (új bejegyzés létrehozásakor a jelenlegi legnagyobb dpostid + 1).
* **Komment**: dtextid > 0, dpostid megegyezik a szülőbejegyzésével. A dtextid értékét szintén a jelenlegi legnagyobb érték + 1 alapján határozzuk meg. 
* **Összekapcsolás**: A rendszer bármilyen bejegyzés- és kommenttípust képes kezelni, akár fájlmelléklettel, akár szavazással együtt.

## Weboldal dizájnja

A felületet próbáltam a lehető legegyszerűbben megoldani, hogy senkinek se akadjon kétsége, amikor ezt az oldalt használja. Az oldalam frontend részét barátaimra hagyatkozva oldottam meg, hisz én épp az egyszerűség miatt az alap formázással és színezéssel is beértem volna. Az oldal mobilbarát kialakítású. Minden dinamikusan jelenik meg, így akár egy kisebb, vagy épp egy nagyobb képernyőn is tökéletesen működik.  
A bejegyzések és kommentek elkülönülnek, a profilkép, név, időbélyeg és mellékletek jól strukturáltan jelennek meg. A szavazások és a fájlok külön blokkokban kapnak helyet, megkönnyítve az áttekintést. Az esemény fül 1210 pixelszélességen túl nem elérhető, megjelenítése meggondolandó. Ez a mérték szándékosan megegyezik az index „mobilnézetével”, az egységes kinézet miatt.

## Különböző körülmények, esetek és hibakezelések

A közösségi média modul működése során számos olyan helyzettel találkozhatunk, amely rendellenes működéshez, hibákhoz vagy jogosultsági problémákhoz vezethet. Ezek megfelelő kezelése elengedhetetlen a stabilitás, biztonság és felhasználói élmény biztosítása érdekében.

A fejlesztés és tesztelés során kiemelt figyelmet fordítottunk a beviteli hibák, a fájlkezelés során fellépő problémák, a jogosultsági ellenőrzések, a szavazások integritása, valamint az adatbázis-kezelés biztonságos működésének biztosítására.

### *Interakció következtében megjelenő hibakezelés*

Jó pár interakció következtében az oldal frissülni kényszerül. Ennek az az oka, hogy a felhasználó ne csak a saját maga által éppen kitett posztot lássa, hanem esetleg új megjelenésű bejegyzéseket, hozzászólásokat, vagy akár a szavazók számát.  
A beviteli hibák kezelése során a rendszer ellenőrzi, hogy a felhasználó nem hagyta-e üresen a kötelező mezőket. Például egy új bejegyzés vagy hozzászólás beküldése csak akkor engedélyezett, ha a szövegmező nem üres. Ezen kívül (jelenleg csak szavazási opciók megadásánál) bizonyos helyeken meg van határozva maximális karakterkorlát.   
A szavazások esetében a rendszer megköveteli, hogy legalább két opció kerüljön megadásra, ellenkező esetben figyelmeztető üzenet jelenik meg, és a szavazás mentése meghiúsul.  
A fájlkezelésnél hiba esetén a közzététel gomb megnyomása után felszólít, hogy a 2MB-nál nagyobb méretű fájlt, aminek a sorszámát ki is írja a felugró párbeszéd ablakban, hogy távolítsa el a fájl neve mellett megtalálható piros X megnyomásával.

### Háttérben folyó hibakezelés

Jogosultsági szempontból a rendszer már az oldal betöltésekor ellenőrzi, hogy a felhasználó be van-e jelentkezve. Amennyiben ez nem teljesül, automatikusan a bejelentkező felületre irányítja a látogatót. A regisztrált felhasználóknak is meg kell felelniük bizonyos feltételeknek: például amennyiben a profiljukhoz nincs hozzárendelve osztály. Ilyen esetben a rendszer figyelmeztet, majd a profiloldalra irányítja őket.  
Minden adatbázisból történő kiíratásnál el lett végezve a HTML tagek elleni védelem. Magyarán, ha akármelyik beviteli mezőbe például egy <button> szócska kerül, akkor az kiírva nem gombként, hanem nyers szövegként jelenik.   
A túlzott adatbázis-terhelést megelőzve kompakt adatbázis lett a cél, illetve tényleges log helyett csak update megy végbe a táblán belül, nem pedig create.  
Az egyedi azonosítók, például a dpostid és dtextid generálásakor a rendszer lekérdezi az aktuális legnagyobb értéket, és azt követően határozza meg az új értéket. Ez a megközelítés megbízható módon akadályozza meg az azonosítóütközéseket, különösen párhuzamos felhasználás esetén, ami a jelenlegi esetben hatalmas befolyásoló tényező.

## Fejlesztési lehetőségek

A jelenlegi közösségi média modul számos funkcióval rendelkezik, amelyek lehetővé teszik a felhasználók közötti kommunikációt, a tartalmak megosztását, szavazások lebonyolítását, valamint fájlok kezelését. Ugyanakkor a fejlesztés során számos olyan területet azonosítottunk, ahol a jövőben tovább lehetne bővíteni vagy finomítani a rendszert, a funkcionalitás növelése, a biztonság erősítése és a felhasználói élmény javítása érdekében.

Az egyik legfontosabb fejlesztési irány a valósidejű funkciók bevezetése. Jelenleg a rendszer csak akkor frissíti a tartalmakat, ha a felhasználó manuálisan újratölti az oldalt, vagy bizonyos műveletek automatikusan újratöltést váltanak ki. Ezt kiválthatná egy WebSocket-alapú megoldás, amely lehetővé tenné, hogy az új hozzászólások, szavazatok vagy fájlfeltöltések automatikusan megjelenjenek a felhasználók felületén. Továbbá fontolóra lehetne venni a bejegyzésekhez ugrást csak egy URL ki- és bemásolásával, ezzel is lehetőséget adni egyes bejegyzések továbbítására. Ezekkel az oldal sokkal dinamikusabbá válna, és jobban megfelelne a mai felhasználói elvárásoknak.  
Végül, érdemes lenne megvizsgálni egy mobilalkalmazás fejlesztésének lehetőségét is, amely a webes felület funkcionalitását natív formában kínálná Android és iOS eszközökön. Ez nemcsak kényelmesebb elérést biztosítana, hanem lehetőséget nyújtana olyan eszközspecifikus funkciók kihasználására is, mint az értesítések, kamera használata fájlcsatoláshoz vagy offline tartalmak mentése.

# Fejlesztői dokumentáció: Csevegő felület

## Különbségek már létező programoktól

A Weissesvagyok.hu csevegő felülete elvi működésben nem sokban tér el a különféle már létező élő csevegésre alkalmas programoktól. Leginkább a felhasznált technológiák azon jellemzők melyek elválasztják a többi hasonló programtól. A fejlesztés során a magasabb szintű technológiák helyett, mint például WebSocket a csevegő felülethez folyamatosan újra frissítő JavaScript logikát használtunk mivel magasabb szintű technológiák esetében tudásunk nem érte el a megfelelő szintet a minőségi munka biztosításához. A program emellett különbözik más populáris csevegő felületektől hiszen ezen program a Weissesvagyok.hu oldalán fut biztosítva az ott kialakuló közösségépítést és könnyű mozgást az oldal többi felületére.

## Fejlesztői környezet

A fejlesztés során többféle technológiát alkalmaztunk, elsősorban a XAMPP nevezetű program csomagot. A XAMPP program csomag tartalmazza az Apache helyi webszerver szolgáltatót, a MySQL adatbáziskezelőt és a PHP támogatáshoz szükséges PhpMyAdmin adminisztrációs felületet. A XAMPP és a benne lévő programok lehetővé tették számunka a lokális azaz a saját számítógépünkön való fejlesztést és tesztelést anélkül hogy szükségünk lett volna egy valós webkiszolgálóra. A kódbázis lefejlesztésére a Visual Studio Code nevezető kódszerkesztőt használtuk. Ezen választás mögött a kódszerkesztő széles, a fejlesztést elősegítő és meggyorsító eszköztára áll. Eszközök mint például automatikus kód kiegészítés/ajánlás, naprakész frissítések, beépített git elérhetőség, fejlesztést megkönnyítő felhasználói felület, és még egyéb kiváló eszközök.

## Felhasznált nyelvek

A program lefejlesztéséhez különféle nyelveket használtunk. A kódbázis legalapvetőbb nyelve a PHP. A hagyományos HTML-el ellentétben a PHP lehetővé tette számunkra a szerver oldali programozást mely rettentő fontos szempont hiszen az oldal felhasználói adatokat kezel. A PHP segítségével a HTML elemek fejlesztése is leszűkült hiszen a PHP dinamikusan generál HTML kódot melyet a kliens oldal azaz a felhasználó kap meg, azonban minden fontos háttérfolyamat a szerveren történik. A PHP mellett természetesen a HTML és CSS is nagy szerepet kap ott ahol fontos háttérfolyamat nem keletkezik. Ezen nyelveken kívül számos elem tartalmaz JavaScript logikát. A JavaScript a programunk esetében olyan, dinamikus vagy látványelemhez köthető folyamatokat intéz mely folyamatok lefuthatnak a kliens oldalon is hiszen különösen védett folyamatot nem végeznek.

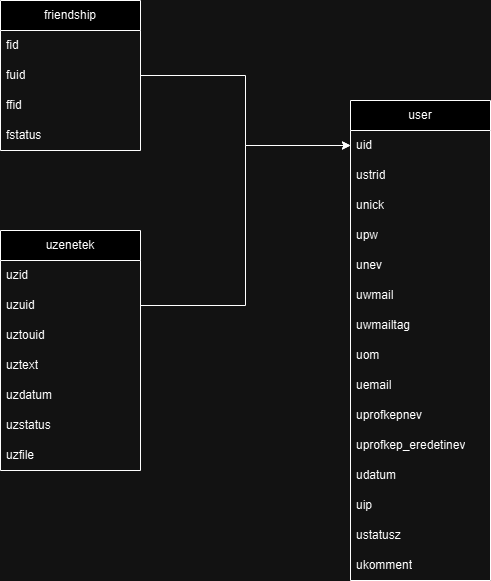
## Kiegészítő elemek

Megemlítésre méltó az is hogy a XAMPP által nyújtott programokat miért és miféleképpen használtuk fel. Az Apache szerver lényegében egy valós webszerver emulálásában segített, különösebb kifejtés nem szükséges. A MySQL azonban már érdekesebb, a MySQL mely egy adatbáziskezelő rendszer a weboldalon tárolandó adatok hatékony tárolására használtuk, választásának oka az is hogy egy, az évek során sokat tesztelt adatbázisról beszélünk így megbízható is. A PhpMyAdmin az adatbázis adminisztrációjára használtuk, ezen program a PHP támogatásának köszönheti a felőlünk érkező kiválasztást, ezen programmal hatékonyan kezelhettük az adatbázisba kerülő adatokat illetve az ott lévő adatok felhasználására is tökéletes lehetőséget és környezetet adott.

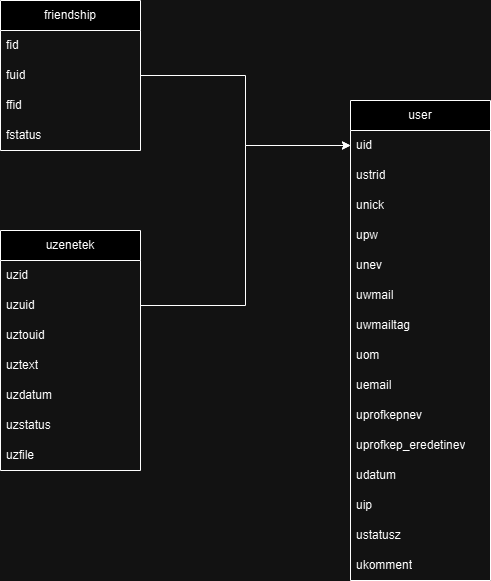
## A fejlesztéshez használt hardver

A fejlesztés során meghatározó többségben az iskolában elérhető, iskolai számítógépeket használtuk, megállapítható tehát hogy a fejlesztés nem igényelt különleges erőforrásokat. Bizonyos esetekben az otthon elérhető számítógépeinket is használtuk azonban a helyzet változatlan.

## Adatszerkezet

A weboldal adatbázisa számtalan táblát tartalmaz azonban a csevegő felület nem használja mindet, a felhasznált táblákat és azok összekapcsolását az alábbi képen vehetjük szemügyre:

### “friendship” tábla



A friendship (barátság) tábla követi nyomon hogy melyik felhasználó barátja melyik felhasználóknak. Amikor egy felhasználó hozzáad egy másik felhasználót a csevegő listájához akkor két új sor kerül a friendship táblába.

#### Fid:

Az fid a friendship tábla elsődleges kulcsa. Ebből fakadóan az elsődleges kulcs automatikusan generálódik.

#### Fuid:

Az fuid tárolja azt az id-t mely id (azonosító) a küldő félhez tartozik, azonban a az adatszerkezet kialakítása végett egy küldés során a fordítottja is bekerül egy új sorba. Nem mellesleg az fuid idegenkulcsként is szolgál

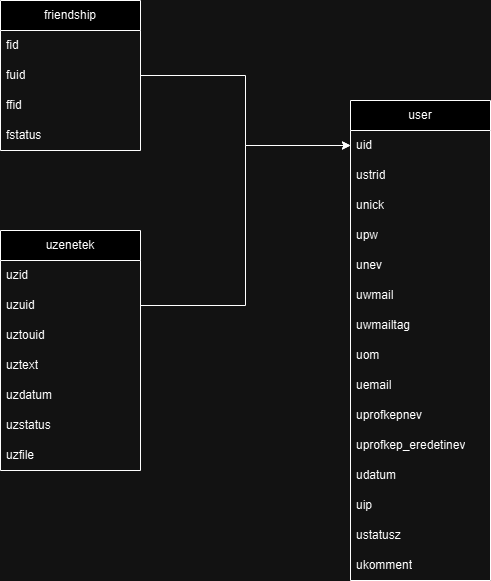
#### Ffid:

Az ffid azt az id-t tartalmazza mely id a fogadó félhez tartozik, azonban a az adatszerkezet kialakítása végett egy küldés során a fordítottja is bekerül egy új sorba.

#### Fstatus:

Az fstatus a két fél státuszát jelöli azaz P (pending, várakozó), ez a státusz addig marad fenn míg a másik fél (fogadó) nem küldött üzenetet vissza. Ha a fogadó fél is küldött vissza üzenetet a státusz A-ra (accepted, elfogadott) státuszra változik. A felhasználók letilthatják egymást, ebben az esetben a státusz B-re (blocked, blokkolt) státuszra változik ekkor a két fél nem képes egymásnak üzenetet küldeni. Fontos hogy az alapján hogy ki tiltott le kit, az fstatus különböző lesz, a letiltó fél ugyanis nem B hanem BB (blocked by, ...által letiltott) státuszt fog kapni. Ezzel biztosítva van hogy csak a letiltó fél legyen képes a tiltást feloldani.

### “uzenetek” tábla



Az “uzenetek” tábla minden, az üzenetekhez kapcsolódó adatot tartalmaz.

#### uzid:

Az uzid az üzenetek tábla elsődleges kulcsa. Ebből fakadóan az elsődleges kulcs automatikusan generálódik.

#### uzuid:

Az uzuid tárolja azt az id-t mely id (azonosító) az üzenetet küldő félhez tartozik. Idegenkulcsként is szolgál.

#### uztouid:

Az uztouid azt az id-t tartalmazza mely id az üzenetet fogadó félhez tartozik.

#### uztext:

Az uztext az elküldött üzenet szöveges tartalmát tárolja.

#### uzdatum:

Az uzdatum az üzenet küldésének pontos időtartamát tárolja (év-hónap-nap-óra-perc-másodperc).

#### uzstatus:

Az uzstatus az üzenet státuszát jelzi mely lehet S (sent, elküldött) vagy D (deleted, törőlt).

#### uztext:

Az uzfile az elküldött fájlt tárolja.

## Algoritmusok

### Csevegések megjelenítése:

Ezen algoritmus a megfelelő csevegések megjelenítéséért felelős. Ez egy dinamikusan bővülő lista mely lekéri az adatbázisból azokat a felhasználókat akik között már létezik csevegés és megjeleníti azt a megfelelő felhasználóknak.



A dinamikusan legenerált gombokra kattintva a megfelelő csevegést nyithatjuk meg és innentől már lehet üzenetet küldeni a következő képeken a dinamikus felület generálást és backend logikát láthatjuk:





### Barát hozzáadása

A barát hozzáadása gombra nyomva egy másik panelre irányít át a program ahol az összes az adatbázisban szereplő felhasználót láthatjuk. Nevükre rákattintva hozzáadhatjuk őket mint barát és onnantól kezdve képesek vagyunk üzenetet küldeni illetve ők is képesek ugyan erre. Amikor egy falhasználó nevére rákattintunk amellett hogy beszélgetésbe kezdhetünk velük az ő nevük már nem lesz benne a barát hozzáadása listában hiszen az már megtörtént. Az alábbi képen látható a dinamikusan generált gombok melyek a felhasználókat tartalmazzák:



Az alábbi képen pedig a kattintást követő esemény látható:



### Üzenet küldése

Miután a barát hozzáadása megtörtént mindkét fél képes egymásnak üzenetet küldeni. Amikor a felhasználó rákattint egy már a barátlistában lévő felhasználó nevére a csevegő felület megnyílik, legenerálódik. Ezután a felhasználó az ott található szöveges mezőbe írhatja üzenetét majd a küldés gombra vagy az enter billentyű (számítógépen) lenyomására elküldheti üzenetét. Ezen backend logikát láthatjuk a következő képen:



# Fejlődési lehetőség

## Merre tovább?

A csevegő felület tömérdek fejlődésen mehet keresztül. Amellett, hogy egy program normális esetben sosem készül el teljesen vannak olyan pontok, ahol ezen csevegő program erőteljesen képes a fejlődésre, illetve mely területeken érdemes mindenképpen fejlődést előidézni.

### Működési alapelv

A program jelenleg JavaScript ablak frissítéssel tölti be a beérkező üzeneteket mely így folyamatos időnkét enyhén irritáló villogást is okozhat, habár ez böngészőtől is függhet, illetve az idő nyomós részében nem különösebben túl irritáló. Azonban annak érdekében, hogy a UX (User Experience / felhasználói élmény) kényelmesebb és hatékonyabb legyen, az olyan technológiák mint az AJAX vagy WebSocket implementálása elengedhetetlen.

### Csoportok kialakítása

A fejlesztés során a csoportok kialakítása bizonyos tényezők miatt (melyek között külsős, a fejlesztők hatáskörén kívül eső problémák is felmerültek) a csoportok kialakítása nem került lefejlesztésre. A számunkra elérhető adatok helyenként hiányosnak bizonyultak ebből fakadóan a csoportok implementálása egy jó fejlesztési lehetőség amint egy megfelelőbb adatrendszer ki tud alakulni mely számunkra még nem volt elérhető.

## Teljes modernizálás

Az oldal PHP-ra épült mely bár egy jól ismert és stabil nyelv az évek során már-már elavultnak számít. Manapság már a Node.js segítségével például lehetőségünk nyílik szerver oldalon programozni JavaScriptben mely így egy gyorsabb élményt biztosít. Elsősorban, a továbbfejlesztés elérhető azzal is, hogy a PHP nyelvnél maradva az oldalt továbbfejlesztjük valamiféle PHP keretrendszerrel (pl Laravel) ebben az esetben egy összeszedetebb és erősebb programot kapunk. Azonban, ha ténylegesen modernizálni szeretnénk JavaScript alapú keretrendszerek hatékonyabb fejlesztési lehetőségnek számítanak. Ezen esetben akár használhatunk React-ot, Vue-t vagy egyéb keretrendszert de akár a React-native segítségével még egy szintet léphetünk és egyenesen fejleszthetünk webre és mobiltelefonra azonban így már a mobiltelefon verzió már applikációként fog futni így még hatékonyabb mobil élményt adhatunk felhasználóinknak.

# Tesztelői hozzáférés

A teszteléshez szükségszerűen egy a megfelelő engedélyekkel rendelkező felhasználót hoztunk létre, ennek adatait itt találhatjuk, ezen felhasználó segítségével a funkciók mindegyike kipróbálásra kerülhet.

## Felhasználónév: Tesztfelhasznalo

## Jelszó: teszt12345

# Felhasználói dokumentáció - csevegőfelület

## Mi is ez a program?

Ez a program egy csevegőfelület a Weissesvagyok.hu oldalán. Tehát egy olyan programról beszélünk melyen keresztül üzeneteket küldhetünk más felhasználóknak. Az egyértelmű üzenetküldés mellett még egy elengedhetetlen funkció a barát felvétele.

## Program használatához szükséges hardver:

Bármilyen, internettel rendelkező eszköz. Természetesen a mobileszközök is bele tartoznak ebbe.

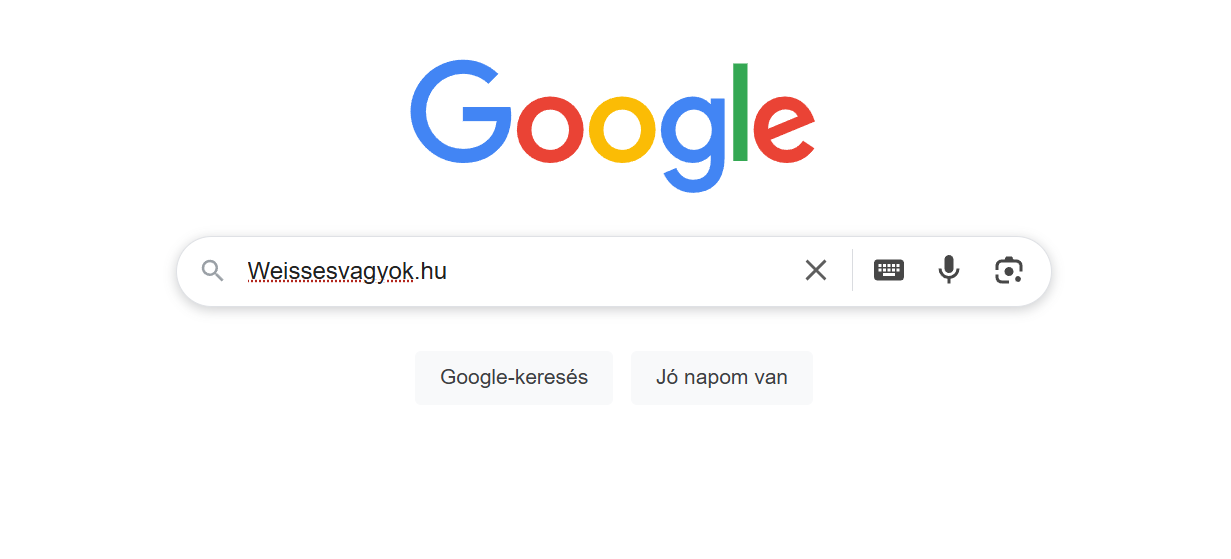
## Program használatához szükséges szoftver:

Bármilyen modern böngésző.

# Program elindítása

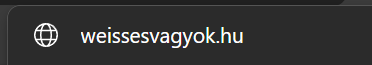
A program elindításához kell egy böngésző és a böngészőn belül két féle képen érhetjük el a programot (weboldalt).

## Egyszerű keresés:

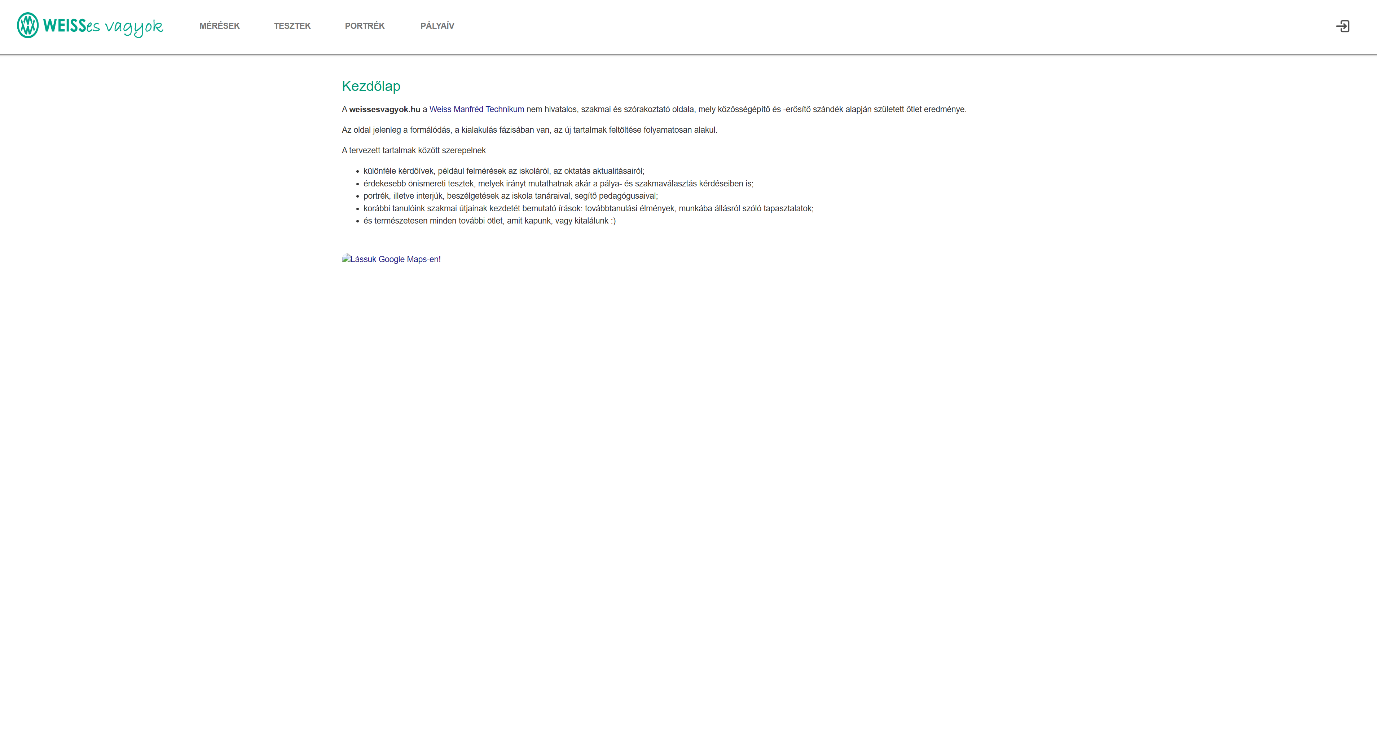


## 

## Keresés URL alapján:



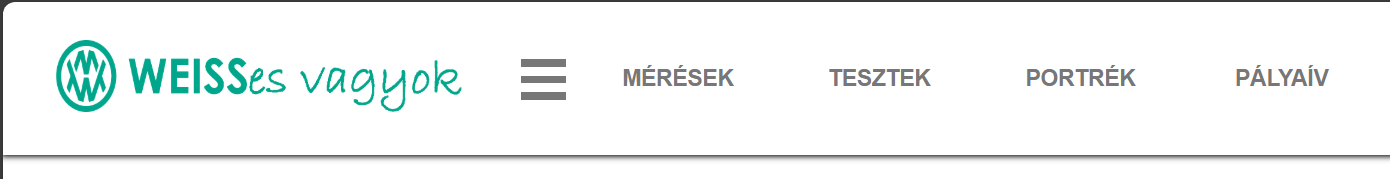
## Az oldalra érkezés után:



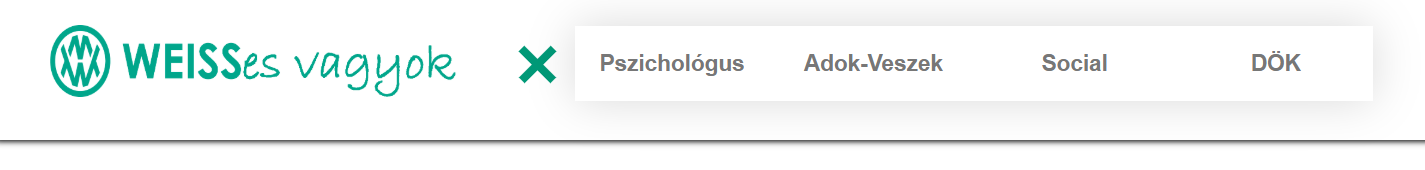


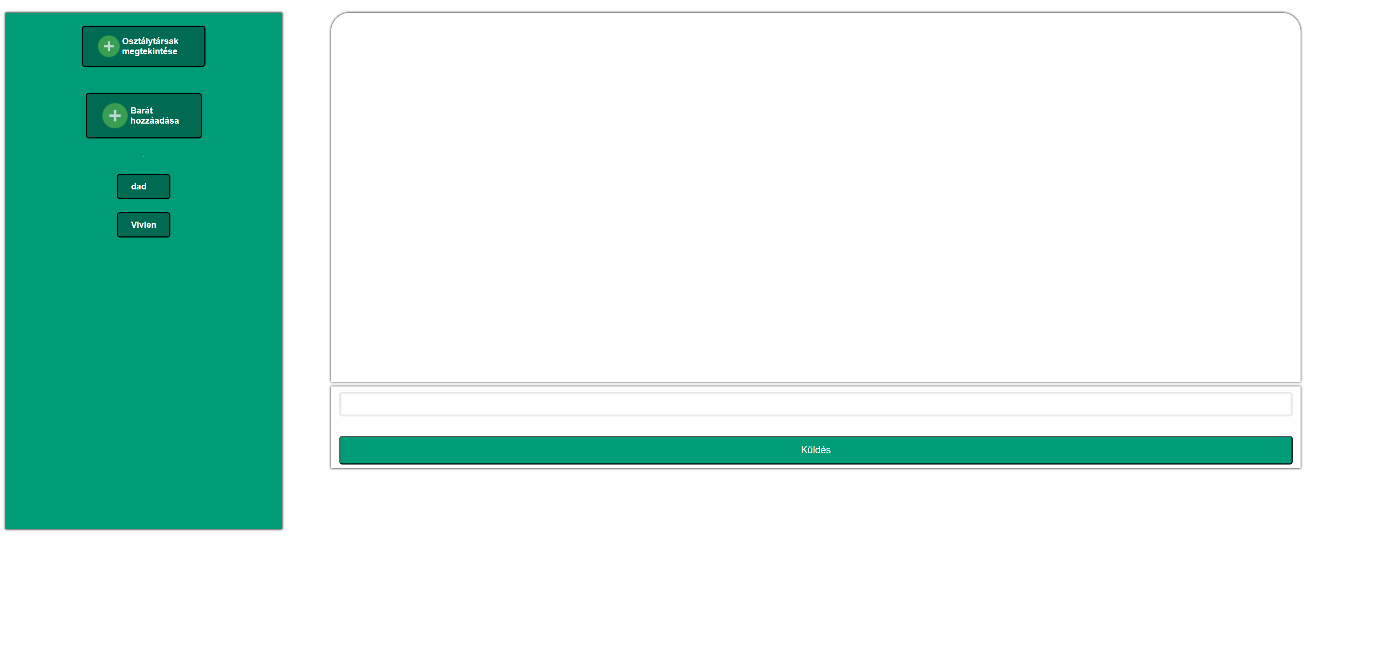
A bejelentkezés gombra kattintva beléphetünk vagy létrehozhatunk egy felhasználót.

## Bejelentkezés után:



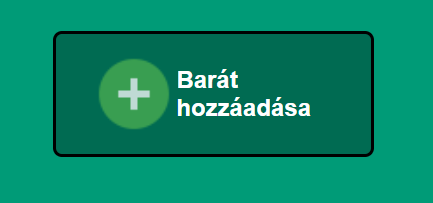
Bejelentkezés után a menüsor kiegészül egy lenyíló elemmel melyet ha megnyitunk különböző kiegészítő tartalmakat láthatunk, a “social” menüpontot kiválasztva érkezhetünk meg a csevegőfelületre.

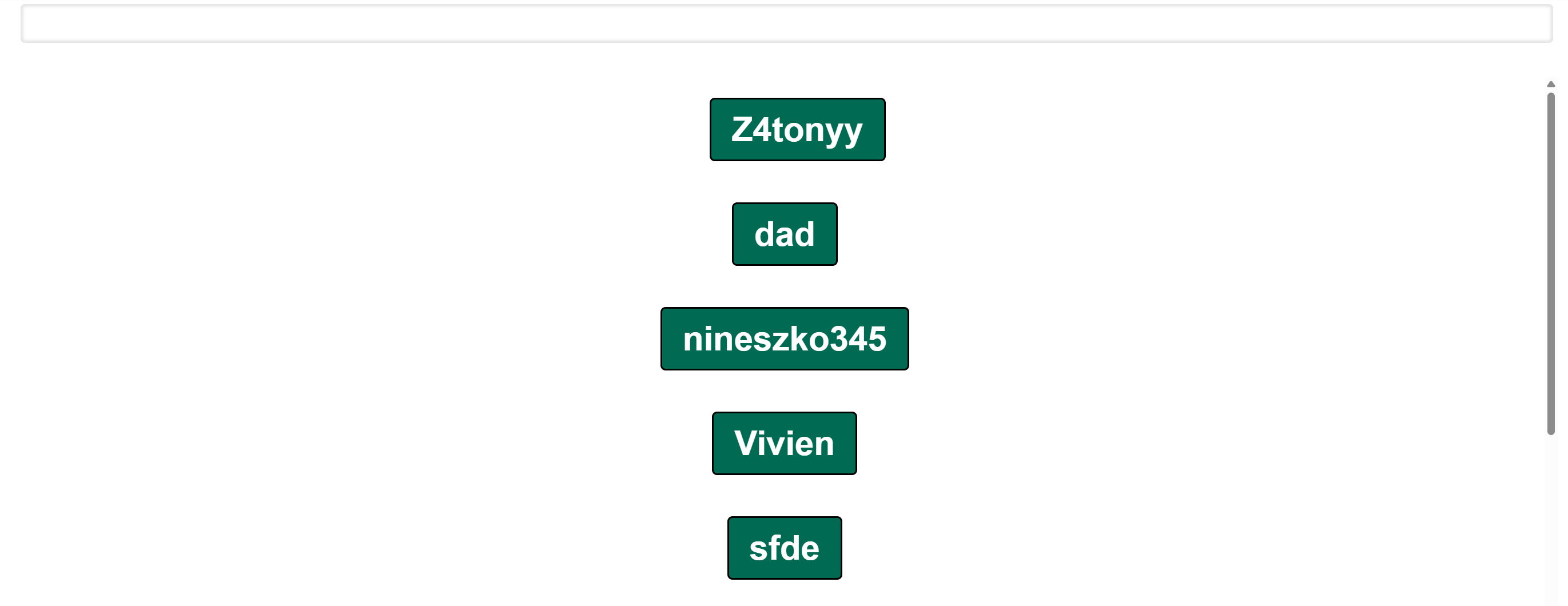




# Program használata

Miután beléptünk és a csevegőfelületre navigáltunk alapértelmezetten (természetesen ha új felhasználók vagyunk) nincs egyetlen barátunk sem, ekkor a barát hozzáadása gombra kattintva tekinthetjük meg az oldalon elérhető felhasználókat.

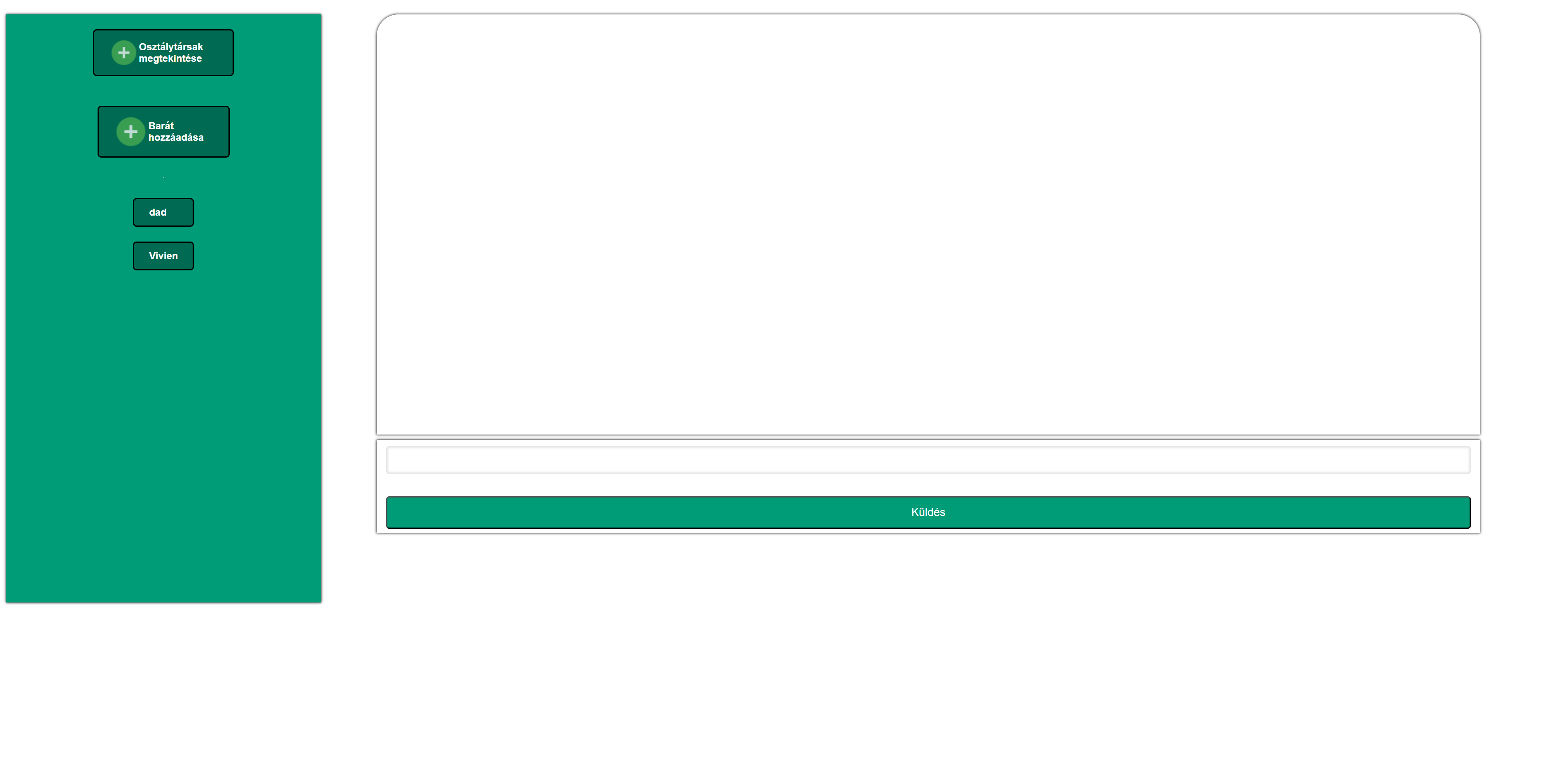


Valamelyik névre kattintva hozzáadhatjuk az adott felhasználót barátlistánkhoz mellyel kezdhetjük is üzenetünk elküldését.

## A barát hozzáadás után:

A beszélgetések gombra kattintva vissza léphetünk beszélgetéseinkhez.



Valamelyik névre kattintva megnyithatjuk a beszélgetést, az üzenetek a nagy ablakban fognak megjelenni.

Az alatta lévő terület pedig az üzenetünké, az az alatt lévő gomb pedig az üzenet elküldésére szolgál, bár az üzenet elküldése az enter billentyű lenyomásával is megtehető.



# Munkám értékelése

## Értékelés:

A munkám értékelése során, figyelembe vettem a befektetett munkát, időt és az ezek szemléletében való megvalósult munka eredményét. Az eltervezett funkciók mintegy 80%-a biztosan megvalósult, esetleges kimaradások, hiányosságok vagy hibák fellelhetőek azonban ezek nem olyan dolgok melyek gyökeresen befolyásolnák a program működését vagy a projekt befejezte után ne lehetne őket megoldani. A munkámat elégségesnek értékelem, a helyzethez és legfőképpen képességeimhez mérten megpróbáltam a lehető legtöbbet kihozni a programból.

## Program hasznosulása:

Személy szerint látok potenciált a csevegőfelületben a továbbiakban. A jelenlegi állapottól a program hatalmas fejlődési lehetőséget rejteget magában így a további fejlesztések sokkal vonzóbbá válhat a diákok számára és mivel a Weissesvagyok.hu oldal is sok vonzó dologgal bővül majd a jövőben a csevegőfelület is egyre nagyobb hangsúlyt kaphat.