

**Baranya Megyei SzC Radnóti Miklós Közgazdasági Technikum**

Szakma megnevezése: Szoftverfejlesztő és –tesztelő

A szakma azonosító száma: 5 0613 12 03

# **Vizsgaremek**

## **Memeparadis**

Készítették: Benkő Zsolt Tamás,  
Libai Kristóf

**Pécs**

**2023**

I.	Bevezető, a feladat rövid ismertetése.....	3
II.	Felhasználói dokumentáció, .....	4
1.	A program általános specifikációja.....	4
2.	Rendszerkövetelmények .....	4
3.	A program telepítése .....	5
4.	A program használatának a részletes leírása .....	7
III.	A fejlesztői dokumentáció, .....	9
1.	Témaválasztás indoklása .....	9
2.	Az alkalmazott fejlesztői eszközök .....	10
3.	Tervezési módszer .....	10
4.	Adatmodell leírása.....	11
5.	Részletes feladatspecifikáció, algoritmusok.....	12
6.	Forráskód.....	14
7.	Tesztelési dokumentáció .....	15
8.	Továbbfejlesztési lehetőségek .....	16
IV.	Összegzés.....	17

## I. Bevezető, a feladat rövid ismertetése

A weboldalunk egy vicces videó, és kép megosztó felület, ahol könnyedén fel lehet tölteni a tartalmakat. A célunk az, hogy olyan tartalmak legyenek a weboldalon, amiket könnyedén elérhetőek, és egy hosszú nap után egy könnyed kikapcsolódásban legyen része a felhasználónak. Ezért törekszünk az egyszerű felületre, elérhetőségre és, kezelhetőségre. Az okból választottuk ezt a témát, mert, évek óta már nézegetünk ilyen videókat/képeket, és azokat egy saját discord csoportba, mentettük le. Jó volt néha napján újra nézni őket, hogy jobb kedvre derüljünk, de sajnos ezek a csoportok nem erre lettek kitalálva, voltak olyan tartalmak, amiket csak hivatkozással tudtuk, berakni a csoportba, és ha azokat az eredeti oldalról/csoportból törölnék, akkor sajnos már nálunk se volt elérhető, ezáltal sajnos sok videót már nem tudunk újra nézni, amikért nagy kár. Ezért is szeretnénk, egy olyan weboldalt csinálni ahol biztonságosan el tudjuk ezeket a tartalmakat menteni, hogy később egyszerűbb módon vissza tudjuk nézni. És azért is szeretnénk egy ilyen weboldalt csinálni, mert nem láttunk olyan kép/videó oldalt, amely külön viccekre és videókra lett fejlesztve, amely az egyszerűsége törekedett.

Az új ismeretek, amiket szereztünk a projekt során az felbecsülhetetlen kezdve attól, hogy milyen is két fejlesztőnek egy hosszadalmasabb és tartalmasabb projekten dolgozni.

Ezek után még azt a tudást is kézhez kaptuk, hogy milyen nehéz is valójában egy backendet és egy frontendet is összekapcsolni.

Még további szakmai tudást, amit szereztünk, ha már eddig csak az alapokról beszéltünk, az az, hogy milyen nehéz is megírni egy Objektum orientált programot, amely képekkel és videók feltöltésével foglalkozik.

Még új ismeretnek mondanám azt is, hogy a projekt későbbi részeiben meg tanultunk jelszót titkosítani és emailt visszaigazolni.

Ezekén kívül sorolhatnánk még számos újdonságot a különböző Postman hibák kezelésén át a szerver oldali buildelési hibákon keresztül a némely apró vessző hibákig. Sok új ismeretet is elsajátítottunk, az angular-ba, a java, és adatbázis fejlesztésben amiket, nem tanítottak meg az iskolába. Az interneten való keresésben is már elsajátítottuk azokat a dolgokat, amivel meg tudjuk oldani a felmerülő akadályokat. A projekt munka készítése alatt sok kérdés tudunk megválaszolni azzal hogy meg tanultuk használni az interneten való keresést, amely a mai világban már nagyon fontos, hogy tudjunk mire rá keresni.

A csapaton belüli munkamegosztás egész egyszerűen működött Zsolt volt a frontendes, Kristóf volt a backendes.

De azért ez nem volt ennyire lekötve, ha kellett segítettünk egymásnak is, de inkább a backend fejlesztésében kellett segíteni, mert ennek az alkalmazásnak egyes részei backenden nehezebbek voltak.

Dolgozni iskolába, és otthon csináltuk, de a legtöbb haladás akkor volt amikor együtt sikerült megoldani egy nagyon problémát.

A megbeszéléseket Discord-on folytattuk le, vagy az iskolában beszéltük meg.

A projekt során két fajta „nyelvet” használtunk backenden a Java Wildflyal Frontenden az Angular volt.

Ezeket a technológiákat azért is választottuk, mert az elmúlt 2 évben nagy részben ezeket tanultuk és ez által valamilyen szinten értettünk is hozzá.

Akár iskolában akár a duális képzőhelyünkön ahol megfelelő tudásra tettünk szert. És az interneten is sok információ áll rendelkezésünkre.

## II. Felhasználói dokumentáció,

### 1. A program általános specifikációja

A mi általunk létrehozott programban elég egyszerű funkciók szerepelnek a regisztráció, a bejelentkezés a kép/videó feltöltés, és a profil szerkesztése. Amit elsőként látni fog a felhasználó az a bejelentkezés oldala, és onnan tovább tud navigálni a regisztrációra. A bejelentkezés után azonnal a fő oldalra dob-be ahol azonnal már a mémeket lehet látni, ezek random módon vannak megjelenítve. A navbar-ba lehet külön rá keresni a képekre, videókra, magyar és, angol tartalmakra, vagy a tag-ekre külön rá keresni. A fő oldalon továbbá a felhasználó név jelenik meg amire rá kattintva a profil oldal jelenik meg, Ahol meg tudja a felhasználó változtatni a felhasználó nevét, emailját, és jelszavát, továbbá itt tud tartalmat feltölteni, és ki jelentkezni. A kép/videó létrehozásánál lehet létrehozni tageket amik segítik a könnyebb keresését az adott tartalomnak, amit keresünk. A mi programunk elég könnyen kezelhető felületként van, létrehozva a felhasználó nem bonyolódik el benne, hogy mi hol van, hanem tisztán átlátható. A bejelentkezés, és regisztráció is úgy működik, mint minden egyes weboldalnál.

### 2. Rendszerkövetelmények

Mivel a weboldal még nincs közzé téve ezért elég sok minden szükséges ahhoz, hogy működjön. Le kell tölteni az Angular-t, a Xamppet, Apache Netbeans-t, Wildfly szerveret. Utána le kell tölteni GitHubról a projektet, és a xampp-on belül a htdocs-ba kell be illeszteni, és ezen felül a projekten belül a frontend mappában lévő két darab php file-ot be kell másolni a htdocs

Processor	Intel Pentium 4 or later	Intel	Intel Pentium 4 or later
Memory	2 GB minimum		
Screen resolution	1280x1024 or larger		
Application window size	1024x680 or larger		
Internet connection	Required		

2.1

	Windows requirements	Mac requirements	Linux requirements
Operating system	Windows 7 or later	macOS High Sierra 10.13 or later	64-bit Ubuntu 14.04+, Debian 8+, openSUSE 13.3+, or Fedora Linux 24+

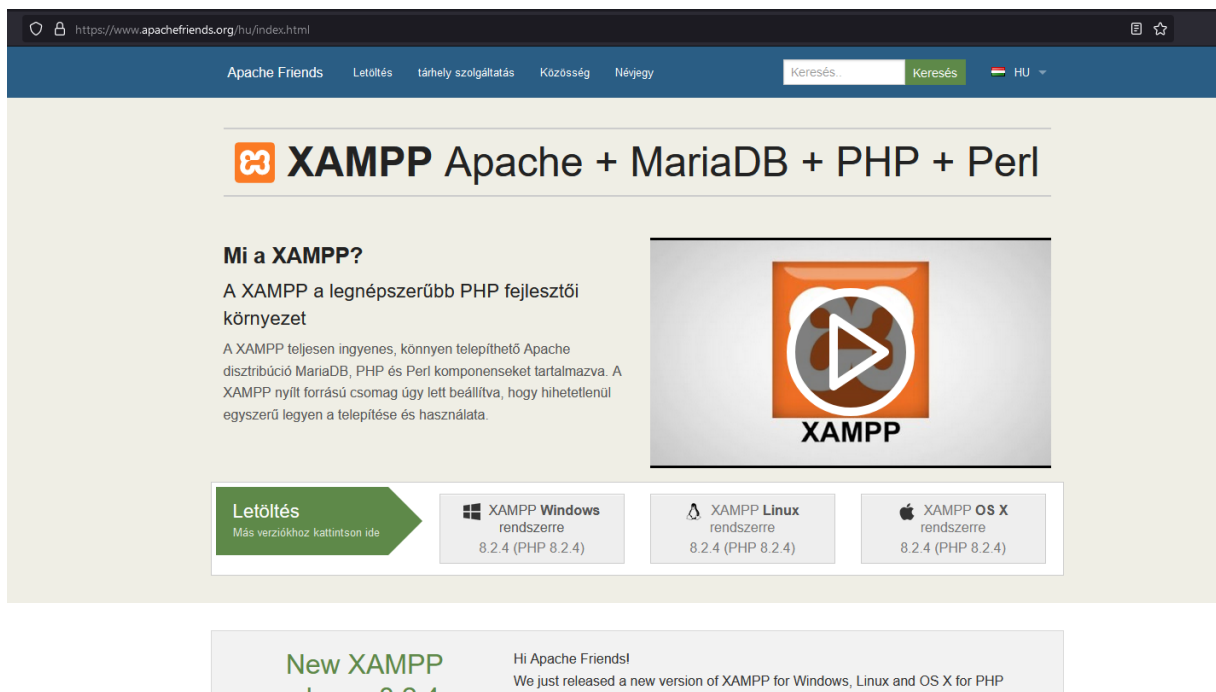
mappába a xampp-on belül, elindítani a Wildfly, Xampp, Angular szervereket, és egy böngészőbe be kell írni a localhost:4200-at, ezek után működni fog a weboldal. Ha közzé van téve a weboldal, akkor csak egy böngésző kell, és a címsorba be kell írni a weboldal elérési

címét. Ezekről függetlenül egy minimális rendszer követelmény kell ahhoz hogy működjön a weboldal. Lásd 2.1-es kép.

A weboldal használata nagyon egyszerű. Az első teendőnk hogy bejelentkezzünk, ha még nincs a felhasználónak saját profilja, akkor a bejelentkező felületen a „Register now”, szövegre kattintson, majd töltse ki a felületet és regisztráljon. Ezután gond nélkül be tud jelentkezni, majd azonnal a fő oldalon találja magát a felhasználó. A fő oldalon lehet nézni, keresni, és szűrni a képeket/videókat. Jobb felső sarokban pedig a felhasználó nevére kattintva tudja a profilját szerkeszteni és képet/videót feltölteni, majd azokat megtekinteni.

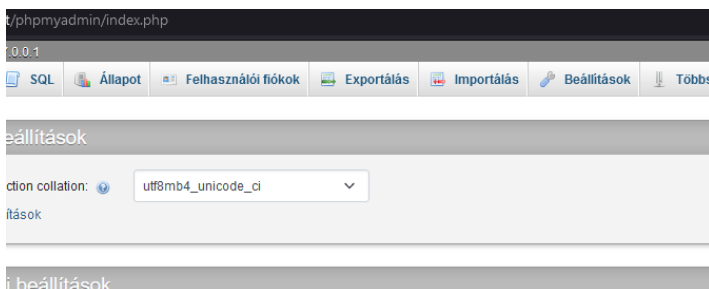
### 3. A program telepítése

Nem igényel különösebb telepítést a programunk hiszen ahogy említjük a fent lévő pontokban ez csak egy szimpla weboldal amely vicces képeket/videókat tartalmaz. Tehát különböző program telepítést nem igényel csak meg kell lennie a megfelelő rendszerkövetelményeknek és a megfelelő alkalmazásoknak amellyel képesek vagyunk futtatni a weboldalt és annak hiba nélküli futtatást.



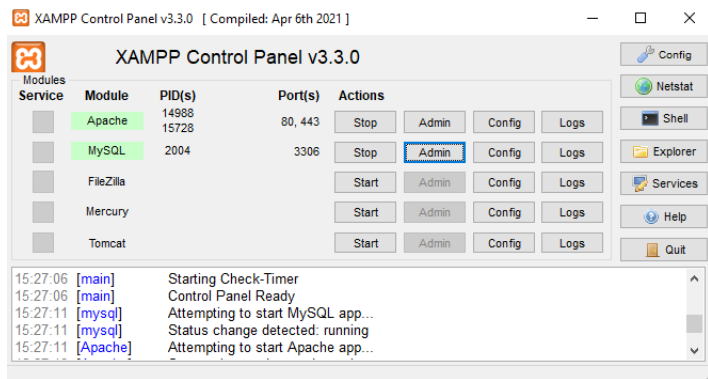
3.1

A fájlokat amelyeket használnunk kell a program futtatása során a program mappájával együtt ha már telepítettünk a Xampp nevű adatbázis kezelő alkalmazását (A letöltése a hivatalos xamp oldalán kell megtenni, lásd 3.1 kép) annak a mappáján belül meg kell keresni a „htdocs” nevezetű mappát, amibe bele kell rakni a projekt mappát és a projekten belül a frontend mappában lévő php file-okat is a htdocs mappába kell bele helyezni, mivel a program csak ebben a mappában képesek futni mert ez a webszerver gyökérkönyvtára az adatbáziskezelő



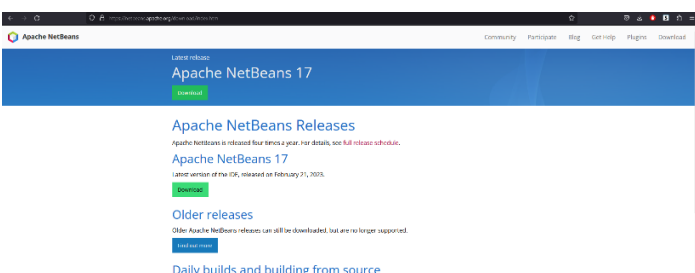
3.3

alkalmazásnak. Ezután a projekt mappában lévő sql adatbázis fájlját (dump) kell be importálni a phpmyadmin oldalán. A szerver elindítása lásd:3.2 kép. A start gombra rá kattintva kell elindítani a megfelelő szervert, majd a késsel jelzett admin mezőre kattintani, ami meg nyitja a phpmyadmin oldalt, és ott kell rá nyomni az exportálás gombra. (lásd:3.3 kép)



3.2

Ezek után ha programot és azt tartalmazó fájlokat a megfelelő helyre tettünk akkor szükségünk van egy backend alkalmazásra fontos hogy ez a Netbeans (A Netbeans telepítését a hivatalos oldalon kell meg csinálni lásd:3.4 kép) legyen. A backend használatához telepítenünk kell egy Wildfly szervert mert a backend ezen keresztül biztosítsa a frontendhez való csatlakozást.



3.4

A Wildflyt az alábbi weboldalról tudjuk leszedni: <https://www.wildfly.org/downloads/> fontos hogy a 26.1.1 es verziónak a Jakarta EE 8 Full & Web Distribution verzióját szedjük le zip formátumban.

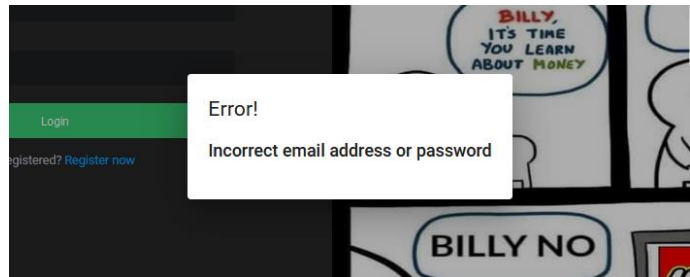
Majd azt kell csinálni a hogy be kell menni a Wildfly mappájába meg kell keresni a bin mappát azon belül a megfelelő operációs rendszerhez való add-user fájlt Windows esetében ez a .bat Linuxban pedig Shell script.

Majd ezek után a fájlt kétszer kell megnyitni, mert kétszer kell regisztrációt véghez vinni.

Ezek után miután létrehoztuk a felhasználónkat belépünk a netbeans-be, és meg nyitjuk bal felső sarokba le nyitjuk a file opciót, és az open projectre rá kattintunk, és ki választjuk a projekten belüli backend mappába lévő java fájlt. Ezek után rá nyomunk a servicesre és a databases-en belül a memeparadis-ra jobb klikkel el startoljuk a kapcsolódást, ezután, a servers fülre kattintva megjelenik a wildfly server amit ugyan úgy elindítjuk mint az adatbázis kapcsolatot. Mind ezek után a projekt mappán belül meg nyitjuk a frontend mappát azon belül is a memeparadis-t és a cím sorba be írjuk hogy cmd majd enter, ki dobja a cmd parancssort amibe azt kell bele írni hogy „ng serve -o”, egy kis idő után meg is nyitja a weboldalt ami használatra kész.

#### 4. A program használatának a részletes leírása

Aki elsőként nyitja meg az oldalt, először a bejelentkezés oldalát láthatja. Az oldal közepe két részre van osztva, jobb oldalon egy teljesen véletlen módon meg jelenített meme szerepel. amire ha rá kattint ki is tudja nagyítani hogy nagy méretarányba is meg tudja nézni, ebből kilépni úgy tud hogy az oldal el szürkült részére klikkel, amivel azonnal vissza is ugrik a kép. Az oldal bal oldalán a bejelentkező felület látható, ahol először is üdvözlő a weboldal, az első ki tölthető mezőbe a felhasználónak az email címét kell be írnia. A második mezőbe pedig a jelszavát kell be írnia. Ezek után ha rá nyom a Login gombra és jól írta be az adatokat akkor be foglya dobni a fő oldalra, de ha nem jó valamelyik adat akkor egy Error hibát dob fel (lásd:5.1 kép)egy ablakba, amit be zárni csak úgy tud ha az elszürkült részre klikkel.

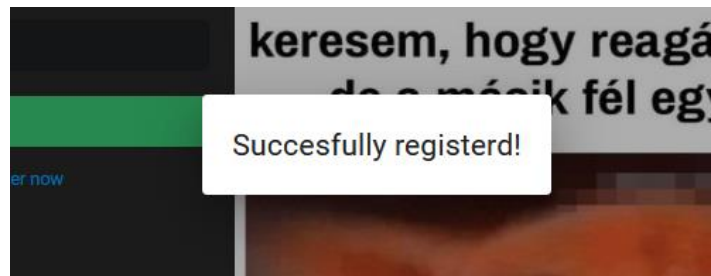


5.1

A gomb alatt található egy rövid szöveg ami azt jelenti hogy, ha még nem vagy regisztrálva akkor regisztrálj most. A kékkel színezett részre klikkelve át is dobja a regisztrációs oldalra.

A regisztrációs oldalon középén meg jelenő űrlapot kell kitölteni. Az első ilyen mező az a felhasználó név, szöveg doboza, a második a felhasználó emailje, majd a jelszó, és a születési év. A felhasználó név minimum 5 karakter hosszú-nak kell lennie ha nem annyi akkor hibát fog ki dobni. Az email címnél csak arra kell figyelni hogy egy érvényes és élő email címet adjunk meg. A jelszóban kötelezően kell szerepelnie e, egy darab nagy betűnek, számnak, és speciális karakternek. A születési dátum-nál csak az tud regisztrálni aki betöltötte a 18-ik élet évét, ennek oka az hogy érzékeny tartalom is lehet az oldalon. Végül ha minden ki lett töltve akkor a

Registration gomb lenyomásával regisztrálni tudunk, ha minden kirtérium sikeresen teljesült. Ha a regisztráció sikeresen teljesült akkor vissza dob a bejelentkezési felületre ahol azonnal be is tud jelentkezni a felhasználó (lásd:5.2 kép). Ha a



5.2

felhasználó mégse szeretne regisztrálni akkor a „Already have an account?” szövegre klikkelve vissza dobja a bejelentkezés oldalára. Ha a bejelentkezés sikeres volt akkor be fog dobni automatikusan a fő oldalra.

A fő oldalon találhatóak a tartalmak véletlen szerűen ki dobva, itt lehet keresgélni, és szórakozni. A képekre rá klikkelve meg nagyobbítani lehet őket, a videóknál pedig a jobb alsó sarokban lehet ki nagyítani, és hangosítani, vagy akár bele is lehet, pörgetni bárhova ahova a felhasználó szeretné. Az oldal tetején lévő hosszú sötét sávon lehet navigálni. A Home gomb az első, amire rá klikkelve csak újra tölthetjük az oldalt. A következő opció a „Picture” amire rá klikkelve csak azokat a tartalmakat foglya ki dobni amik csak képek. A következő gomb a „Videos” ami pedig csak a videókat foglya kidoni. A következő pedig az „English”, ami csak az angol szöveges tartalmakat foglya kidobni, és az utolsó ilyen filter gomb pedig a „Hungary”

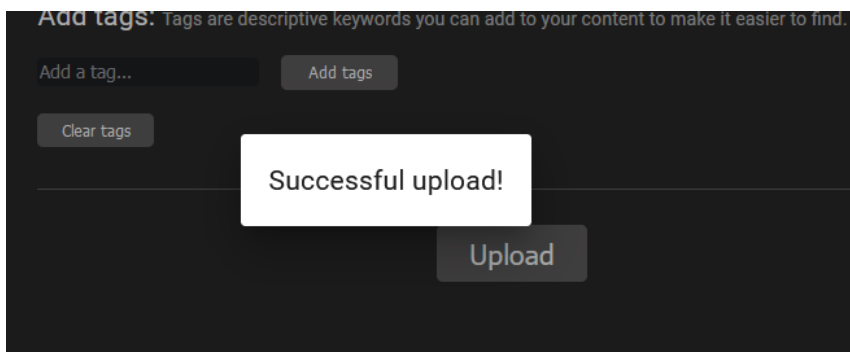
ami csak a magyar tartalmakat foglya ki dobni. Ezek mellet van egy kereső mező, ami a tagek alapján keres, például ha a képen vagy videóba van egy alma és be írjuk a kereső mezőbe, akkor csak azokat a tartalmakat foglya kidobni, amiben szerepel az alma. A legjobboldalibb szöveg az mindenkinek egyéni lesz, mert az a felhasználó név, amire rá klikkelve meg tudjuk nézni a profilunkat. A profil oldalunkon először azt vehetjük észre hogy a felső sötét csíkban csak a felhasználó név és a „Home” gomb maradt. A „Home” gomb visszavisz a fő oldalra, a felhasználó név gomb pedig csak újra betölti a profil oldalt. A profil oldal is két nagy részre van osztva. Bal oldalt található a navigációs oszlop, ami a „Profile” szóval kezdődik. A következő gomb az a „Profile Setting” amire rá klikkelve a profil adatokat mutatja meg. Ez után az „Upload Content” gomb van, ami át irányít arra a felületre ahol fel lehet tölteni tartalmakat az oldalra. Ez alatt van az „Uploaded Content” ahol azok a tartalmak vannak, amit a felhasználó feltöltött. Majd végül legalul van a „Sign out” gomb, amire rá klikkelve, kijelentkezünk, és azonnal visszaszok a bejelentkezési felületre. A profil oldal jobb nagy részén található a gombok által meg nyitott opciók, alapértelmezetten a „Profile Setting” (lásd:5.3 kép) lesz meg nyitva. Elsőként láthatjuk hogy a felhasználó név ott van amit a jobb oldali „Editing” gomb

5.3

segítségével meg lehet változtatni. Amikor a gombot megnyomjuk akkor a felhasználó névnél át alakul egy kitölthető felületté, amiben meg változtathatjuk a felhasználó nevet, eközben a gomb is meg változik „Send!”-re. Ha változtatunk a felhasználó néven akkor a „Send!” gomb meg nyomásával sikeresen el tudjuk végezni ezt a műveletet, de ha nem változtattunk rajta semmit akkor a „Send!” gomb meg nyomásával vissza tudjuk állítani alap helyzetbe a felhasználó nevet. Az alatta lévő Email címet is ugyan ezzel a móddal meg lehet változtatni. A jelszó meg változtatását ugyan úgy a tőle jobbra levő „Edititng” gombbal tudjuk meg csinálni. Amikor meg nyomjuk a gombot akkor két darab kitölthető mezőt látunk, az első mezőbe a jelenlegi jelszót kell be írni, és az alattiba pedig az új jelszót kell, majd a „Send!” gomb meg nyomásával végre tudjuk ezt hajtani, de csak akkor fog működni a változtatás ha mind a kettő mező a kritériumoknak meg felel, ha például csak az egyik mező van csak ki töltve és így akarjuk el küldeni a változtatást akkor nem fog meg változni, csak vissza vonja a kitölthető



mezőket. Végül legalul a regisztráció pontos ideje látható, amit nem lehet meg változtatni, csak tájékoztatás gyanánt van ott. Az „Upload Content” gomb meg nyomásával megnyitjuk azt a felületet jobb oldali nagy részben ahol a tartalmakat tudjuk fel tölteni. Az első dolog ami ott van az egy lista doboz, amiben meg kell adni hogy milyen tartalmat szeretnénk feltölteni, két opció van videó vagy kép-et, ha ez ki lett választva akkor a tőle kicsit jobb oldalra lévő „Choose file...” gombra klikkelve tudjuk ki választani a saját mappájából azt a tartalmat amit fel szeretnénk tölteni. Ha a kép opciót választottuk akkor csak a képeket foglyuk tudni ki választani, ha a videókat, akkor csak a videókat



5.4

tudjuk ki választani a mappáinkból. Ha ez sikerült akkor a következő választható listából választani kell hogy angol, magyar, vagy „Other”=más nyelven van a tartalom, az „Other”-be bele tartozik az is hogy ha semmi féle nyelv vagy szöveg nincs a tartalomba. A következő rész az a „Add tags” rész, ahol kulcs szavakat adhatunk meg a feltölteni kívánt tartalomhoz hogy könnyedén rá tudjunk keresni. A szöveg dobozba be írunk egy rövid szót, majd rá nyomunk az „Add tags” gombra, ami hozzá adja a tartalomhoz azt a szót amivel majd tudunk rá hivatkozni. Egyszerre többet is hozzá tudunk adni, és ha egy se szükséges akkor a „Clear tags” gombbal ki tudjuk törölni mindegyiket, de ha nem akarjuk törölni az összeset akkor csak a törölni kívánt szóra rá klikkelünk és már torlódik is. Ha megfelelően minden ki lett töltve akkor az „Upload” gomb segítségével fel tölthetjük a tartalmat, egy kis idő után pedig ha sikeres volt a feltöltést akkor egy felugró ablakkal foglya jelezni.(lásd:5.4 kép)

Ha pedig valamit nem jól töltöttünk ki akkor egy Error oldalt fog be dobni. Az „Uploaded Content”-re klikkelve pedig meg tudjuk nézni a felhasználó által feltöltött tartalmakat, ami a legfisebbtől a leggrégebben feltöltött tartalomig foglya meg jeleníteni. Itt is ugyan úgy lehetrá nagyítani a képekre videókra mint a fő oldalon. A profil oldal utolsó funkció gombja pedig a „Sign out” ami kijelentkezteti a felhasználó és ki dobja a bejelentkezési felületre.

### III. A fejlesztői dokumentáció,

#### 1. Témaválasztás indoklása

Azért választottuk ezt a témát mert a mai világban úgy gondoltuk hogy az emberek szórakoztatása a legfontosabb, az emberek többsége a órákat töltene a TikTok-on, a mi oldalunkat is lehet hozzá hasonlítani, de mi a mémeket szeretnénk fókuszban tartani. Ezen az oldalon a felhasználó nem készíti a tartalmat hanem csak egy tartalmat feltölt, nem csak azokat a mémeket lehet ide fel tölteni amiket a felhasználó csinált, hanem ide bárkiét fel lehet tölteni, mert az oldal lényege a mémek eltárolása és meg nézése, hogy bármikor fel lehessen lépni az oldalra nevetni egyet. Ha egy ember lát egy vicces képet és szeretné másokkal is megosztani vagy inkább csak el tárolni hogy könnyedén hozzá férjen és hogy ne a saját számítógépén

foglalja helyet, csak fel tölti az oldalunkra és bármikor meg tudja nézni akadály nélkül és akkor más vicces tartalmakra is rá talál. Nyilván első sorban a fiatalok vannak megcélózva hisz inkább ők nézik az ilyesmi tartalmakat.

Még azért is álltunk emellé a téma mellé mert akár különböző korosztályokat is elérhetünk vele nem csak a fent említett korosztályt, mert akár régi fa vicceket is fel lehet öltani kép vagy videó formátumban.

Még azért is az alábbi oldal mellett döntöttünk mert hamar terjedni tudnak a rajta lévő képek mert ha tetszik egy adott embernek tovább tudja küldeni az ismerősének/barátjának és együtt tudnak rajta nevetni akár boldogságuktól sírni is.

Mert köztudott hogy viccek a nap bármelyik pontját feldobhatják akár hajnalban akár késő éjjel is, bárkinek bármilyen humor érzéke van.

## 2. Az alkalmazott fejlesztői eszközök

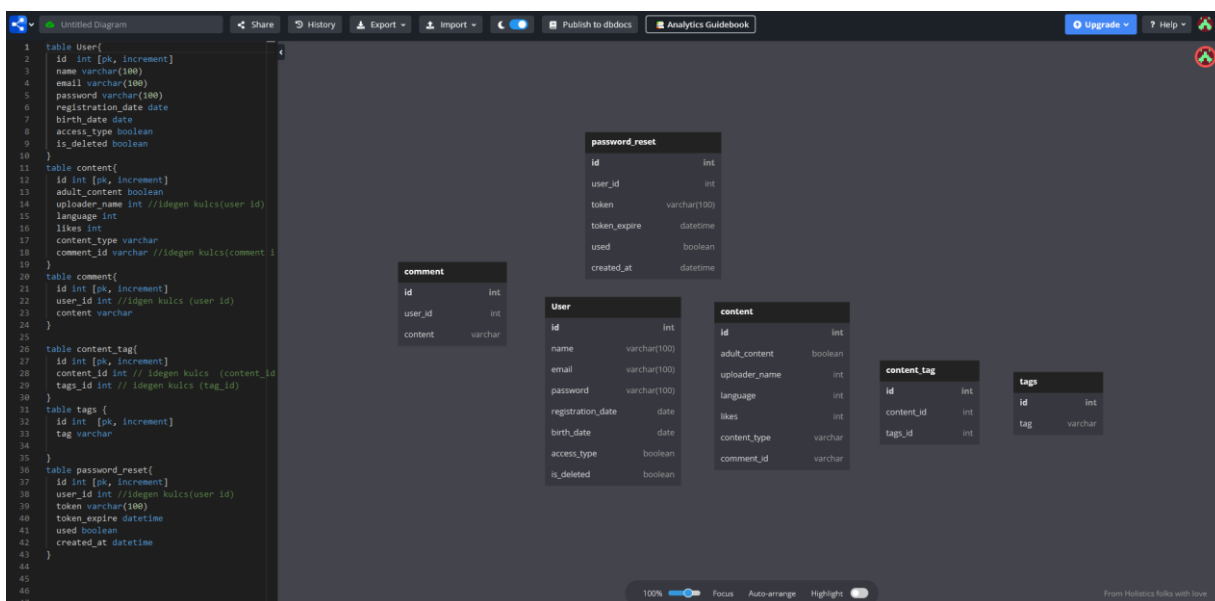
Az alkalmazott fejlesztői eszközök első sorban a programozási nyelvek voltak amely Java és Angular volt adatbázis kezeléshez pedig a Phpmyadmint vagy egykoronként néha a Xamppot használtuk mint Mysql biztosító szerveret.

A folyamatos rugalmasság érdekében GitHubot használtunk amely könnyen segíthetett a dokumentációnk megírásában és folyamatos előrehaladásában. Egyéb programokat nem használtunk. A projekteben található egy kevés php kód is ami a tartalmak feltöltésért szolgál. A frontend részben még egy 23 sornyi Javascript is található hogy a képekre rá lehessen közelíteni. A frontendben egy kevés bootstrap, és angular material is található amivel ikonokat és a responzivitást oldottuk meg.

## 3. Tervezési módszer

Az alkalmazás megtervezése egyszerűen de mégis gyorsan zajlott hogy mit szeretnénk értékesíteni a vizsgára.

Kezdeti tervezés az volt hogy megterveztük az adatbázist a <https://dbdiagram.io/> oldal (lásd:8.1 kép) segítségével majd a weboldal design tervét találta ki Zsolt, amit paintben tervezet meg



8.1

sikeresen, később az eljárások következtek majd külön a backend és a frontend oldali programozás megkezdése.

Frontend oldalon először meg lett csinálva a regisztráció és a bejelentkezési felületet.

Backended is a regisztráció és a bejelentkezés következett amelyet Postman segítségével teszteltünk. Frontendenben következnek profil oldal lett elkészítve majd a fő oldal lett utoljára el készítve. Backendben eközben a kép feltöltés/videó feltöltés módszere következett.

Majd elkezdtek megtervezni a jelszó titkosítást amely könnyedén meg lett oldva.

Ha az Objektum Orientáltságról akarunk beszélni akkor a programunk 4-5 osztályból áll kezdve a Usertől Contentig.

Természetesen a különböző API, Controller és Service rétegeket se hagyhatjuk ki.

Az OOPt az szemlélteti hogy minden osztályban nem egy változó van hanem több pl a Userben (lásd:8.2 kép) :

Vannak egyedi változók mind például: az email, name, és az id.

```
public User(Integer id, String name, String email, String password, Date registrationDate, Date birthDate, boolean ac
    this.id = id;
    this.name = name;
    this.email = email;
    this.password = password;
    this.registrationDate = registrationDate;
    this.birthDate = birthDate;
    this.accessType = accessType;
    this.isDeleted = isDeleted;
```

8.2

#### 4. Adatmodell leírása

Az adatbázis tábláink a következő képpen épültek fel.

**User:** Az alábbi tábla az adatbázisban a felhasználók adatait kezeli ahol az alábbi adatokat tárolják : Id, nevét, email, regisztráció ideje, a jelszó, születési dátum.

A User tábla még arra szolgál hogy eltárolja a már regisztrált felhasználókat akiknek ezáltal nem kell folyamatosan újra regisztrálni az oldalra hanem csak szimplán belépni ezzel eltárolva a bejelentkezési adataikat .A jelszó MD5-ös titkosítással van ellátva.

**Content:** A content tábla név magáért beszél mert az alábbi tábla az oldalon lévő tartalmakat tárolja az egyedi azonosítójától kezdve.

Majd jön az adult\_content oszlop amely azt szolgálja hogy az egyes képek felnőtt tartalmak e vagy tartalmaznak e káromkodást ami miatt felnőtt tartalomnak minősíthető.

Az uploader\_name oszlop a feltöltő nevét tartalmazza, ami össze van kötve a User tábla id-val.

A language azt tartalmazza hogy az alábbi kép/videó milyen nyelven van.

A likes oszlop a poszton való kedvelések száma.

A content\_type oszlop arra szolgál hogy megállapítja hogy az adott tartalom videó-e vagy kép-e. ezáltal a felhasználók dolgát és megkönnyíti hogy milyen fajta tartalmat akarnak keresni.

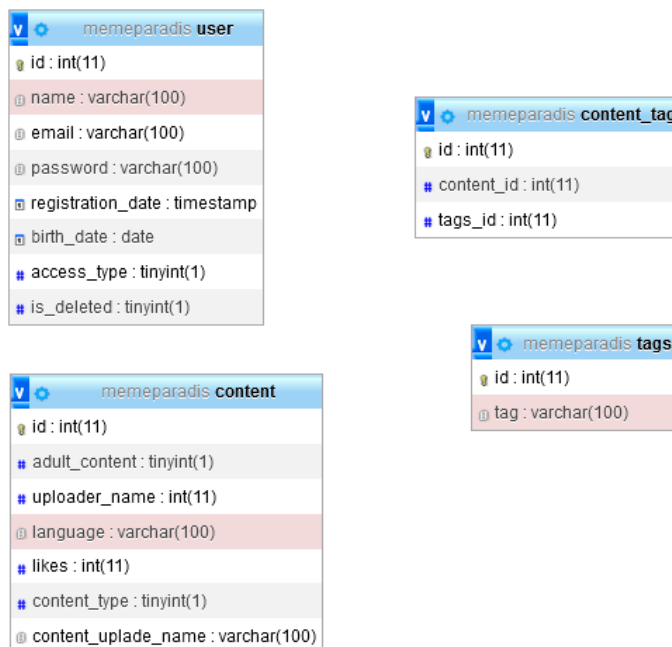
Az uploader\_name oszlop abban segít nekünk hogy tudjuk melyik felhasználó töltötte fel az adatot tartalmat ezzel megakadályozva azt hogy mindenki betudjon írni egy random felhasználót hogy kinek a nevével szeretné fel tölteni az adott tartalmat.

**Content tag:** A content\_tag tábla köti össze a content és a tag táblát hogy egy tartalmon több tag is legyen a tartalmak keresése érdekében. A táblában van először a content\_id, ami össze

van kötve a content tábla id-val. A másik adat az pedig a tags\_id ami pedig a tags táblával van össze kapcsolva.

**Tags:** A tag tábla legfontosabb adata a tag, amiben a szöveges kereső kulcsok vannak tárolva.

---



## 5. Részletes feladat-specifikáció, algoritmusok

Az alábbi pontban a különböző eljárások funkciót és paramétereit ismerttettem (backend). A Userrel kapcsolatos eljárások a delete, create, update jöttek létre. contentnél szintén.

A Tageknél csak egy update tag jött létre amivel tudjuk módosítani a képünkön használt tageket.

A checkEmailUnique az email ellenőrzi hogy egyedi e az email cím míg GetMostLikedPost a legtöbbet likeot kapott posztot adja vissza.

addLiketoContent: ez az eljárás egy bemenő paramétert vár még pedig egy idt ő azt biztosítja hogy ha kedvelnek egy tartalmat ahhoz hozzá is legyen adva a kedvelés ennek a testvér párja a removeLikeContent ami elveszi a likeot a tartalomról ha azt valaki ki kedveli.

addNewUser: Ez az eljárás az aki biztosítsa felhasználót a regisztrációjáról négy paramétert vár be egy felhasználó nevet email címet születési dátumot és jelszót.

A felhasználót nem engedi regisztrálni az oldal ha nincs meg 18 mivel lehetnek felnőtt tartalmú tartalmak amik a nagyon durva káromkodásnál állnak meg.

login: Biztosít minket az oldalba való bejelentkezéshez meg azt is ellenőrzi hogy az adott felhasználó regisztrált e az oldalon ha nem megkéri arra hogy regisztráljon.

CreateContent: Az alábbi eljárás arról biztosít minket hogy lehessen az oldalra tartalmakat feltölteni ő maga a feltöltő eljárás 6 bemenő paramétere van az hogy felnőtt tartalom e ,a feltöltőjének az azonosítója milyen nyelven van az adott tartalom, mi a típusa kép vagy videó és a tartalomnak az idja.

CreateContent tag: Az alábbi eljárás biztosít minket arról hogy a tagek a megfelelő tartalomhoz kerüljenek és ne másikhoz.

A createTag. Maga a tagek létrehozását biztosítsa a képekhez.

deleteUser és deleteContent:A felhasználói fiók és az adott tartalom törlését szolgálják ki.

getAllContentRand:Lekéri az összes tartalmat véletlenszerű sorrendben.

getContentById:Lekéri a tartalmat az azonosítója alapján.

getContentByUserId:Lekéri a felhasználó azonosítója alapján hogy az adott felhasználó milyen tartalmakat töltött fel.

getContentBytag: Az alábbi eljárás lekéri az adott tartalmakat az alapján a tag alapján amire rákeresett.

getEnglishContents: Az alábbi eljárás lekéri az Angol nyelvű tartalmakat az oldalról véletlenszerű sorrendben természetesen.

getHungarianContents: Az alábbi eljárás lekéri a Magyar nyelvű tartalmakat az oldalról véletlenszerű sorrendben természetesen.

A getPictures és a getVideos : Leszűri a képeket és videókat véletlenszerű sorrendben.

getHowManyContent:Megszámolja a tartalmak számát az oldalon.

updateUserName:Biztosítsa az adott felhasználót a felhasználói oldalán arról hogy tudja változtatni a nevét.

updatePassword:Biztosítsa a felhasználót a jelszavának változtatási lehetőségére.

updateEmail:Ha megváltozott a felhasználó email címe biztosítsa ennek is a változtatásáról.

updateBirthDate: Biztosítsa a felhasználót a születési dátumának megváltoztatásra ha adott esetben elírta volna regisztráláskor azt.

Az alábbi pontban a különböző eljárások funkciót és paramétereit ismerttettem (frontend).

goToPage: Különböző oldalakra lehet át navigálni vele. Egy bemenő paramétere van, ami az oldal neve amire át szeretnénk navigálni. Gombokra hívtam hogy oldalak között lehessen lépkedni.

loginProduct: A bejelentkezés működteti, egy formba van meg hívva (ngsubmit)-al, egy bemenő paramétere van és az a formban ki töltött felhasználó adatok fogadására van, típusa any.

opendialog: Meg nyit egy komponenst ami egy egyedi alert.

searchcontent: A fő oldalon a kereső gombra van meg hívva, ebben van benne a keresés eljárás meghívása.

loadallpicture: Egy eljárás meg hívás van benne, és csak a képeket szűri be a weboldalra.

loadallvideo: Ugyan az mint a loadallpicture csak videókat szűr be a html oldalra.

loadEnglish: Meghívja a csak angol tartalmakat és be szűrja az oldalra.

loadHungary: Ugyan az mint a loadEnglish csak a magyarokat adja vissza.

editusername: A felhasználó nevet lehet meg változtatni, egy bemenő paraméter van ami any típusu.

edituseremail: Az email címet lehet meg változtatni vele.

edituserpassword: A jelszót lehet megváltoztatni vele.

logoutfunction: A kijelentkezésben segít.

creatContentfunctione: A tartalom feltöltésért van, több kérés is meg van hívva benne, és ebben van meg hívva a php kód.

loaduploadedcontent: A felhasználó által feltöltött-tartalmakat adja vissza és adja hozzá egy div elemhez.

onProductCreate: A regisztrációhoz kell.

Van két darab model osztály is ami az egyik a user osztály reprezentálja, a másik pedig a content-et.

## 6. Forráskód

A forráskódnál is törekedtünk az egyszerűsége. minél átláthatóbb és rövidebb legyen. hogy bővíteni és javítani is könnyedén tudjuk A frontend kódban túlnyomó többségben csak API kérések vannak, amiket a backend-nek továbbít hogy a megfelelő adatokat adja vissza.

```
onProductCreate(addNewUser: any){
  console.log(addNewUser);
  this.http.post('http://127.0.0.1:8080/MemeparadisEE7-1.0-SNAPSHOT/resources/User/addNewUser',addNewUser).subscribe((res)=>{
    console.log(res);
  });
}
```

A fenti képen egy ilyen látható ahol, az új regisztráció megy végbe, egy egyszerű post kérés segítségével, Amikor a felhasználó kitölti a regisztrációs felületet és rá nyom a regisztrációs gombra akkor erre a címre „<http://127.0.0.1:8080/MemeparadisEE7-1.0-SNAPSHOT/resources/User/addNewUser>” elküld egy kérést a backendre, hogy hozza létre a usert. A frontend-ben sok hasonló ilyen kóddal dolgozunk, hogy elérjük a backendet.

Számos fontos kódrészlet volt az oldal létrehozásának a működése során ami segítette a tartalm Az addNewUser eljárás forráskódja backenden a különböző Service,Controller illetve modell rétegeken. ak és a felhasználók kezelését is.

```
return bigInt.toString(16);
}
}
public String addNewUser(User u){
    EntityManagerFactory emf = Persistence.createEntityManagerFactory(Database.getPuName());
    EntityManager em = emf.createEntityManager();

    try{

        StoredProcedureQuery spq = em.createStoredProcedureQuery("addNewUser");

        spq.registerStoredProcedureParameter("nameIN", String.class, ParameterMode.IN);
        spq.registerStoredProcedureParameter("emailIN", String.class, ParameterMode.IN);
        spq.registerStoredProcedureParameter("passwordIN", String.class, ParameterMode.IN);
        spq.registerStoredProcedureParameter("birth_dateIN", Date.class, ParameterMode.IN);

        spq.setParameter("nameIN", u.getName());
        spq.setParameter("emailIN", u.getEmail());
        spq.setParameter("passwordIN", encryptString(u.getPassword()));
        spq.setParameter("birth_dateIN", u.getBirthDate());

        spq.execute();
        return "Successfully registered!";
    }
    catch(Exception ex){

        if(ex.getMessage().equals("org.hibernate.exception.ConstraintViolationException: Error calling CallableStatement"))
            return "Some unique value is duplicate!";
    }
}
```

```

@POST
@Path("addNewUser")
@Consumes(MediaType.APPLICATION_JSON)
public Response addNewUser(User user){
    String result = us.addNewUser(user);
    return Response.status(Response.Status.OK).entity(result).type(MediaType.APPLICATION_JSON).build();
}

```

```

    return BigInt.toString(10);
}

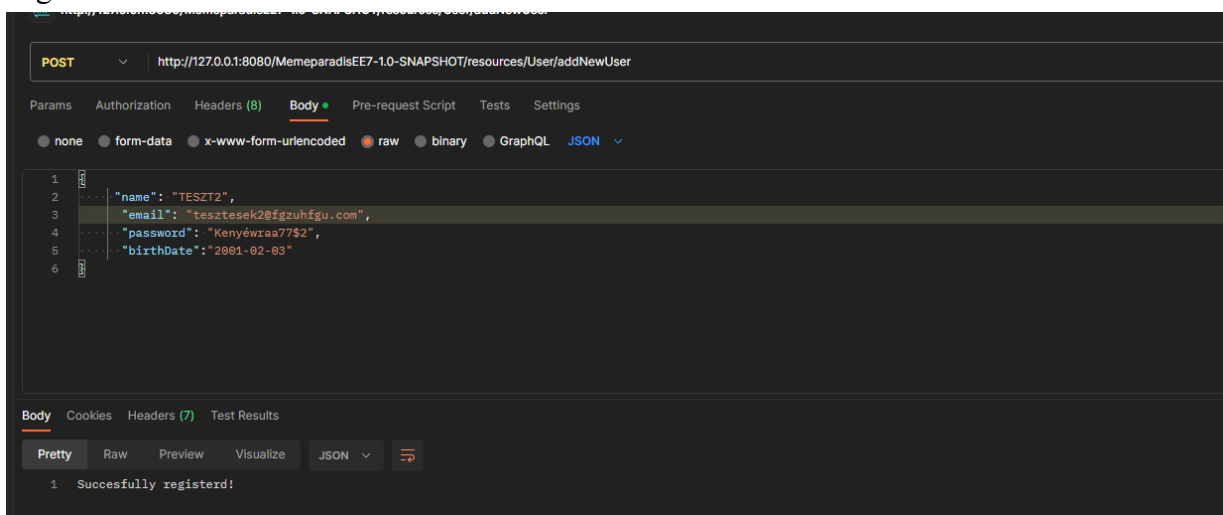
public String addNewUser(User user){
    String result = "";
    if( !(u.checkEmailUnique(user.getEmail()))){
        result = "Email is already taken!";
    }
    else if(user.getName().length() < 5){
        result = "firstname must be at least 5 characters";
    }
    else{
        try{
            User.validatePassword(user.getPassword()); result = u.addNewUser(user);
        }
        catch(Exception ex){
            result = ex.getMessage();
        }
    }
    return result;
}
}

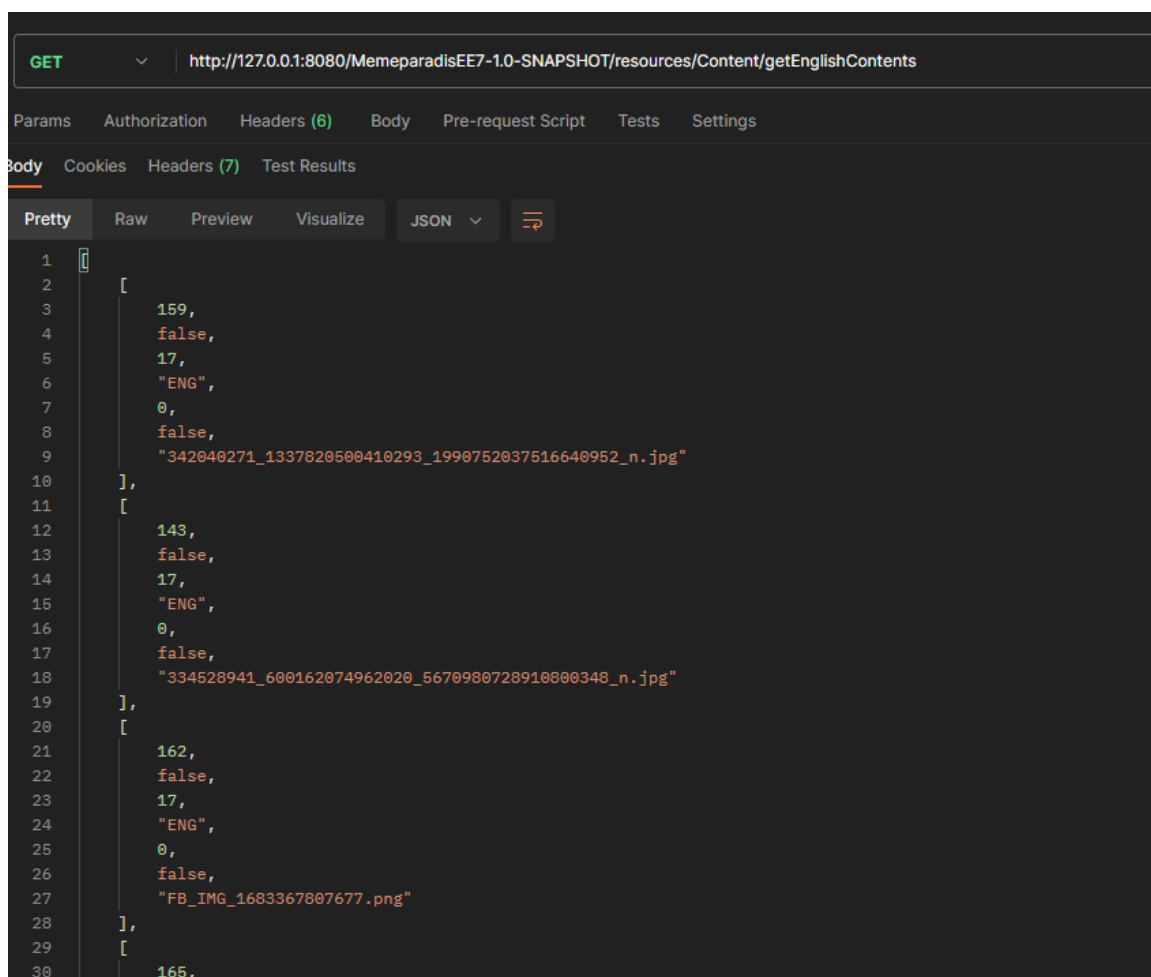
```

Az encryptString a jelszónak a titkosítását biztosítja.

## 7. Tesztelési dokumentáció

A backend-es tesztelések nagy részét postmanban volt csinálva, onnan ki derült mindig hogy a backenden jól lett e lekezelve az eljárás. Mindegyik használatban lévő eljárás tesztelését ott végeztük el.





```
GET http://127.0.0.1:8080/MemeparadisEE7-1.0-SNAPSHOT/resources/Content/getEnglishContents

Body
Pretty
1  [
2    [
3      159,
4      false,
5      17,
6      "ENG",
7      0,
8      false,
9      "342040271_1337820500410293_1990752037516640952_n.jpg"
10   ],
11   [
12     143,
13     false,
14     17,
15     "ENG",
16     0,
17     false,
18     "334528941_600162074962020_5670980728910000348_n.jpg"
19   ],
20   [
21     162,
22     false,
23     17,
24     "ENG",
25     0,
26     false,
27     "FB_IMG_1683367807677.png"
28   ],
29   [
30     165,
```

A frontend tesztelésnél próbáltunk minden szóba jöhető problémára, egy kezelést létre hozni, például a felhasználó nem helyesen tölti ki a mezőket, vagy nem felelnek meg az adatok a minimum követelményeknek, vagy a nincs keresési eredmény, ezeket próbáltuk megoldani egyéni alartekkel hogy ne consolba írja ki a hibákat, és hogy a felhasználó megfelelően értesüljön a hibákról. Igyekeztünk le kezelni azokat a dolgokat is hogy ha üres mező ellenére is meg nyomja a gombot akkor se dobjon ki hibát.

## 8. Továbbfejlesztési lehetőségek

A jövőben szeretnénk még foglalkozni a weboldallal, bővíteni egyszerűsíteni. Például a tartalmakhoz való hozzászólást lehetőségét is fejlesztési lehetőségként lehet mondani mivel ha egy adott tartalom nagyon tetszik egy felhasználónak biztosan hozzáfog szólni hogy több ember is láthassa ezt a tartalmat. A továbbfejlesztéshez sorolható még az email visszaigazolás amikor regisztrál egy felhasználó kapjon arról emailt hogy sikeresen regisztrált az oldalra. Még a tovább fejlesztési lehetőségek közé tartozik a Forgot Password opció is ami magyarul az elfelejtett jelszó ami arra biztosít hogyha elfelejtet a jelszod kapsz egy tokenes linket ami egy adott időig él és ezen új jelszót tudsz generáltatni ezáltal sikeresen újra betudsz majd lépni az Oldra. Ezek voltak azok amik nem fértek bele az időnkbe. Ami tovább fejlesztéshez is mondható az hogy szeretnénk bővíteni majd gif-ekkel is a weboldalt. Azon is gondolkodtunk hogy elérhetővé tesszük az interneten is a weboldalt. Az is egy további célunk hogy az adatok maximum értéke ne csak 10 mb legyen. Lehetővé tenni az oldalt több fajta nyelven és ne csak angolul legyen



hanem akár magyarul is mert ugye Magyar tartalmakat is lehet elérni vagy akár Franciául, Japánul Portugálul vagy számos nyelveket amelyek elterjedtek a mai világban.

Továbbfejlesztésnek mondható még a beszélgetés rész is a felhasználók között amely segítségével könnyen mutatnak meg olyan tartalmakat az oldalról amit a másik nem lát.

A support lehetősége ez arra szolgálhatna ha az adott felhasználó nem tudna tartalmat feltölteni miért nem tud hogy hálózati vagy esetleg szerver oldali probléma e a hiba.

A tovább fejlesztések közé tartozhat az is hogy az oldal fő oldalán a legnépszerűbb tageket megjelenítjük.

## **IV. Összegzés**

A vizsgaremek elkészítése során számos területen segített fejlődni, mind szakmailag, mind pedig más területeken. A projektben való munka lehetőséget adott arra, hogy elmélyítsük a front-end, és backend fejlesztési készségeinket és megismerjük a kód nyelvek legújabb funkcióit és lehetőségeit.

Az egyik legnagyobb kihívás az elkészítés során a projekt strukturálása és az adatok kezelése. Az adatok kezelésekor pedig oda kellett figyelni a hatékony API-hívásokra és a megfelelő adatkezelési technikákra. További nehézség volt még hogy csapatban dolgoztunk, és össze kellett hangolnunk a munkánkat, és ki segíteni egy mást ha valami nem működött.

A vizsgaremek elkészítése során rájöttünk, hogy a folyamatos tanulás és a kihívásokkal való szembesülés elengedhetetlenek a szakmai fejlődéshez. A projektnek köszönhetően felismertük a saját erősségeinket és gyengeségeinket, és megtanultuk hatékonyan kezelni a nehézségeket.

A jövőbeni célok tekintetében, szeretnénk tovább fejlődni e téren. Továbbá, terveink között szerepel egy valós projekten való részvétel, amely lehetőséget ad arra, hogy az eddig megszerzett tudást gyakorlatban alkalmazzuk, és új kihívásokkal szembesülünk. És így a végén visszatekintve csak annyit mondunk, hogy élveztük a projektmunka elkészítését.

# Vizsgaremek

Készítették: Benkő Zsolt Tamás,  
Libai Kristóf

**Pécs**

**2023**