** **

**ZÁRÓDOLGOZAT**

Készítették:

Lengyel Bálint

Kelemen Ádám

Juhász Balázs

Konzulens:

Farkas Zoltán

Miskolc

2024.

Miskolci SZC Kandó Kálmán Informatikai Technikum

Miskolci Szakképzési Centrum

SZOFTVERFEJLESZTŐ- ÉS TESZTELŐ SZAK

ZÁRÓDOLGOZAT

**Connect2Gether (Közösségi média) felület**

Lengyel Bálint - Kelemen Ádám - Juhász Balázs

2023-2024

Tartalom

[**A PROJEKTRŐL** 5](#_Toc165274790)

[**FELHASZNÁLT TECHNOLÓGIÁK ÉS PROGRAMOZÁSI NYELVEK** 6](#_Toc165274791)

[**FRONTEND** 6](#_Toc165274792)

[**HTML:** 6](#_Toc165274793)

[**CSS:** 6](#_Toc165274794)

[**React:** 7](#_Toc165274795)

[**Bootstrap:** 7](#_Toc165274796)

[**BACKEND** 8](#_Toc165274797)

[**Asp.NET Core 8.0 Web API:** 8](#_Toc165274798)

[**Entity Framework Core:** 9](#_Toc165274799)

[**XAMPP:** 9](#_Toc165274800)

[**MYSQL:** 9](#_Toc165274801)

[**JWT Token:** 10](#_Toc165274802)

[**FELHASZNÁLT PROGRAMOK** 11](#_Toc165274803)

[**Visual Studio 2022:** 11](#_Toc165274804)

[**Visual Studio Code:** 11](#_Toc165274805)

[**KOMMUNIKÁCIÓS FELÜLETEK** 12](#_Toc165274806)

[**Trello:** 12](#_Toc165274807)

[**Github:** 12](#_Toc165274808)

[**Discord:** 12](#_Toc165274809)

[**ADATBÁZIS SZERKEZET** 13](#_Toc165274810)

[**WEBOLDAL BEMUTATÁSA** 16](#_Toc165274811)

[**Főoldal:** 16](#_Toc165274812)

[**Regisztráció:** 16](#_Toc165274813)

[**Bejelentkezés:** 17](#_Toc165274814)

[**Felhasználói oldal:** 18](#_Toc165274815)

[**Publikus profil oldal:** 18](#_Toc165274816)

[**Post oldal:** 19](#_Toc165274817)

[**Profil oldal:** 19](#_Toc165274818)

# **A PROJEKTRŐL**

A csoportunk egy közösségi média felületet készített webes és egy hozzátartozó WPF-es grafikus oldallal.

Sokat agyaltunk azon, hogy milyen projektet készítsünk, de végül azért döntöttünk e mellett, mivel kellően egyedinek éreztük ezt a témakört és izgalmasnak tartottuk, hogy egy konkrét közösségi média felületet készíthetünk. A mai világban szinte mindenki használ ilyen oldalakat, vegyük csak például a Facebook-ot, Twittert vagy akár az Instagramot. Mindegyikben az a közös, hogy az emberek megtudják osztani tartalmaikat a nyilvánossággal. Többféle formájuk létezik, mint például a szöveges tartalmak, a képek vagy a videók publikálása. Mi ezeken belül a szövegeset céloztuk meg. A felhasználó létrehoz egy postot és erre a többi felhasználó kommentelni tud. A projekt részleteit ebben a dokumentációban fogjuk elmagyarázni, hogy mindenkinek egyértelmű legyen a működése és részletes betekintést engedjünk a projekt fejlesztésében.

# **FELHASZNÁLT TECHNOLÓGIÁK ÉS PROGRAMOZÁSI NYELVEK**

## **FRONTEND**

### **HTML:**

A HTML egy jelölő nyelv, kódnyelv, a weboldalak készítéséhez fejlesztették. Maga a rövidítés, az angol HyperText Markup Language (hiperszöveges jelölőnyelv) kifejezésből alakult mozaikszó.

A HTML a weboldalak létrehozásának hiper szöveges jelölő nyelve, amely leírja a weboldalunk felépítését. A hipertext, a világhálón lévő dokumentumok sokasága, amelyben képek, szövegek, animációk, videók, hanganyagok találhatóak.

A HTML formáját tekintve egy sorozat, de több elemet tartalmaz. Ezek az elemek mutatják meg a böngészőnek, a tartalom megjelenítését, egy dokumentumban, weboldalon: pl. dőlt betűvel, vastagon szedve stb. Mögötte egy-egy jelölés áll, amely nélkül csak formátlan, ömlesztett szöveg lenne.

### **CSS:**

A CSS az angol (Cascading style sheets) kifejezés rövidítése, ami magyarul „egymásba ágyazott stíluslapokat” jelent. Míg a HTML a weblap szerkezetét határozza meg (főcímek, bekezdések, stb.), és lehetővé teszi, hogy különböző elemeket ágyazz webes dokumentumban, addig a CSS a weblap vizuális stílusáért felel. Az oldal elrendezéséért, a színekért, a betőkészletekért, azok méretéért, és így tovább.

### **React:**

A React.js, gyakran egyszerűen csak React-nak nevezik, egy Facebook által fejlesztett nyílt forráskódú JavaScript könyvtár, amely dinamikus felhasználói felületek létrehozására szolgál. Lehetővé teszi a fejlesztők számára, hogy olyan webalkalmazásokat hozzanak létre, amelyek hatékonyan frissíthetőek és renderelhetőek az adatváltozásokra válaszul, így ideális az egyoldalas alkalmazások fejlesztéséhez, ahol a sima felhasználói élmény kiemelt szerepet játszik.

A React komponens alapú architektúrájú ösztönzi a használható felhasználói felület komponensek létrehozását, ami javítja a kód konzisztenciáját és karbantarthatóságát.

A React.js az egyszerűsége, rugalmassága és skálázhatósága miatt a frontend fejlesztés egyik legnépszerűbb JavaScript könyvtárává vált. A képessége, hogy javítsa a webalkalmazások fejlesztési folyamatát és teljesítményét, a világ minden tájáról fejlesztők számára az egyik legjobb választássá tette.

### **Bootstrap:**

A Bootstrap egy nyílt forráskódú keretrendszer (framework), mely HTML, CSS, JavaScript technológiákat használ. Alapvetően arra jó, hogy nagyon könnyedén, és minimális energia befektetéssel tudjon valaki jól kinéző, bármilyen képernyőméreten szépen megjelenő weboldalakat készíteni.

A Bootstrap előnye, hogy a legtöbb „problémára”, mellyel egy fejlesztő találkozik weboldalak arculatának kialakításakor, már egy előre megoldást kínák, ami vagy teljesen megfelel az adott célra, vagy pedig minimális átalakítással.

**Axios:**

Az Axios egy egyszerű ígéret alapú HTTP-kliens a böngésző és a node,js számára. Az Axios egy egyszerűen használható könyvtárat biztosít egy kis csomagban, nagyon bővíthető felülettel.

## **BACKEND**

### **Asp.NET Core 8.0 Web API:**

Az Asp.NET Core 8.0 Web API egy keretrendszer a .NET fejlesztők számára, amely lehetővé teszi a hatékony és korszerű webes alkalmazások fejlesztését. Az API-k adatokat szolgáltatnak más alkalmazások, szolgáltatások vagy felhasználói felületek számára, például mobil alkalmazások vagy weboldalak számára.

Lehetővé teszi a fejlesztők számára, hogy egyszerűen és hatékonyan fejlesszenek HTTP API-kat. A keretrendszer tartalmazza azokat az alapvető komponenseket, amelyek szükségesek a Web API-k fejlesztéséhez, például a HTTP kezelést, a routolást és a modell validációt.

A keretrendszer lehetővé teszi a fejlesztők számára, hogy különféle hitelesítési és azonosítási módszereket alkalmazzanak, például a JWT-t. További funkciói közé tartozik a middleware-k használata, amely lehetővé teszi a fejlesztők számáraaz alkalmazás különböző szintű beállításait és konfigurációit.

### **Entity Framework Core:**

Az Entity Framework Core egy nyílt forráskódú, könnyű súlyú, cross-platform ORM (Object-Relational Mapping) keretrendszer. Lehetővé teszi az alkalmazás fejlesztők számára, hogy a .NET alkalmazásaikat objektum-orientált módon építsék fel.

Az Entity Framework Core támogatja a különböző adatbázis-motorokat, például az SQL Server és a MySQL. Lehetőség van az adatbázisokkal való kommunikációra a LINQ (Language Integrated Query) nyelv használatával.

Könnyen integrálható az ASP.NET Core alkalmazásokban, és számos szolgáltatást és lehetőséget kínál az adatbázis-kezeléssel kapcsolatban, amelyek nagyban segítik az alkalmazás fejlesztőket az adatbázisok hatékony és biztonságos kezelésében.

### **XAMPP:**

A XAMPP egy ingyenes és nyílt forráskódú szoftvercsomag, amelyet webfejlesztők használnak webkiszolgálók létrehozásához és üzemeltetéséhez.

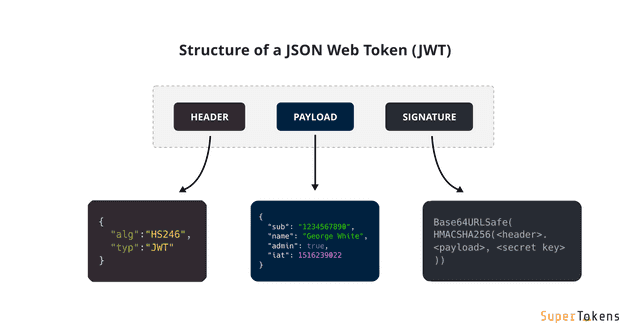
### **MYSQL:**

A MySQL egy nyílt forráskódú relációs adatbázis-kezelő rendszer, amely lehetővé teszi az adatok hatékony kezelését és tárolását. Egyik előnye az adatbázis-skálázhatóság, ami azt jelenti, hogy a rendszer képes megbírkózni nagy mennyiségű adattal és sok egyidejű felhasználóval.

Az egyik legnépszerűbb relációs adatbázis-kezelő rendszer a világon, és széles körben használják a különféle alkalmazásokban, például webalkalmazásokban, adatbázisokban, szoftverekben és még sok másban. Az adatok lekérdezése SQL nyelven keresztül történik, amely az adatbázis-kezelő rendszerek leggyakoribb nyelve.

### **JWT Token:**

A JSON Web Token (JWT) egy nyitott szabvány, amely meghatározza a felek közötti információk biztonágos továbbításának kompakt és önálló módját JSON objektumként. Ez az információ ellenőrizhető és megbízható, mivel digitálisan alávanírva. A JWT-ket titok felhasználásával lehet aláírni vagy nyilvános / magán kulcspár felhasználásával RSA vagy ECDSA.

Kompakt formájában a JSON Web Tokens három részből áll: első a fejléc (header) jellemzően két részből áll: a token típusa, amely JWT, és az alkalmazott aláíró algoritmus, például HMAC SHA256 vagy RSA. A második a tartalom (payload) ez tartalmazza a követeléseket ezek egy entitásra (általában felhasználóra) vonatkozó állítások és kiegészítő adatok, háromféle követelés létezik: bejegyzett, nyilvános és magán igények. A harmadik pedig az aláírás (signature), annak ellenőrzésére szolgál, hogy az üzenet nem változott-e az út során, és magánkulccsal aláírt tokenek esetében, azt is ellenőrizheti, hogy a JWT feladója az, aki azt mondja.

1. ábra - JWT Token működése

# **FELHASZNÁLT PROGRAMOK**

## **Visual Studio 2022:**

A Visual Studio egy hatékony fejlesztői eszköz, amellyel a teljes fejlesztési ciklust egy helyen végezheti el. Ez egy átfogó integrált fejlesztői környezet (IDE), amellyel kódot írhat, szerkeszthet, hibátkereshet és kódot építhet, majd telepíheti az alkalmazást.

A kódszerkesztésen és hibakeresésen túl a Visual Studio fordítókat, kódkiegészítő eszközöket, forrásvezérlést, bővítményeket és még sok más funkciót tartalmaz a szoftverfejlesztési folyamat minden szakaszának javításához.

A Visual Studio számos funkciójának és nyelvi támogatásának köszönhetően az első „Hello World” program megírásától az alkalmazások fejlesztéséig és üzembe helyezéséig fejlődhet.

## **Visual Studio Code:**

A Visual Studio Code (röviden VS Code vagy Code) a Microsoft cég által fejlesztett ingyenes és nyílt forráskódú szövegszerkeztő. Támogatja az automatikus kódkiegészítést (InteliSense, Emmet, TabNine), rendelkezik beépített hibakeresővel, és Git verziókezelő támogatássa, és még számos egyéb szolgáltatással. Bővítmények segítségével további szolgáltatások, és funkciók adhatóak hozzá.

A felhasználói igényeknek megfelelően testreszabható, ebben sokat segítenek a letölthető színsémák. A VS Code több platformos alkalmazás, elérhető Windows, Linux és OS X operációs rendszerekre.

# **KOMMUNIKÁCIÓS FELÜLETEK**

## **Trello:**

A Trello olyan vizuális eszköz, amely segíti a csapatok bármilyen projekt-, munkafolyamat- vagy feladatkövetés kezelésében. Hozzálehet adni fájlokat, ellenőrzőlistákat, vagy akár automatizálást: az csapatok úgy szabhatják testre, ahogy nekik a legjobb.

## **Github:**

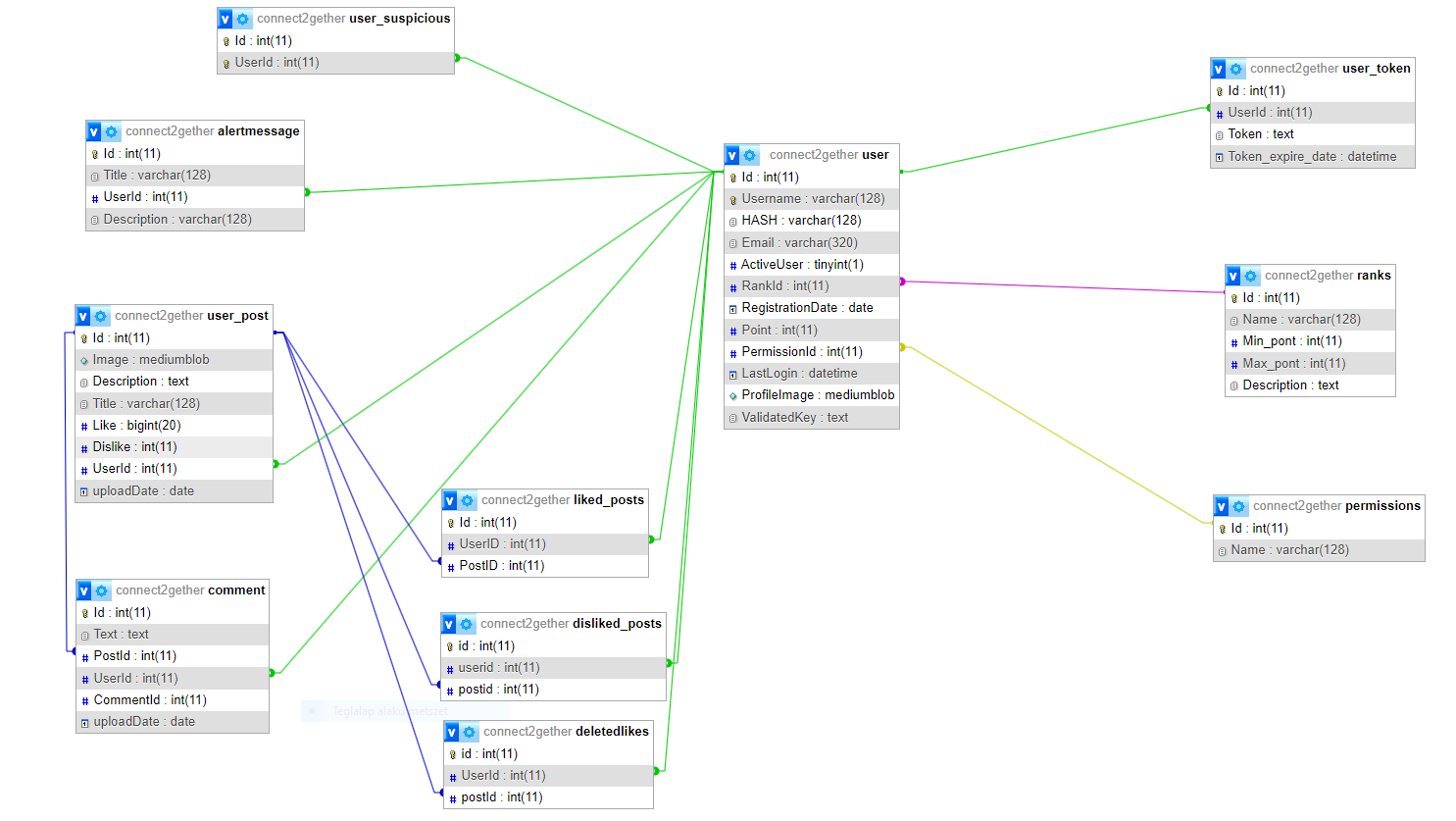
A GitHub ehy felhőalapú platform, amely a Gitet, egy elosztott verziókövetési rendszert használ a magjában. A GitHub platform leegyszerűsíti a projekteken való együttműködés folyamatát, és egy webhelyet, parancssori eszközöket és általános folyamatot biztosít, amely lehetővé a fejlesztők és a felhasználók együttműködését.

## **Discord:**

A Discord, egy VoIP (Voice-over-Internet Protocol) alkalmazás, ami lehetőséget nyújt a kommunikációra a felhasználók között. A készítők azt a szempontot vették figyelembe a tervezés során, hogy a nagy gépigényű alkalmazások futtatásával egyidejűleg működőképes alkalmazást hozzanak létre, úgy, hogy az a gép erőforrásait ne használja feleslegesen.

A funkciói miatt számos területen helyt tud állni, akár a mi projektünk készítése közben fontos kommunikációs platformként üzemel. Könnyű kezelhetősége és funkciói megadják a lehetőséget, hogy létrehozzanak külön szobákat egyes projektekhez vagy az eszközök kijelzőinek streamelésével bemutathassák egymásnak munkáikat.

# **ADATBÁZIS SZERKEZET**



2. ábra - Az adatbázis szerkezete

A **user** táblában tároljuk a felhasználó adatait, az id-ját (ami az elsődleges kulcs), a felhasználó nevét, a hashelt jelszót, az email címet, az ActiveUsert ami kezdetben 0 azaz false értékkel tér vissza a regisztráció után. Miután validálják a felhasználót ez az érték 1 azaz true-val tér vissza és utána be lehet jelentkezni ezzel a felhasználóval. A rankid-t (ami egy idegen kulcs), a RegistrationDate-t ami a regisztráció dátuma, a point ami az adott felhasználó pontszáma, a permissionid (ami szintén egy idegen kulcs), a lastlogin ami a legutolsó bejelentkezés dátumával tér vissza, a profileimage a felhasználó profilképe és a validatedkey ami a profil validásához szükséges.

A **user\_post** táblában tároljuk a postok adatait, az id-ját (ami az elsődleges kulcs), az imaget ami a poston található kép amennyiben a posthoz lett adva kép. A description a post szövege, a title a post címe, a like és dislike, a userid (ami egy idegen kulcs, ez van összekötve a user tábla id-jával) az a felhasználó, akié a post és az uploaddate ami a post feltöltésének dátuma.

A **comment** táblában tároljuk a kommentek adatait, az id-ját (ami az elsődleges kulcs), a text a komment szövege, a postid (ami egy idegen kulcs, ez van összekötve a user\_post tábla id-jával) az a post, ahová írva lett a komment, a userid (ami szintén idegen kulcs, ez van összekötve a user tábla id-jával) az a felhasználó, akiéa komment, a commentid hogyha egy kommentre kommentelnek annak az id-ja és az uploaddate ami a komment feltöltésének dátuma.

Az **alertmessage** táblában tároljuk a felhasználó figyelmeztető üzeneteinek adatait, az id-ját (ami az elsődleges kulcs), a title ami az üzenet címe, a userid (ami szintén idegen kulcs, ez van összekötve a user tábla id-jával) az a felhasználó, aki kapta az üzenetet és a description a komment szövege.

A **ranks** táblában tároljuk a rangok adatait, az id-ját (ami az elsődleges kulcs, ez van összekötve a user tábla rankid-jával), a rang nevét, a rang minimum és maximum pontszáma és a description ami a rang leírása.

A **permissions** táblában tároljuk a role-ok id-ját (ami az elsődleges kulcs, ez van összekötve a user tábla permissionid-jával) és a role nevét.

A **user\_token** táblában tároljuk a tokennek id-ját (ami az elsődleges kulcs), a userid (ami szintén idegen kulcs, ez van összekötve a user tábla id-jával) az a felhasználó, akié a token, maga token és a token lejárati dátuma.

A **user\_suspicious** táblában tároljuk a gyanus felhasználók userid-ját (ami az elsődleges kulcs, ez van összekötve a user tábla id-jával).

A **liked\_posts** táblában tároljuk a userid-t (ami egy idegen kulcs, ez van összekötve a user tábla id-jával) az a felhasználó, aki likeolt és a postid-t (ami szintén egy idegen kulcs, ez van összekötve a user\_post tábla id-jával) az a post, ami likeolva lett.

A **disliked\_posts** táblában tároljuk a userid-t (ami egy idegen kulcs, ez van összekötve a user tábla id-jával) az a felhasználó, aki dislikeolt és a postid-t (ami szintén egy idegen kulcs, ez van összekötve a user\_post tábla id-jával) az a post, ami dislikeolva lett.

A **deletedlikes** táblában tároljuk a userid-t (ami egy idegen kulcs, ez van összekötve a user tábla id-jával) az a felhasználó, aki már likeolt, kapott pontot és hogyha újra likeolja ugyanazt a postot akkor már nem kap pontot, és a postid-t (ami szintén egy idegen kulcs, ez van összekötve a user\_post tábla id-jával) az a post, ami már likeolva volt.

# **WEBOLDAL BEMUTATÁSA**

## **Főoldal:**

A főoldalon a cím alatt található egy rövid leírás az oldal funkciójáról vagy rendeltetéséről. Az oldal alján találjuk a bejelentkezést és regisztrációt.

## **Regisztráció:**

A regisztrációnál meg kell adjunk egy Felhasználó nevet, egy email címet és egy jelszót. A jelszónak tartalmaznia kell kis és nagy betűket, legalább egy darab számot, egy speciális karaktert és legalább 8 karakter hosszúságúnak kell lennie. Regisztrálás után az email címre érkezik egy hitelesítő link amit meg kell nyitni ami átirányít egy oldalra ahol egy gomb nyomással hitelesíthetjük a fiókot. Ez után már bejelentkezhetünk.

## **Bejelentkezés:**

A bejelentkezésnél meg kell adjuk a felhasználó nevet és jelszavunkat. Amennyiben sikeres a bejelentkezés akkor átirányít a felhasználói főoldalra.

A „Forgot your password?” feliratra kattintva módosíthatjuk jelszavunkat ha elfelejtettük volna azt. A szövegre kattintás után meg kell adjuk az profilhoz tartozó email címet. Az oldal küld egy linket amire kattintva átirányít egy oldalra ahol megadhatjuk az új jelszavunkat.

## **Felhasználói oldal:**

Ez az oldal fogad bejelentkezés után. Itt érjük el az összes posztot és itt tudjuk azokat likolni vagy dislikolni, ha postnál egy gombal meg tudjuk jeleníteni a kommenteket és itt mi is tudunk írni azt. Ha a post szövegére kattintunk akkor át irányít egy oldalra ahol csak azt a postot láthatjuk. Ha a post feltöltőjének vagy az eggyik kommentelőnek a nevére kattintunk akkor pedig a profilját tudjuk megtekinteni.

A postok felett található az új post feltöltése aminél meg kell adni egy címet és egy szöveget.

A navigációs panelen érjük el a keresőt amiben felhasználót és postot is kereshetünk, Ha csak szimplán beleírunk akkor a post címét keresi, ha egy @ jelet írunk a szó elé akkor pedig a felhasználó nevét keresi. Ha @ jellel beírunk egy felhasználó nevet és utánna szóközzel elválasztva el kezdünk egy post címet írni akkor az attól a felhasználótól származó posztokba keresi a címet.

A Navigációs panelen érjük el egy gomb segítségével a további lehetőségeket. Itt tudjuk az oldal szinét átváltani világosról sötétre vagy fordítva. A „Details” gombal pedig a profilunk adataihoz navigál.

## **Publikus profil oldal:**

Ez az oldal akkor jelenik meg ha egy felhasználó nevére kattintunk a postnál vagy kommentnél. Itt jelennek meg azok az adatok amik publikusak mint a felhasználó név, rang, pontszám, profil létrehozásának a dátuma, és az utolsó bejelentkezés ideje. A felhasználó postjait is láthatjuk itt.

## **Post oldal:**

Ez az oldal akkor jelenik meg ha egy post szövegére kattinunk. Itt azt az egy postot jeleníti meg az oldal a hozzászólásokkal. Ennek az oldalnak az url -ét kimásolva megoszthatunk egy posttot.

## **Profil oldal:**

A navigációs panelen a „Profile” gombnál a „Details” fülre kattintva érjük el ezt az oldalt. Itt érjük el a személyes adatainkat, mint példáúl az email címünket. A jelszavunkat itt változtathatjuk meg a régi jelszó megadásával. A felhasználó nevünket és email címünket is megváltoztathatjuk itt.

A saját postjainkat és kommenteket is láthatjuk, amiket törölhetünk és módosíthatunk is innen.