



**Eötvös Loránd Tudományegyetem**

**Informatikai Kar**

**Informatikatudományi Intézet**

**Média- és Oktatásinformatika Tanszék**

# Algoritmusok és adatszerkezetek, illetve C nyelvben történő memóriakezelés témakörök fogalmainak játékosítása

**Szerző:**

Oláh Norbert

Programtervező informatikus BSc.

**Témavezető:**

Menyhárt László Gábor Dr.

adjunktus, PHD

**Budapest, 2024**

# EÖTVÖS LORÁND TUDOMÁNYEGYETEM

## INFORMATIKAI KAR

### SZAKDOLGOZAT TÉMABEJELENTŐ

**Hallgató adatai:**

**Név:** Oláh Norbert

**Neptun kód:** PST8RA

**Képzési adatok:**

**Szak:** programtervező informatikus, alapképzés (BA/BSc/BProf)

**Tagozat:** Nappali

Belső témavezetővel rendelkezem

**Témavezető neve:** Dr. Menyhárt László Gábor

munkahelyének neve, tanszéke: **ELTE IK, Média- és Oktatásinformatika Tanszék**

munkahelyének címe: **1117, Budapest, Pázmány Péter sétány 1/C.**

beosztás és iskolai végzettsége: **adjunktus, PHD**

**A szakdolgozat címe:** Algoritmusok és adatszerkezetek, illetve C nyelvben történő memóriakezelés témakörök fogalmainak játékosítása

**A szakdolgozat témája:**

(A témavezetővel konzultálva adja meg 1/2 - 1 oldal terjedelemben szakdolgozat témájának leírását)

A szakdolgozat témája az algoritmusok és adatszerkezetek témakörben foglalt fogalmak, illetve a C nyelvben történő memóriakezelésnek a játékosítását foglalja magában egy grafikus program formájában.

A játékosítandó témák az algoritmusok és adatszerkezetek témaköréből:

-Elemi adatszerkezetek: Tömbök, Verem, Sorok, Listák, Fák, Hasító táblák, Gráfok

-Rendezések: Beszűrő rendezés, Kiválasztó rendezés, Összefésülő rendezés, Gyorsrendezés, Kupacrendezés, Radix rendezés, Leszámláló rendezés, Edényrendezés

-Gráfok: Gráfok ábrázolásai, Szélességi bejárás, Mélységi bejárás,

Legrövidebb út probléma algoritmusai

A játékosítandó témák a C nyelv témaköréből:

-Memóriakezelő függvények (malloc(), realloc(), calloc(), free()) működésének szemléltetése

-Pointerek működésének szemléltetése

A játékosítás megvalósítása különböző módszerekkel történik, először is a felhasználók által megtanulandó száraz, tömör tananyagot átírjuk kisebb, egyszerűbben emészthető részekre.

Ezután interaktív animációkkal illusztráljuk az algoritmusok működését, kódpéldákkal illusztrálva különböző nyelvekben (C/C++, Java, Python).

A játékosítást elősegítő szint- és pontrendszer is a felhasználók segítségére lesz a tananyag megértésében.

Különböző feladatok és/vagy kvízek helyes megoldásával és kitöltésével a felhasználók pontokat gyűjthetnek és szintet léphetnek, ezzel is elősegítve az egymás közötti versengést és ezáltal a játékosítást.

Budapest, 2023. 12. 05.

# Tartalomjegyzék

1.	Bevezetés .....	1
1.1.	A szakdolgozat témája, motiváció.....	1
1.2.	A szakdolgozat felépítése .....	1
2.	Felhasználói dokumentáció.....	3
2.1.	Bevezetés, fontos információk.....	3
2.2.	A weboldal használata.....	3
2.3.	Oldaltérkép.....	4
2.4.	Oldalak felépítése, használata.....	5
2.4.1.	Navigációs sáv, lábléc .....	5
2.4.2.	Bejelentkezés.....	6
2.4.3.	Elfelejtett jelszó, jelszó visszaállítása .....	7
2.4.4.	Regisztráció .....	9
2.4.5.	E-mail cím megerősítése .....	10
2.4.6.	Vezérlőpult .....	12
2.4.7.	Profil (saját) .....	13
2.4.8.	Közösség .....	15
2.4.9.	Profil (más felhasználó) .....	18
2.4.10.	Tananyag .....	19
2.4.11.	Feladat .....	20
2.4.12.	Feladat eredménye .....	21
2.4.13.	Kijelentkezés.....	23
3.	Fejlesztői dokumentáció .....	24
3.1.	Megvalósíthatóság .....	24
3.2.	Követelmények.....	24
3.3.	Implementáció .....	25

3.3.1.	A probléma megközelítése .....	25
3.3.2.	Felhasznált technológiák, stack.....	26
3.3.3.	Tervezés.....	27
3.3.4.	A View bemutatása .....	36
3.3.5.	Az adatbázis felépítése .....	39
3.3.6.	A Model bemutatása .....	41
3.3.7.	A Controller bemutatása .....	44
3.3.8.	A backend egyéb részeinek bemutatása.....	45
3.4.	Tesztelés .....	47
3.4.1.	Egység tesztek .....	47
3.4.2.	Felhasználói felület tesztelése.....	53
3.5.	Üzembe helyezés, karbantartás .....	53
3.5.1.	Konfigurálás.....	53
4.	Összefoglalás .....	55
4.1.	Skálázhatóság .....	55
4.2.	További fejlesztési lehetőségek .....	55
5.	Irodalomjegyzék .....	56
5.1.	Felhasznált dokumentációk.....	56
5.2.	Felhasznált irodalom .....	56
6.	Ábrajegyzék .....	60
7.	Felhasznált technológiák.....	62
7.1.	Backend .....	62
7.1.1.	Production .....	62
7.1.2.	Production és development .....	62
7.2.	Frontend .....	63
7.2.1.	Production .....	63

7.2.2.	Production és development .....	64
7.3.	Felhasznált programok, IDE-k.....	65

# 1. Bevezetés

## 1.1. A szakdolgozat témája, motiváció

Egyetemi éveim során többször is hallottam szaktársaimtól, hogy nehézségeket okoz nekik az algoritmusok működésének elsajátítása, illetve implementációjuk bizonyos nyelvekben. Elgondolkodtam, hogy hogyan is lehetne ezeket az ismereteket egy helyre gyűjteni és egy olyan formában visszaadni, ami érdekes lehetne a hallgatóknak. Ezen a gondolatmeneten továbbmenve döntöttem úgy, hogy egy játékosított, versengésre buzdító webapplikáció keretein belül lenne ezt a legérdekesebb megvalósítani, ami így a szakdolgozatom témája is lehetne.

Így készült el az [Algoritmizátor](#). A továbbiakban a webapplikációra így fogok hivatkozni. Az Algoritmizátor használata regisztrációhoz és bejelentkezéshez kötött, amihez inf-es e-mail cím szükséges. A szükséges autentikációt követően viszont a felhasználók szabadon barangolhatnak a weboldalon, barátkérelmeket küldhetnek egymásnak, böngészhetik a leckéket, pontokat gyűjthetnek a feladatok megoldásával és versenghetnek egymás között.

Minden tananyag illusztrálva van kódpéldákkal, még hozzá 4 féle nyelvben, ezek a nyelvek a C++, a Java, a Python és a JavaScript. Ezek is hozzájárulnak az ismeretek gyakorlati alkalmazásainak megértéséhez.

## 1.2. A szakdolgozat felépítése

A dolgozat 6 egységből, illetve azok alegységeiből áll. A felbontás a következő:

### 1. Egység: Jelen bevezető

Ebben a részben bemutatom a szakdolgozat témáját, illetve felépítését.

### 2. Egység: Felhasználói dokumentáció

Ebben a részben részletesen bemutatom a felhasználónak a program elérését, illetve a használatához szükséges követelményeket, továbbá szemléltetem az Algoritmizátor működését, funkcióit és az egyes oldalak használatát.

### 3. Egység: Fejlesztői dokumentáció

Ebben a részben bemutatom a megvalósításhoz szükséges erőforrásokat, illetve követelményeket, továbbá bővebb magyarázatot adok a program által használt megközelítésekre, technológiákra is. Ezután részletesen bemutatom a tervezési fázist és az implementációt is, kitérve a választott architektúra minden részére. Itt mutatom be a program tesztelését, a verziókezelőhöz kötött teszt pipeline-t, valamint az üzembe helyezéshez és a karbantartáshoz szükséges információkat.

### 4. Egység: Összefoglalás

Ebben a részben kitekintést nyújtok az Algoritmizátor jövőjére nézve, hogyan, mennyire skálázható, illetve milyen fejlesztési ötleteket lehetne még implementálni, amik a program jelen fejlesztési ciklusából kimaradtak.

### 5. Egység: Irodalomjegyzék

Ebben a részben közzéteszem az általam felhasznált programokat, dokumentációkat és szakirodalmakat, amiket a fejlesztés során felhasználtam.

### 6. Egység: Ábrajegyzék

Ebben a részben találhatóak meg az általam használt ábrák, mind egy helyen.

### 7. Egység: Felhasznált technológiák

Ebben a részben sorolom fel az Algoritmizátor által felhasznált technológiákat, verziójukkal együtt.

## 2. Felhasználói dokumentáció

### 2.1. Bevezetés, fontos információk

Az Algoritmizátort a [jxn.ddns.net/algoritmizator](http://jxn.ddns.net/algoritmizator) linken keresztül lehet elérni. Webapplikáció lévén a használatához JavaScript-et futtatni képes webböngésző szükséges, viszont fontos megjegyezni, hogy a mobilokon való megjelenítés csak korlátozottan elérhető.

### 2.2. A weboldal használata

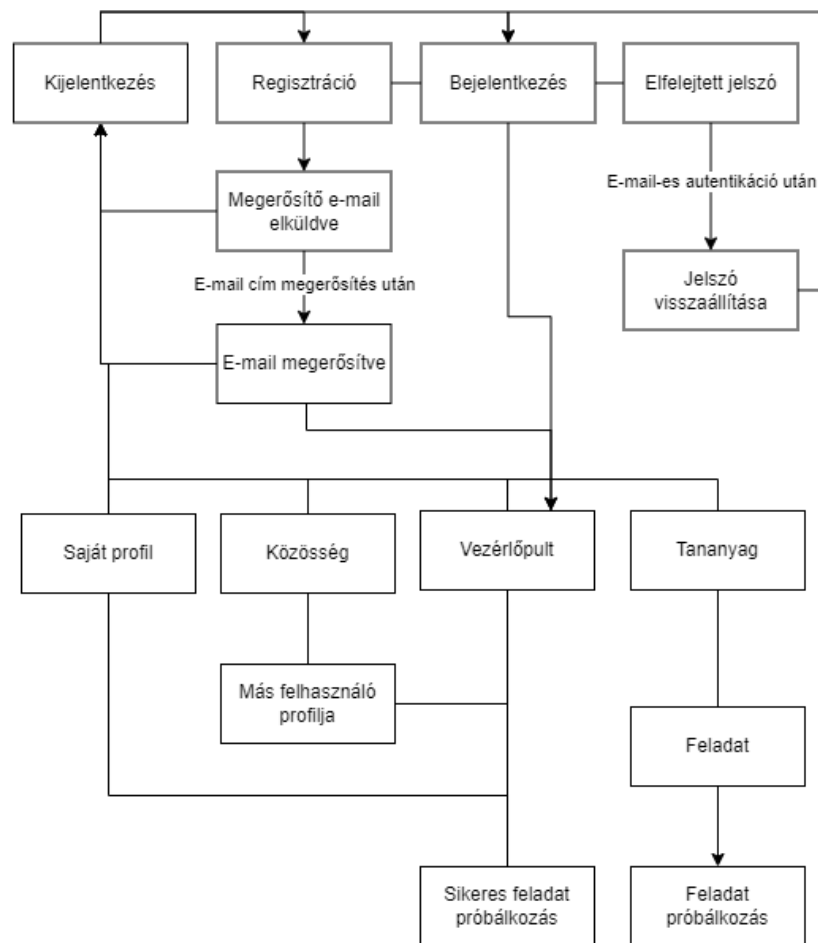
A weboldal használatához először regisztrálni kell a rendszerbe, ehhez meg kell adni egy tetszőleges nevet, felhasználónevet, egy inf-es e-mail címet, illetve egy kellően erős jelszót. Ezután a felhasználó kap egy megerősítő e-mail a megadott inf-es e-mail címére, ahol a megerősítés gombra kattintva igazolja a személyazonosságát. A gombra kattintás kivételével a folyamat teljesen automatikus. A hitelesítést követően a felhasználó átirányítódik a vezérlőpultra.

Fontos megjegyezni, hogy bár a regisztráció sikeres az e-mail cím megerősítése nélkül is, az oldal funkcionalitása csak megerősítés után válik elérhetővé. Ez azt jelenti, hogy a kijelentkezésen kívül minden oldal vissza fogja irányítani a felhasználót az e-mail megerősítése oldalra.

Abban az esetben, ha az e-mail elveszlik, vagy lejár, van lehetőség az újraküldésére, a fentebb említett oldalon keresztül.



## 2.3. Oldaltérkép



1. ábra - Oldaltérkép

A weboldalon történő navigálás az oldaltérkép szerint történik (lásd 1. ábra - Oldaltérkép). Itt a térképet érdemes két részre bontani, egy bejelentkezés előtti és egy utáni részre.

Az első részen csak a Bejelentkezés, Regisztráció és Elfelejtett jelszó oldalak elérhetőek. Ezek között az oldalak között szabad a navigáció. Sikeres bejelentkezés vagy regisztráció és e-mail megerősítés után értelemszerűen a felhasználó átkerül a második részre. A második részen érhetőek el az Algoritmizátor lényegi oldalai, a Vezérőpult, a Tananyag, a Közösség, a Saját profil és az ezekből leágazó oldalak.

Fontos megjegyezni, hogy a két rész között nincs átjárás, tehát például a Bejelentkezés oldal nem tekinthető meg kijelentkezés nélkül és a Vezérőpult sem tekinthető meg bejelentkezés nélkül.

## 2.4. Oldalak felépítése, használata

### 2.4.1. Navigációs sáv, lábléc

Még mielőtt az egyes oldalak felépítését bemutatnám, fontosnak tartom kiemelni, hogy minden oldalon található navigációs sáv és lábléc. A lábléc minden oldalon ugyanaz, és nem is szolgál nagy jelentőséggel (2. ábra - Lábléc).



2. ábra - Lábléc

A navigációs sávon találhatóak meg bejelentkezés után a főbb oldalakhoz tartozó linkek, a kijelentkezés, illetve a profil egy kevés információval (3. ábra), bejelentkezés előtt pedig maga a bejelentkezési oldal és a regisztrációs oldal linkje található meg (4. ábra).



3. ábra - Navigációs sáv bejelentkezés előtt



4. ábra - Navigációs sáv bejelentkezés után

### 2.4.2. Bejelentkezés

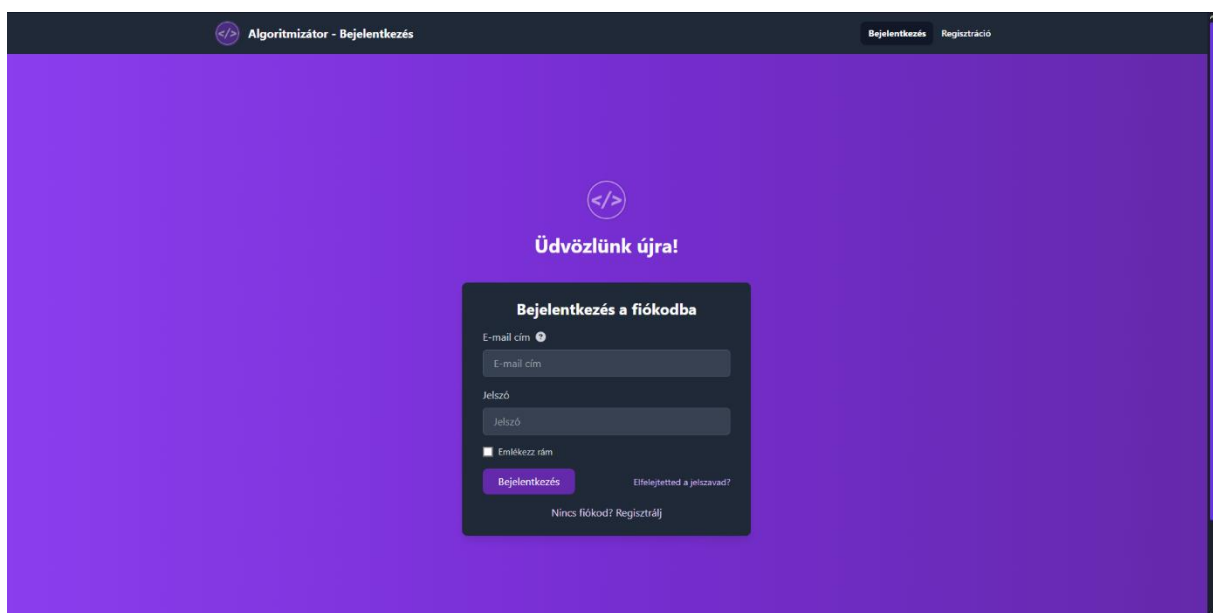
A bejelentkezés oldalon két szövegdozsoz található, egy az e-mail címnek és egy a jelszónak, továbbá egy 'Emlékezz rám' jelölőnégyzet. Ezek kitöltésével majd a 'Bejelentkezés' gombra való kattintással lehet bejelentkezni az Algoritmizátorba. Az 'Emlékezz rám' jelölőnégyzet bepipálásával a rendszer emlékezni fog a felhasználóra, ami annyit jelent, hogy amíg ki nem jelentkezik, addig akárhányszor megnyitja az oldalt, mindig automatikusan be fog jelentkezni.

A sikertelen bejelentkezési kísérletet az oldal felugró ablakkal jelzi, illetve többszöri sikertelen kísérlet után az oldal alján is megjelenik egy segítő szöveg. Sikeres bejelentkezés esetén az oldal átirányít a Vezérlőpult oldalra.

Található még az oldalon egy 'Elfelejtetted a jelszavad?' link, erre kattintva lehet jelszó visszaállítást kérni.

Továbbá található még egy link az oldal legalján, ami a regisztrációs oldalra vezet át.

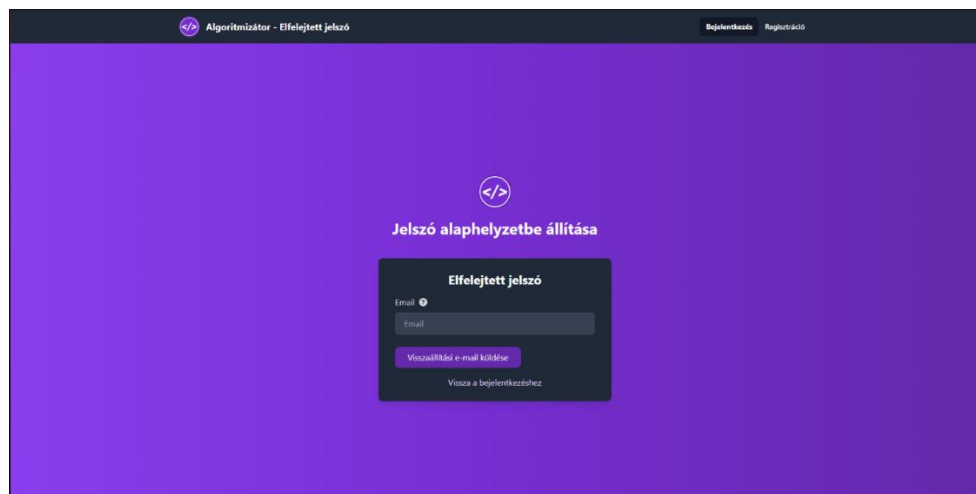
A bejelentkezési oldalt alább.



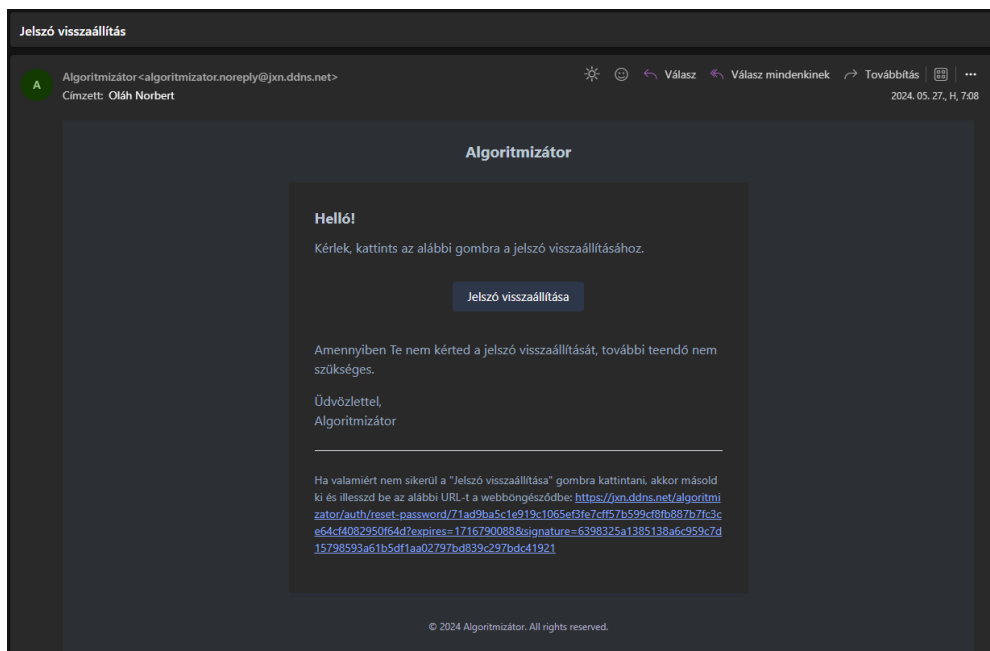
5. ábra - Bejelentkezés

### 2.4.3. Elfelejtett jelszó, jelszó visszaállítása

Az elfelejtett jelszó oldalon (lásd: 6. ábra) egy szövegdox található, amibe a felhasználó e-mail címe kerül. A 'Visszaállítási e-mail küldése' gombra kattintva - feltéve, hogy helyes az e-mail cím – egy e-mail lesz elküldve a megadott címre (lásd: 7. ábra), aminek a segítségével visszaállítható a felhasználó jelszava, továbbá az oldal értesíti a felhasználót az elküldés sikerességéről az oldal alján egy kis üzenet formájában. Ha a felhasználó egy nem regisztrált e-mail címet ad meg, akkor a rendszer egy felugró ablakban tájékoztat erről.



6. ábra - Elfelejtett jelszó



7. ábra - Jelszó visszaállítás e-mail

Az e-mailben lévő gombra kattintva a felhasználó a 'Jelszó visszaállítása' oldalra lesz átirányítva (lásd: 8. ábra). Ezen az oldalon három szövegdoboz található, egy az e-mail címnek biztonsági okokból, kettő pedig az új jelszónak, illetve annak megerősítésének. Az új jelszó megadásánál a rendszer valós időben jelzi a felhasználónak a jelszó erősségét. A jelszónak legalább 'Erős'-nek kell lennie. Erős jelszó és helyes e-mail cím esetén az oldal lent jelzi a visszaállítás sikerességét, ellenkező esetben vagy a szövegdobozok alatti részen jelzi a releváns hibát, vagy egy felugróablakban.

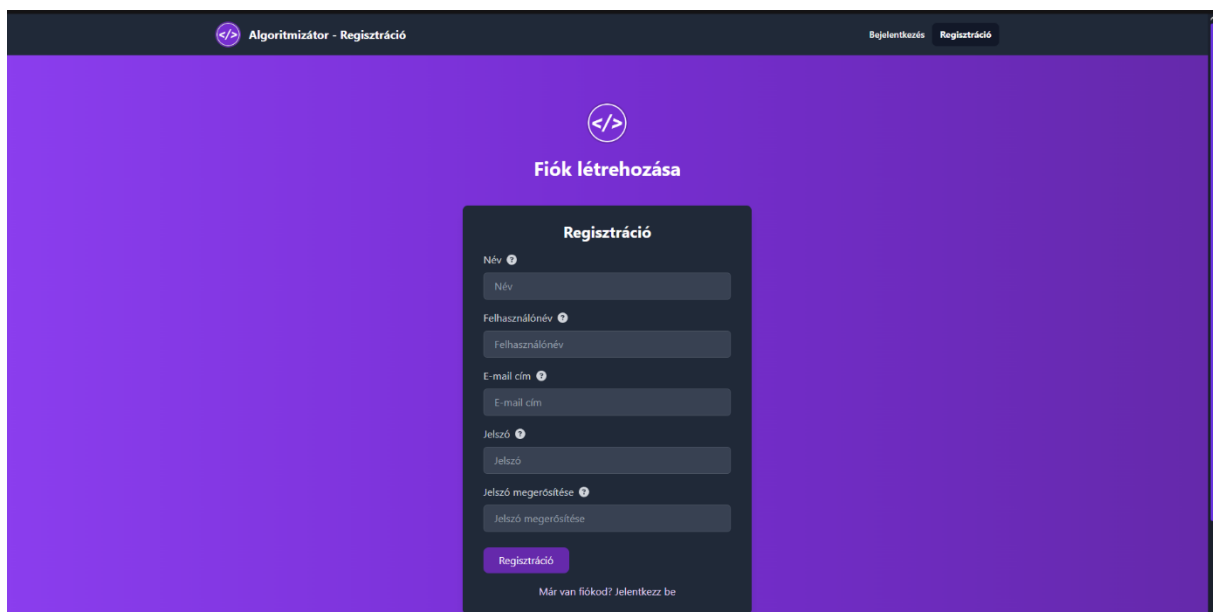
8. ábra - Jelszó visszaállítása

#### 2.4.4. Regisztráció

A regisztrációs oldalon (lásd: 9. ábra) öt szövegdoboz található, amikbe a felhasználó megadhatja a nevét, felhasználónevét, e-mail címét, illetve a jelszavát és annak megerősítését. Ezen mezők helyes kitöltésével az oldal felugró ablakban jelzi a regisztráció sikerességét, majd a felhasználó átirányítódik az 'E-mail cím megerősítése' oldalra és ezzel együtt be is jelentkezik a frissen létrehozott fiókjába, de az oldalt teljes értékűen majd csak megerősítés után használhatja.

Fontos megjegyezni, hogy a felhasználónévnek egyedinek kell lennie, illetve a jelszónak itt is legalább 'Erős'-nek kell lennie, valamint értelemszerűen nem lehet kétszer ugyanazzal az e-mail címmel regisztrálni. További kitétel, hogy az e-mail cím inf-es kell, hogy legyen, tehát inf.elte.hu-val kell, hogy végződjön.

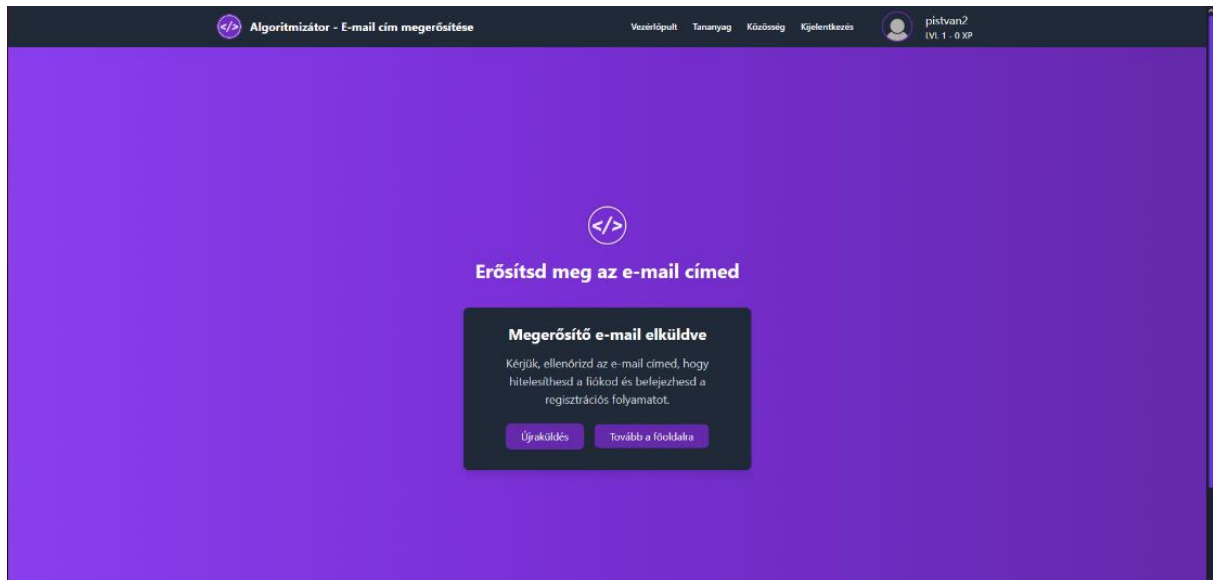
Ha a felhasználó bármilyen módon valahol helytelen adatokat ad meg, azt az oldal a megfelelő helyen, a szövegdobozok alatt, vagy felugró ablakban jelzi.



9. ábra - Regisztráció

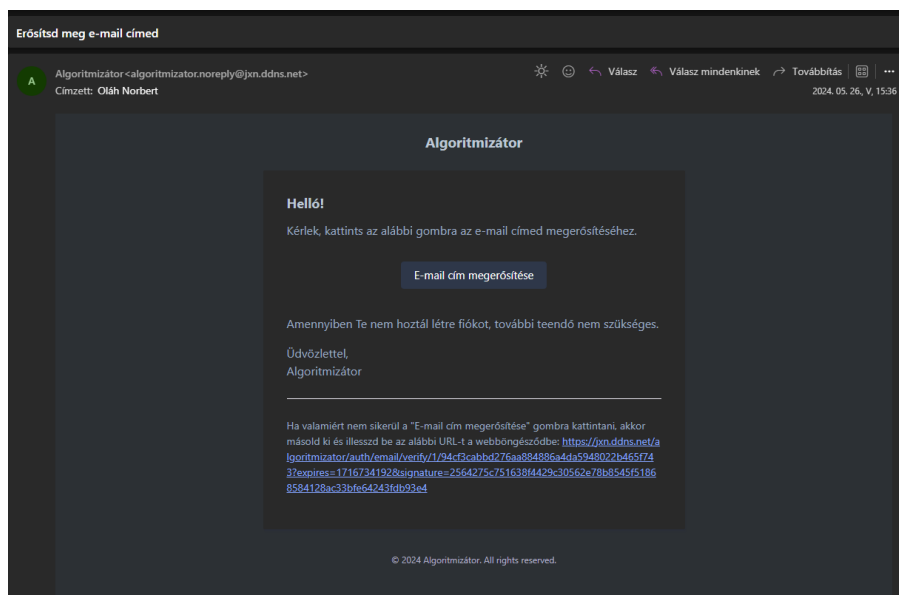
#### 2.4.5. E-mail cím megerősítése

Az 'E-mail cím megerősítése' oldalon (lásd: 10. ábra) egy tájékoztató üzenet jelenik meg a megerősítő e-mail elküldéséről, illetve lehetőség van ezen e-mail újraküldésére, ha az esetleg elveszett volna.

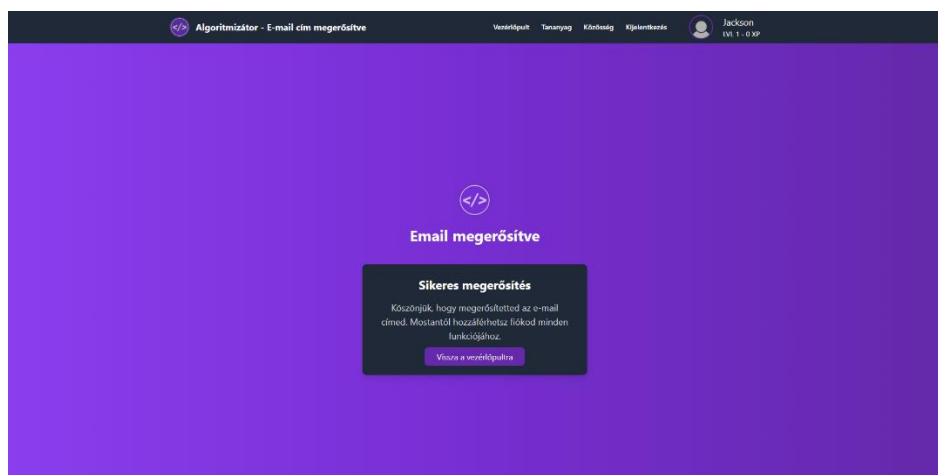


10. ábra - E-mail cím megerősítése

Az e-mailben lévő gombra kattintva (lásd: 11. ábra) a felhasználó automatikusan igazolja személyazonosságát és átirányítódik az 'E-mail cím megerősítve' oldalra (lásd: 12. ábra), ahol egy tájékoztató üzenet közli a megerősítés sikerességét. Ezután a felhasználó be van jelentkezve és elérhetővé válnak az Algoritmizátor további oldalai.



11. ábra - Megerősítési e-mail



12. ábra - E-mail cím megerősítve



#### 2.4.6. Vezérlőpult

Sikeres bejelentkezés és vagy regisztráció után a felhasználó a vezérlőpultra érkezik (lásd: 13. ábra). A középső részen egy üdvözlő üzenet alatt jelenik meg az aktuális óra leckéje. Itt óránként változik, hogy milyen lecke van éppen.

Jobb oldalt jelennek meg az éppen online lévő barátok. Itt minden barátnál meg lehet tekinteni az adott barát nevét, felhasználónevét, profilképét, szintjét és XP-jét (eXperience Point), valamint kattintással meg lehet nyitni a profilját.

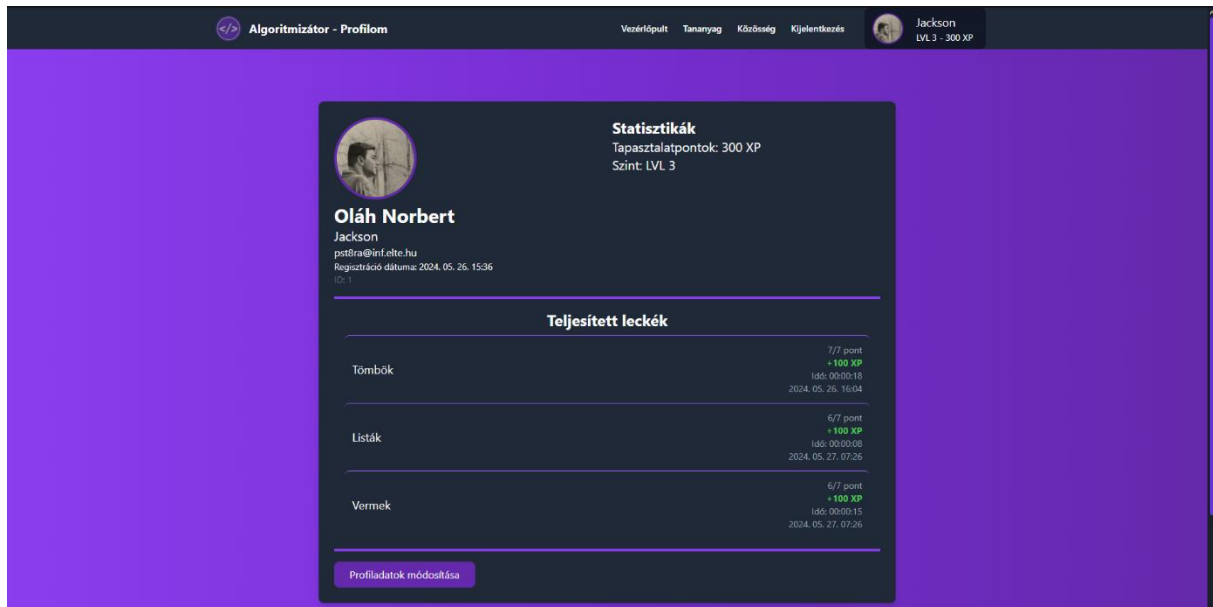
Bal oldalt az eddig sikeresen kitöltött feladatokat lehet megtekinteni. Minden feladatnál látható az adott feladat neve, a megszerzett pontszám, a megszerzett XP, a kitöltéshez szükséges idő és a kitöltés dátuma. Mivel egy-egy feladatot többször is ki lehet tölteni, így itt egy adott feladathoz mindig a legjobban sikerült próbálkozás fog megjelenni. A feladatokra való kattintással meg lehet nyitni az adott próbálkozás oldalát, ahol megtekinthetők a feladat kérdései, illetve az azokra adott válaszok is.



13. ábra - Vezérlőpult

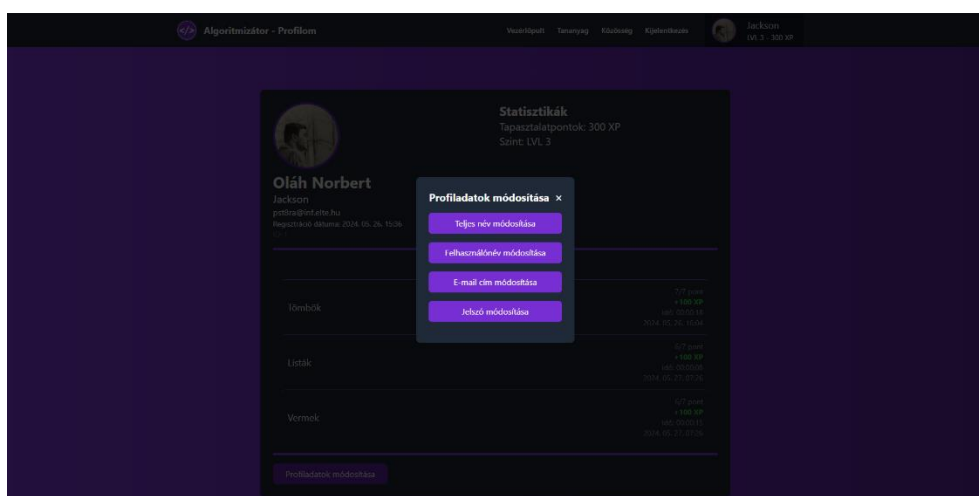
#### 2.4.7. Profil (saját)

A profil oldalon meg lehet tekinteni a felhasználó profilképét, nevét, felhasználónevét, e-mail címét, regisztrációjának dátumát, eddig összegyűjtött XP-jét és a szintjét, valamint az eddig teljesített feladatokat (lásd: 14. ábra). Itt is ugyanúgy meg lehet nyitni a próbálkozásokat, mint a vezérlőpulton.



14. ábra - Saját profil

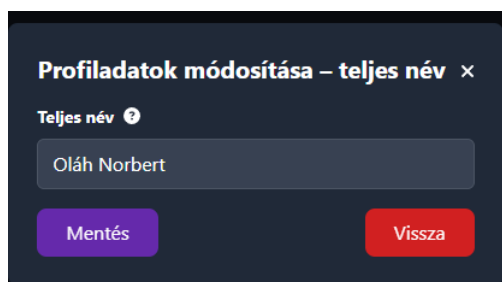
Azonban a profiladatok megtekintése mellett van lehetőség az adatok módosítására is a lent található 'Profiladatok módosítása' gombra segítségével. A gombra kattintva megjelenik egy felugró ablak, amin a felhasználó ki tudja választani, hogy milyen adatot szeretne módosítani (lásd: 15. ábr).



15. ábra - Profiladatok módosítása

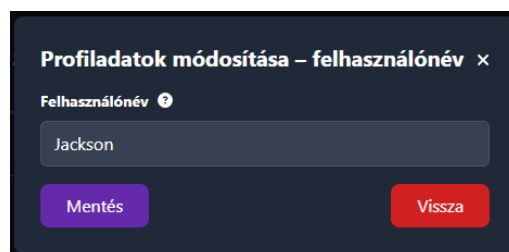
A megfelelő gombokra kattintva megjelennek a megfelelő felugró ablakok, ahol az eddig megszokott módon tudjuk az adatokat bevinni (lásd: 16. ábra, 17. ábra, 18. ábra, 19. ábra). Itt is, mint az eddigiekben a sikeres, illetve a sikertelen változtatásokról is tájékoztat az oldal, vagy a megfelelő szövegdobozok alján, vagy pedig felugró ablak segítségével.

Van lehetőség a profilkép módosítására is, ezt a profilképre kattintva lehet megtenni. Ekkor felugrik egy fájlkezelő ablak, ahol ki tudjuk választani a beállítani kívánt profilképet és fel tudjuk tölteni. Ha az megfelel a követelményeknek (maximum 2MB, .jpeg, .jpg, .png, .gif, .svg formátumok) akkor az új profilkép elmentésre kerül. Mind sikeres mind sikertelen feltöltés esetén az oldal felugró ablakban tájékoztatja a felhasználót.



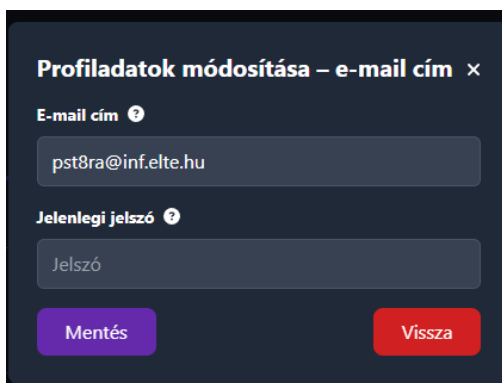
The screenshot shows a modal window titled "Profiladatok módosítása – teljes név" with a close button (X). It contains a label "Teljes név" with a help icon, a text input field containing "Oláh Norbert", and two buttons at the bottom: "Mentés" (purple) and "Vissza" (red).

16. ábra - Profiladatok módosítása, teljes név



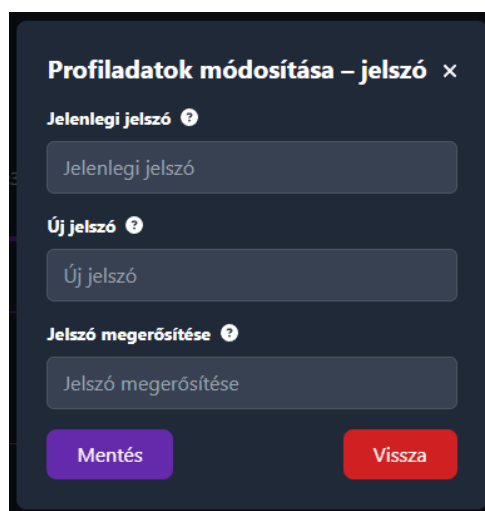
The screenshot shows a modal window titled "Profiladatok módosítása – felhasználónév" with a close button (X). It contains a label "Felhasználónév" with a help icon, a text input field containing "Jackson", and two buttons at the bottom: "Mentés" (purple) and "Vissza" (red).

18. ábra - Profiladatok módosítása, felhasználónév



The screenshot shows a modal window titled "Profiladatok módosítása – e-mail cím" with a close button (X). It contains two labels with help icons: "E-mail cím" and "Jelenlegi jelszó". The "E-mail cím" field contains "pst8ra@inf.elte.hu" and the "Jelenlegi jelszó" field contains "Jelszó". There are two buttons at the bottom: "Mentés" (purple) and "Vissza" (red).

17. ábra - Profiladatok módosítása, e-mail cím



The screenshot shows a modal window titled "Profiladatok módosítása – jelszó" with a close button (X). It contains three labels with help icons: "Jelenlegi jelszó", "Új jelszó", and "Jelszó megerősítése". The "Jelenlegi jelszó" field contains "Jelenlegi jelszó", the "Új jelszó" field contains "Új jelszó", and the "Jelszó megerősítése" field contains "Jelszó megerősítése". There are two buttons at the bottom: "Mentés" (purple) and "Vissza" (red).

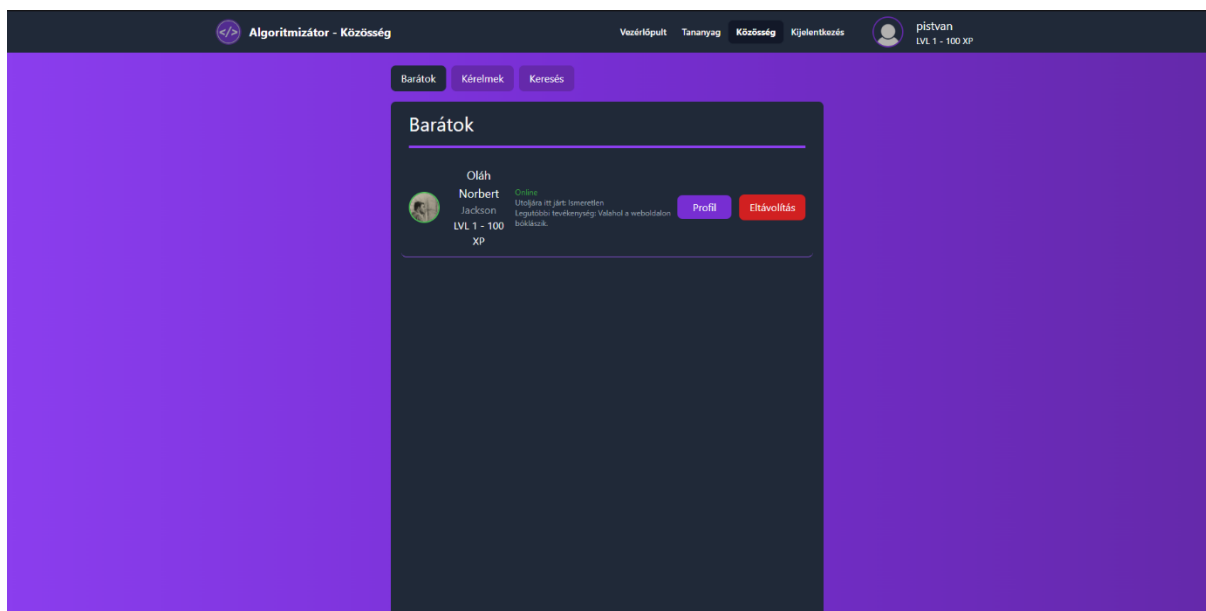
19. ábra - Profiladatok módosítása, jelszó

#### 2.4.8. Közösség

A közösség oldalon lehetőség nyílik böngészni a barátok között, küldeni, elfogadni, valamint elutasítani barátkérelmeket, illetve keresgélni a felhasználók között. Az oldal 3 részre van bontva, a részek között a megfelelő fülekre kattintva lehet navigálni.

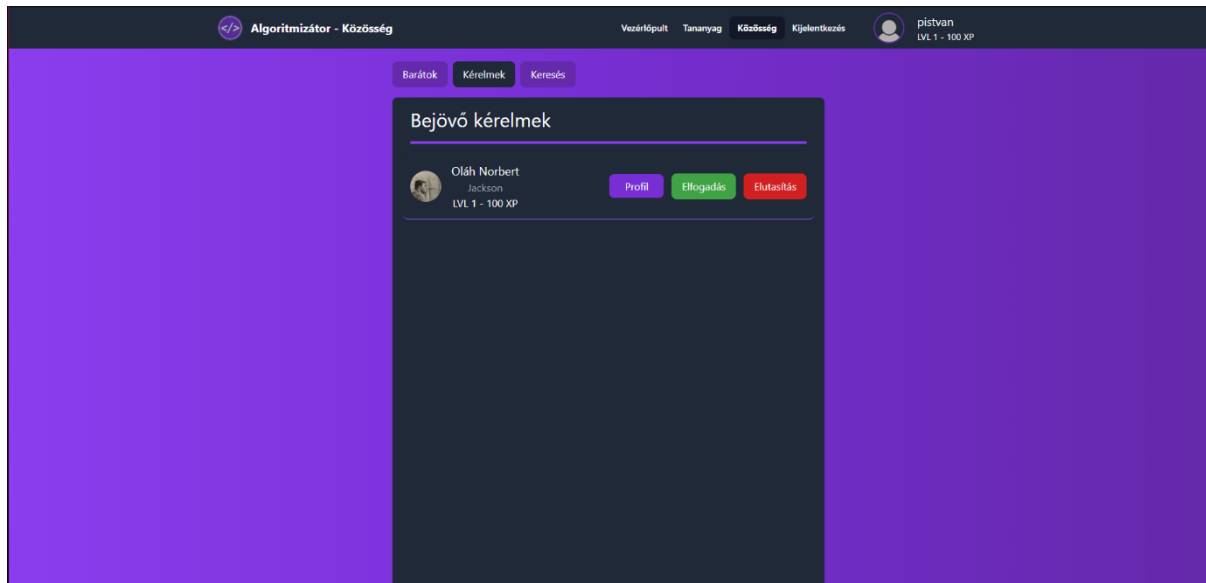
Az első fülön listázódnak ki a barátok (lásd: 20. ábra). Itt is meg lehet tekinteni azokat az adatokat, mint a vezérlőpulton, viszont itt nem csak az online barátok szerepelnek, illetve több információt is kapunk. Megtekinthetjük például, hogy legutóbb melyik oldalon jártak és mit csináltak, valamint mikor voltak legutóbb online vagy, hogy éppen aktívak-e.

Ezen az oldalon nyílik lehetőség egyes barátok esetleges eltávolítására is. Az oldal az eddigiekben megszokott módon felugró ablakban közli az eltávolítás sikerességét, illetve esetleges sikertelenségét is. A barátokhoz tartozó profil gombra kattintva megtekinthető az adott barát saját profilja is.



20. ábra – Közösség, barátok

A második fülön lehet megtekinteni a bejövő barátkérelmeket (lásd: 21. ábra). Az itt megjelenő felhasználóknak csak a profilképüket, nevüket, felhasználónevüket, illetve az eddig megszerzett szintjüket és XP-jüket lehet megtekinteni, azonban a profiljukat meg lehet nézni a profil gombra kattintva. Van lehetőség a barátkérelmek elfogadására és elutasítására a megfelelő gombokra kattintva, itt is felugró ablakban értesül a felhasználó a művelet sikerességéről.

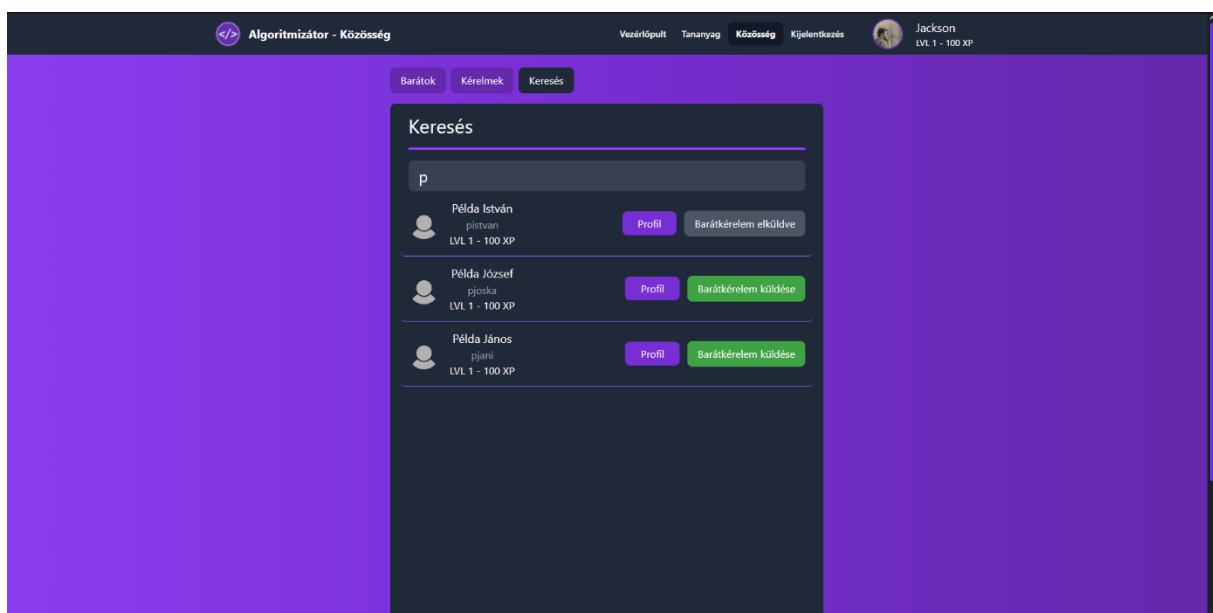


21. ábra - Közösség, barátkérelmek

A harmadik fülön egy keresősáv segítségével lehet keresgélni a felhasználók között (lásd: 22. ábra).

A keresés automatikus, minden leütött karakterrel frissül a találatok listája. Itt is hasonlóképpen jelennek meg a felhasználók, mint a második fülön, annyi kivétellel, hogy itt barátkérelem küldésére van lehetőség, nem annak elfogadására vagy elutasítására.

Egy-egy felhasználónak csak egyszerre egy barátkérelmet lehet küldeni, azon felhasználóknak, akiknek már lett küldve barátkérelem, de még nem fogadták el, a küldés gomb zöld helyett szürkére vált és nem-kattinthatóvá válik.

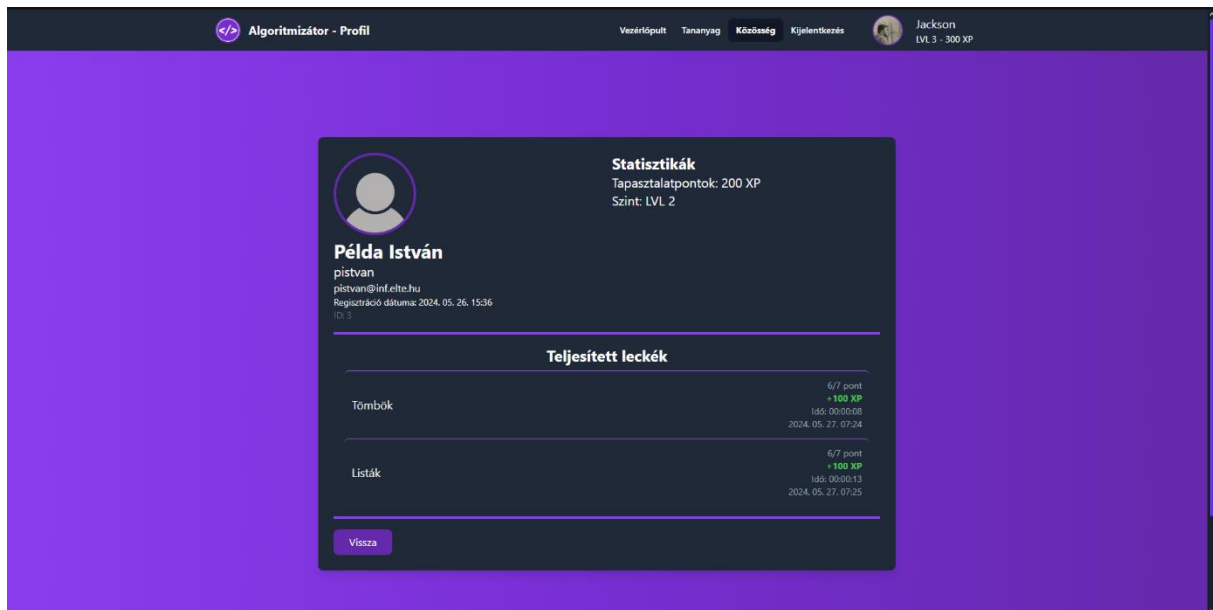


22. ábra - Közösség, keresés

#### 2.4.9. Profil (más felhasználó)

Ezen az oldalon jelenik meg más felhasználók profilja, akár a vezérlőpultról, akár a közösség oldalról lett megnyitva (lásd: 23. ábra).

Az oldal annyiban különbözik a saját profiltól, hogy itt értelemszerűen nem lehet más felhasználó adatait megváltoztatni, illetve a teljesített leckéket nem lehet részleteiben megtekinteni.



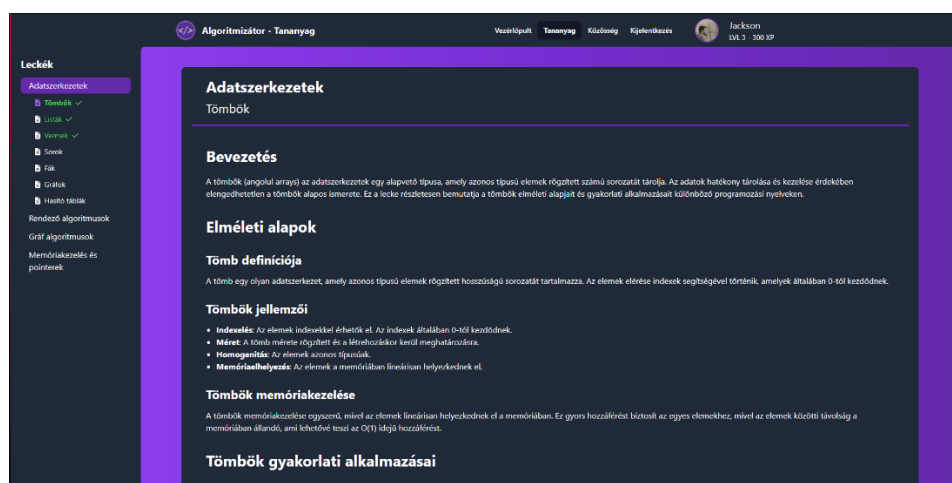
23. ábra - Profil, más felhasználó

## 2.4.10. Tananyag

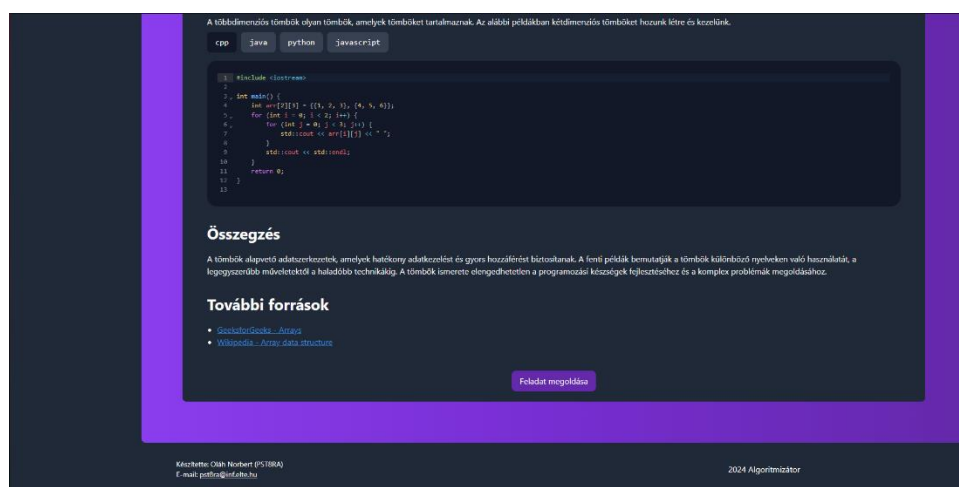
A tananyag oldalon lehet megtekinteni a leckéket (lásd: 24. ábra). Minden lecke egy-egy nagyobb témakörbe van fogva, amik között a jobb oldalt található navigációs sávon lehet navigálni. Azon leckék mellett, amikhez tartozik feladat egy kvíz ikon található, illetve már sikeresen megoldott feladatok esetén a leckék zölddel jelennek meg.

Minden leckében található több olyan egység, amiben kódrészlet található. Ezek interaktív és szerkeszthető kódszerkesztők, amikben szabadon lehet kísérletezni az adott kóddal. A kódszerkesztők fölött lévő fülek segítségével pedig lehet váltogatni, hogy az adott algoritmus milyen nyelvű implementációját szeretnénk megtekinteni vagy szerkesztgetni.

Azon leckék alján, amikhez tartozik feladat, található egy gomb, ami segítségével az adott feladat elkezdhető (lásd: 25. ábra).



24. ábra - Tananyag



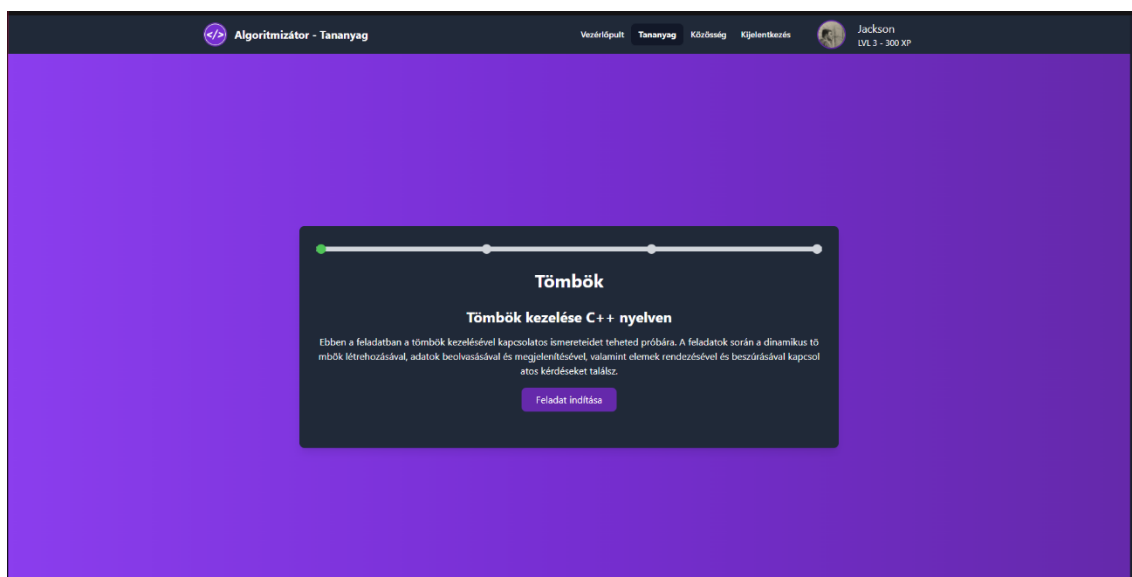
25. ábra – Tananyag, kódszerkesztő és feladat gomb



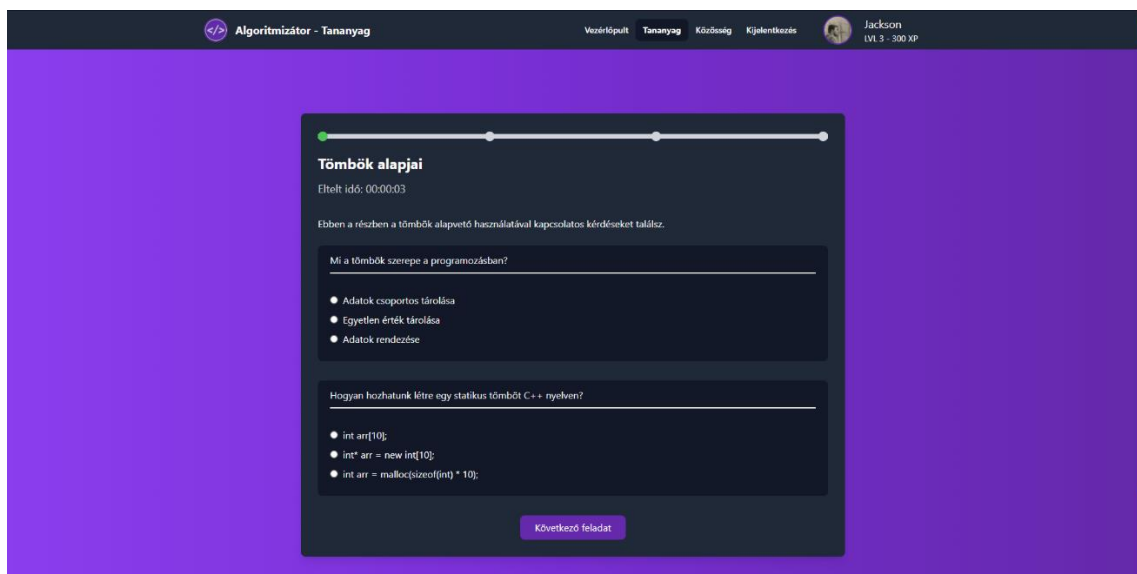
### 2.4.11. Feladat

Az adott feladat oldalára érkezve először a feladat neve, illetve egy eligazító szöveg található, ami ismerteti a feladatot (lásd: 26. ábra). Az ez alatt lévő 'Feladat indítása' gombra kattintva tudjuk elkezdni a kitöltést.

Ekkor elindul egy óra és meg is jelennek az első kérdések, illetve választható opciók (lásd: 27. ábra). Felül látható egy előrehaladás-jelző, ezen láthatjuk, hogy hol járunk az adott részfeladat kitöltésében, mennyi van még hátra.



26. ábra - Feladat elindítása



27. ábra – Feladat, első kérdések

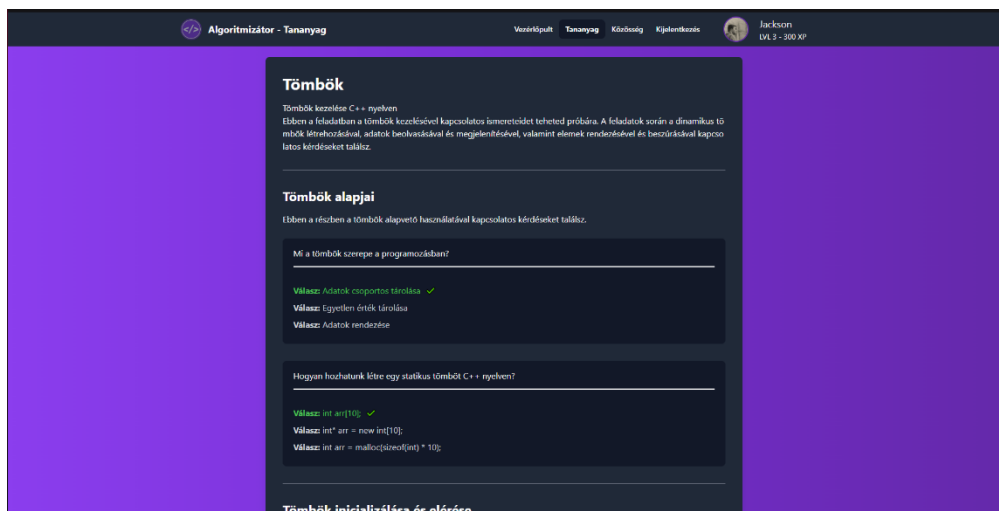
Egy részfeladatról a következőre csak akkor lehet váltani, ha minden kérdés meg lett válaszolva, ellenkező esetben az oldal felugró ablakban közli ezt a felhasználóval. Ha minden kérdés meg lett válaszolva, akkor az oldal alján található gombbal lehet haladni a következő részfeladatra. Visszamenni előző részfeladatokra nem lehetséges.

Ha már nem maradt több részfeladat, akkor az oldal alján lévő gombra kattintva lehet a feladatot beadni. Ezt követően a felhasználó azonnal meg is tekintheti lepontozva a beadott feladatát.

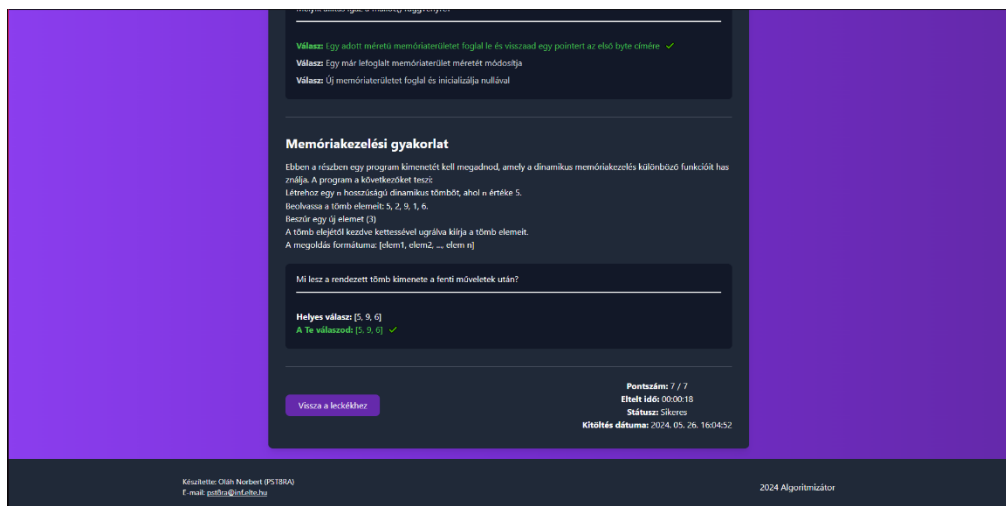
#### 2.4.12. Feladat eredménye

A beadott feladatot, illetve a vezérlőpultból, illetve a saját profilból megnyitott sikeres próbálkozásokat is ezen az oldalon lehet megtekinteni. Leadás után a felhasználó egyből át is irányítódik erre az oldalra.

Az oldal annyiban különbözik a feladat oldalától, hogy itt tömbösítve szerepelnek a részfeladatok, nem kell lapozgatni, valamint megtekinthetőek a helyes és a helytelen válaszok, illetve az összpontszám (lásd: 28. ábra, 29. ábra). A helyes válaszok zölddel, a helytelenek pirossal, míg a nem választott, de helyes válaszok sárgával jelennek meg. Az oldal alul jelzi a próbálkozás sikerességét vagy sikertelenségét. Sikeres kitöltés esetén a feladathoz tartozó XP jóváíródik a felhasználó profiljához. Fontos megjegyezni, hogy egy feladathoz tartozó XP csak egyszer íródik jóvá.



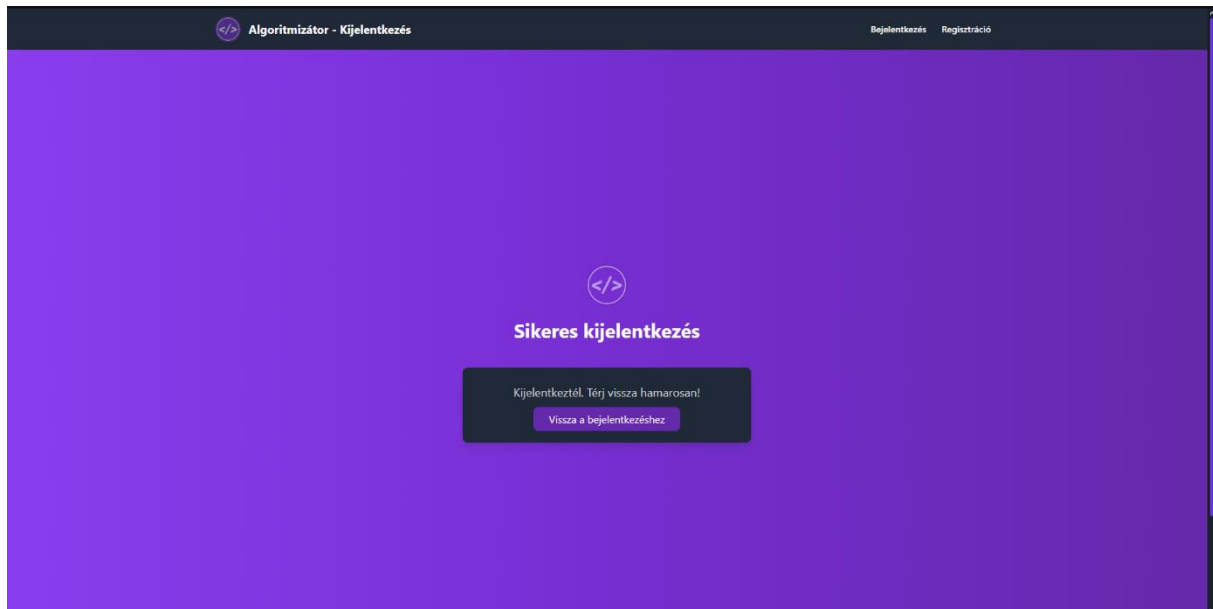
28. ábra - Feladat eredménye



29. ábra - Feladat eredménye, összpontszám, sikeresség

#### 2.4.13. Kijelentkezés

A navigációs sávon lévő Kijelentkezés gombra kattintva lehet kijelentkezni az Algoritmizátorból. A gombra kattintás után a felhasználó átirányítódik a kijelentkezés oldalra, ahol az oldal tájékoztatja a kijelentkezés sikerességéről (lásd: 30. ábra). Innen lehetőség nyílik újra bejelentkezni, vagy akár regisztrálni is.



30. ábra - Kijelentkezés

### 3. Fejlesztői dokumentáció

#### 3.1. Megvalósíthatóság

A projekt megvalósításához mindenképpen szükséges egy internetet elérő megfelelően konfigurált webszerver, adatbázisszerver és e-mail szerver, illetve egy fejlesztői számítógép, amin a megfelelő szoftverek telepítve vannak (erről bővebben a 3.3.2 számú fejezetben). Ezek akár lehetnek külön-külön felhő szolgáltatások keretein belül, vagy – ahogy az én esetemben – egy fizikai eszközön, egy otthoni szerveren. Nyilván a megvalósítás költsége függ a választott szolgáltatásoktól. Egy otthoni szerver üzemeltetése viszont viszonylag költséghatékony, ha elérhető egy felhasználható számítógép számunkra. Mindösszesen egy Dynamic DNS szolgáltatásra van szükség (vagy bármilyen más módon statikus IP-címre), egy SSL tanúsítványra, illetve egy SMTP port feloldásra. Ezek éves költsége körülbelül 25.000 Ft.

#### 3.2. Követelmények

A programmal szemben támasztott követelmények a következők:

- Hatékonyság:
  - A weboldal erőforrásigénye minimális, az adatbázis tárhelyigénye a számított forgalom függvényében elhanyagolható. (<1 Gb)
- Megbízhatóság:
  - A weboldal futása gyors és akadálymentes, a rendszer terheltségétől nem függ.
- Biztonság:
  - A weboldal tárol érzékeny információt, email címeket és jelszavakat, de megfelelő titkosítást és biztonsági szabványokat alkalmazva az incidensek elkerülhetőek.
- Hordozhatóság:
  - A weboldal bármilyen webszerveren futtatható, ahol elérhető megfelelően telepített és konfigurált PHP és SQL környezet. Az email funkcionalitáshoz egy megfelelően konfigurált email szerver is szükséges.
- Felhasználhatóság:
  - A weboldal könnyedén elérhető bárholonnan egyetlen URL segítségével.
  - A weboldal felhasználóbarát, a felhasználói felület intuitív, könnyen kezelhető.
  - A felhasználói felület navigálását több helyen is beépített súgó segíti.

Az adatforgalommal szemben támasztott követelmények a következők:

- Minden adatforgalom a felhasználó és az adatbázis között validált, mind szerver, mind kliens oldalon.
- A szenzitív információk titkosítottak.
- A weboldal és a felhasználó védett a Cross Site Request Forgery (CSRF)<sup>1</sup> támadások ellen.

### 3.3. Implementáció

#### 3.3.1. A probléma megközelítése

##### 3.3.1.1. Megközelítés, architektúra

A probléma megoldásához leginkább felhasználóbarát és intuitív megközelítés, egy webapplikáció létrehozása. Igazából más megközelítés nem is nagyon jöhetne szóba egy ilyen probléma esetében, tekintve az igényt a felhasználók közötti versengésre, az adatok nyilvántartására, illetve a felhasználóbarát megjelenésre. Adja magát az MVC architektúra használata, amihez egy jó választás backend keretrendszernek a Laravel PHP keretrendszer.

##### 3.3.1.2. Backend

A Laravel ökoszisztémája ráadásul nagyon sok opciónak nyújt teret a fejlesztés során, autentikációtól kezdve, az e-mail küldésen keresztül, a munkamenet kezelésén át az adatbázis kezeléséig.

##### 3.3.1.3. Adatbázis

A Laravelbe épített Eloquent ORM Object Relational Mapping segítségével pedig az adatbázisműveletek zökkenőmentesen beépíthetőek a backendbe. Az adatbázis választásánál, figyelembe véve az egyszerű adattárolást, illetve a Laravel által nyújtott munkamenet menedzsmentet és egyéb szolgáltatásokat, a MySQL-re esett a választás.

---

<sup>1</sup> Bővebben a CSRF-ről itt: <https://owasp.org/www-community/attacks/csrf>

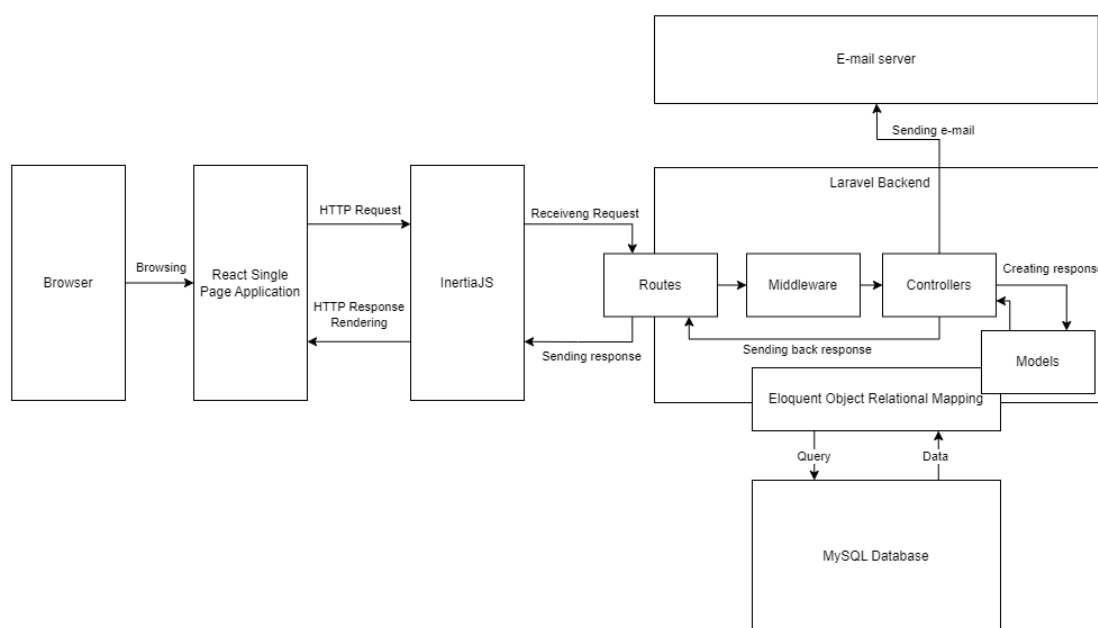
#### 3.3.1.4. Frontend

Megjelenítés tekintetében már több opció is szóba jöhet, egyrészt adná magát a Laravel beépített Blade template-jeinek használata, ami szerver oldali megjelenítés lévén egyszerűen meg tudná jeleníteni a backend felől érkező adatokat. Ennek a megközelítésnek a hátulütője a megfelelő testreszabhatóság hiánya, kevesebb opciónk van, mint mondjuk egy React.js, vagy egy Vue.js frontend keretrendszer esetében. Másrészt lehetne használni egy önálló frontend keretrendszert, viszont akkor meg kell oldani a backend és a frontend közötti kommunikációt. A választás végül az utóbbi megközelítésre esett, a React.js szolgáltatja a frontendet, a két oldal közötti kommunikációt pedig az Inertia végzi. Ezzel két legyet ütünk egy csapásra, a Reacttal nagyszerűen testre tudjuk szabni a weboldal kinézetét és viselkedését, míg a Laravel szolgáltatásait ugyanúgy tudjuk használni, mintha a Blade template-eket alkalmaznánk.

A CSS tekintetében a TailWindCSS keretrendszerre esett a választás, míg a kódszerkesztők a CodeMirror segítségével lettek megvalósítva.

#### 3.3.2. Felhasznált technológiák, stack

Az így előállt Laravel - React JS - MySQL stack végül minden igényt ki fog tudni szolgálni mind a fejlesztés során, mind pedig az üzemeltetés során a felhasználók számára. Az architektúra működését a következő ábra szemlélteti.



31. ábra - Komponens diagram

A projekt által felhasznált technológiák így a 7.3-as szekcióban megtekinthetők.

### 3.3.3. Tervezés

#### 3.3.3.1. Drótvázterv

Az alábbiakban bemutatom a webapplikáció felhasználói felületének drótváztervét. A megvalósított webapplikáció végleges felhasználói felülete néhány esetben eltérhet a drótváztervtől, de a lényegi elemek konzisztensek.

Főoldal / bejelentkezés

Algoritmizátor

**Algoritmizátor**  
Lorem ipsum  
dolor sit amet

**Bejelentkezés**  
Nincs még fiókod?  
[Regisztrálj](#)

Email-cím

Jelszó

[Efejeztetted a jelszavad?](#)

**Bejelentkezés**

Készletle: Oláh Norbert (PST8RA)

32. ábra - Drótvázterv, bejelentkezés

Főoldal / regisztráció

Algoritmizátor

**Algoritmizátor**  
Lorem ipsum  
dolor sit amet

**Regisztráció**  
Már van fiókod?  
[Bejelentkezz](#)

Felhasználónév

Email-cím

Jelszó

Jelszó még egyszer

**Regisztráció**


Készletle: Oláh Norbert (PST8RA)

33. ábra - Drótvázterv, regisztráció

Profil oldal

Algoritmizátor | [Leckék](#) | [Közösség](#) | **Profil**

**Profilom**

 **Példa név** **X. Szint / 2000 XP**  
**Leckék teljesítve: 13**

Felhasználónév: példapelda  
E-mail cím: pelda@pelda.com

**Felhasználónév megváltoztatása** **E-mail cím megváltoztatása** **Jelszó megváltoztatása**

Készletle: Oláh Norbert (PST8RA)

34. ábra - Drótvázterv, saját profil

Profil oldal / felhasználónév megváltoztatása

Algoritmizátor | [Leckék](#) | [Közösség](#) | **Profil**

**Felhasználónév megváltoztatása**

**Felhasználónév megváltoztatása**

Készletle: Oláh Norbert (PST8RA)

35. ábra - Drótvázterv, felhasználónév megváltoztatása



Profil oldal / e-mail cím megváltoztatása

Algoritmizátor    Leckék    Közösség    Profil 

### E-mail megváltoztatása

Új e-mail cím

Új e-mail cím még egyszer

**E-mail cím megváltoztatása**

Készítette: Oláh Norbert (PST8RA)

36. ábra - Drótvázterv, e-mail cím megváltoztatása

Profil oldal / jelszó megváltoztatása

Algoritmizátor    Leckék    Közösség    Profil 

### Jelszó megváltoztatása

Régi jelszó

Új jelszó


Új jelszó még egyszer

**Jelszó megváltoztatása**

Készítette: Oláh Norbert (PST8RA)


37. ábra - Drótvázterv, jelszó megváltoztatása

Közösség oldal / barátok kezelése

Algoritmizátor    Leckék    **Közösség**    Profil 

Barátok kezelése    Barátkérélmek    Keresés

Barát1  
Barát2  
Barát3

 Név **Barát eltávolítása**

Szint / XP **Tiltás**

Legutóbb online: ekkor


Teljesített leckék:

- Lecke 1
- Lecke 2
- Lecke 3
- Lecke 4

Készítette: Oláh Norbert (PST8RA)


38. ábra - Drótvázterv, közösség fül, barátok

Közösség oldal / barátkérélmek kezelése

Algoritmizátor    Leckék    **Közösség**    Profil 

Barátok kezelése    **Barátkérélmek**    Keresés

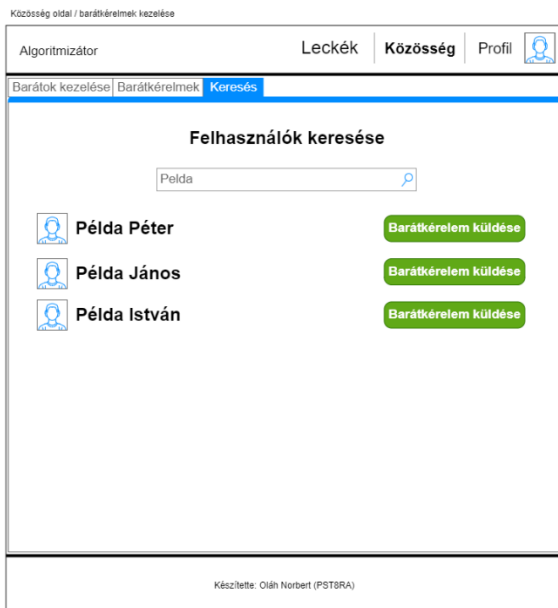
### Bejövő barátkérélmek

 Barátkérelm 1 **Elfogadás** **Elutasítás** **Tiltás**

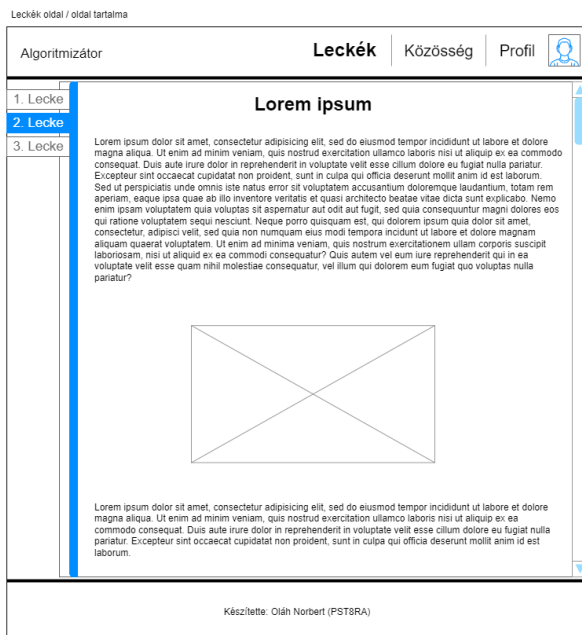
 Barátkérelm 2 **Elfogadás** **Elutasítás** **Tiltás**

Készítette: Oláh Norbert (PST8RA)

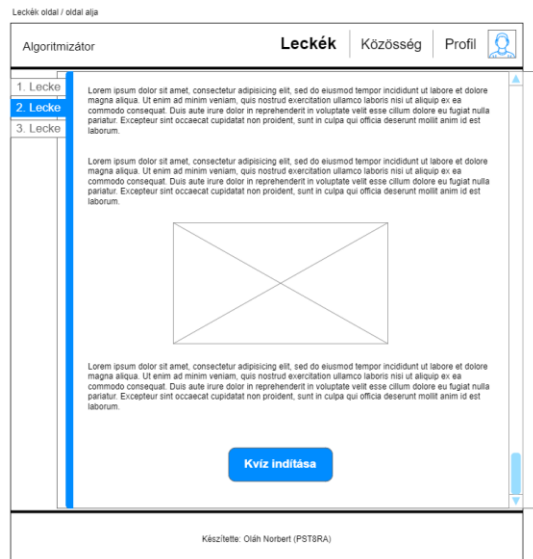
39. ábra - Drótvázterv, közösség fül, barátkérélmek



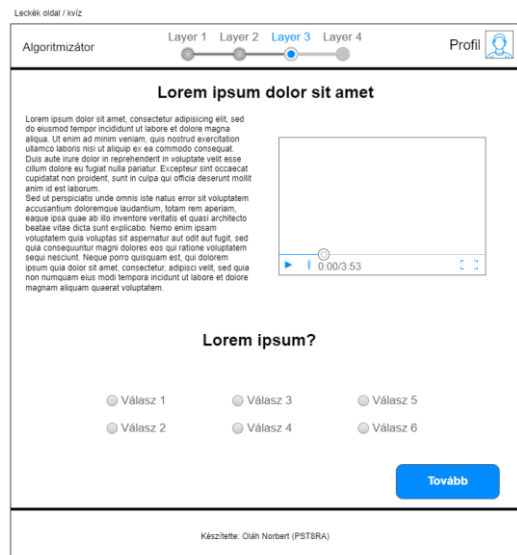
40. ábra - Drótvázterv, közösség fül, keresés



41. ábra - Drótvázterv, leckék fül



42. ábra - Drótvázterv, leckék fül, oldal alja



43. ábra - Drótvázterv, kvíz

Algoritmizátor

Profil

## Lorem ipsum dolor sit amet

### Eredmények

- Feladat 1: 4/6
- Feladat 2: 3/6
- Feladat 3: 6/6
- Feladat 4: 6/6
- Összesen: 19/24

A teszt sikeres!

Megszerzett XP: 100

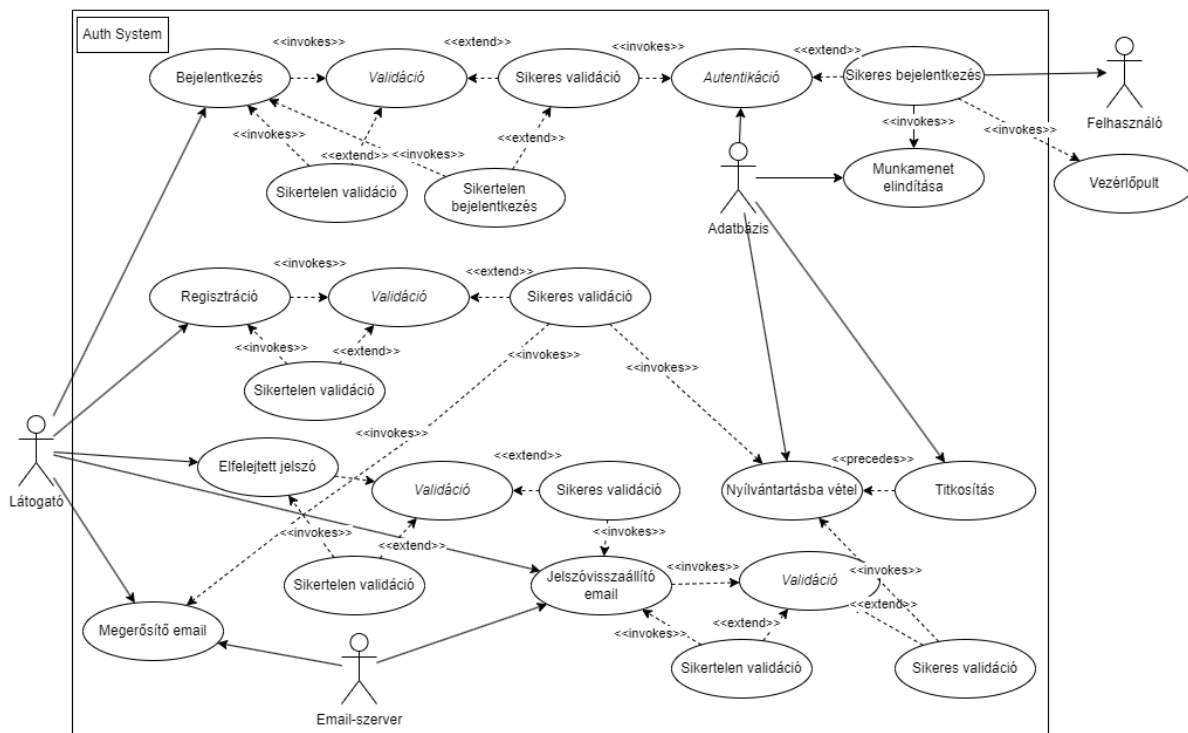
[Tovább a lecke oldalra](#)

Készítette: Oláh Norbert (PST8RA)

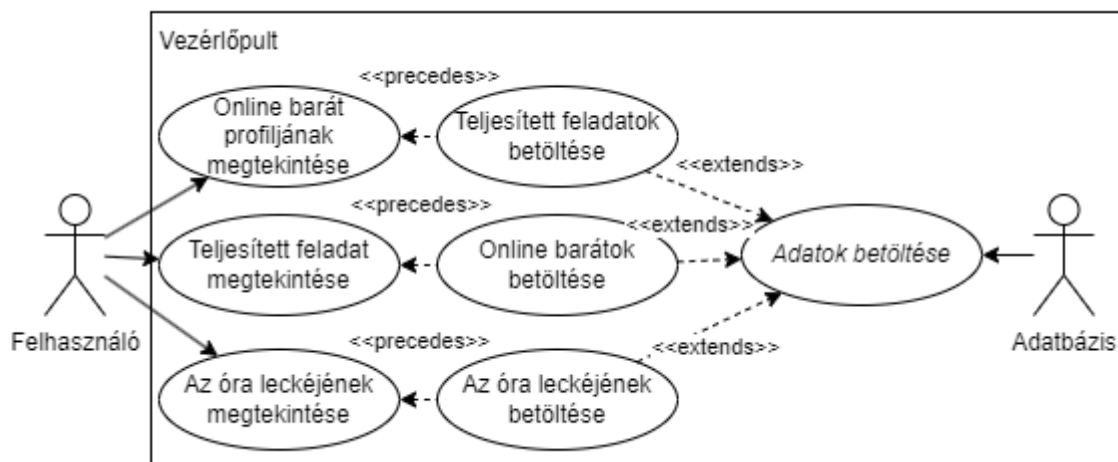
44. ábra - Drótvázterv, kvíz eredmény

### 3.3.3.2. Használati eset diagramok

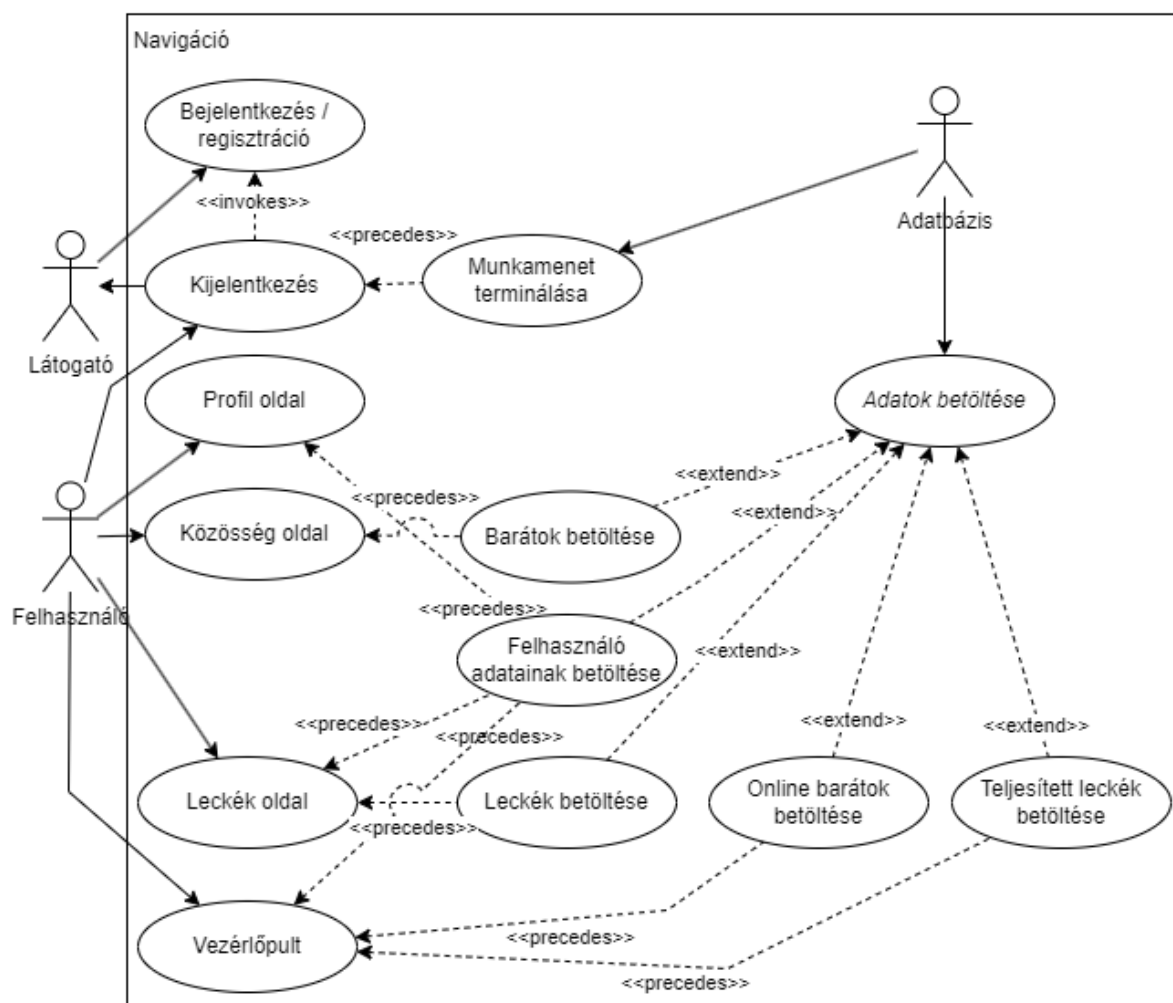
Az alábbiakban bemutatom a webapplikációhoz készített használati eset diagramokat a különböző modulokhoz.



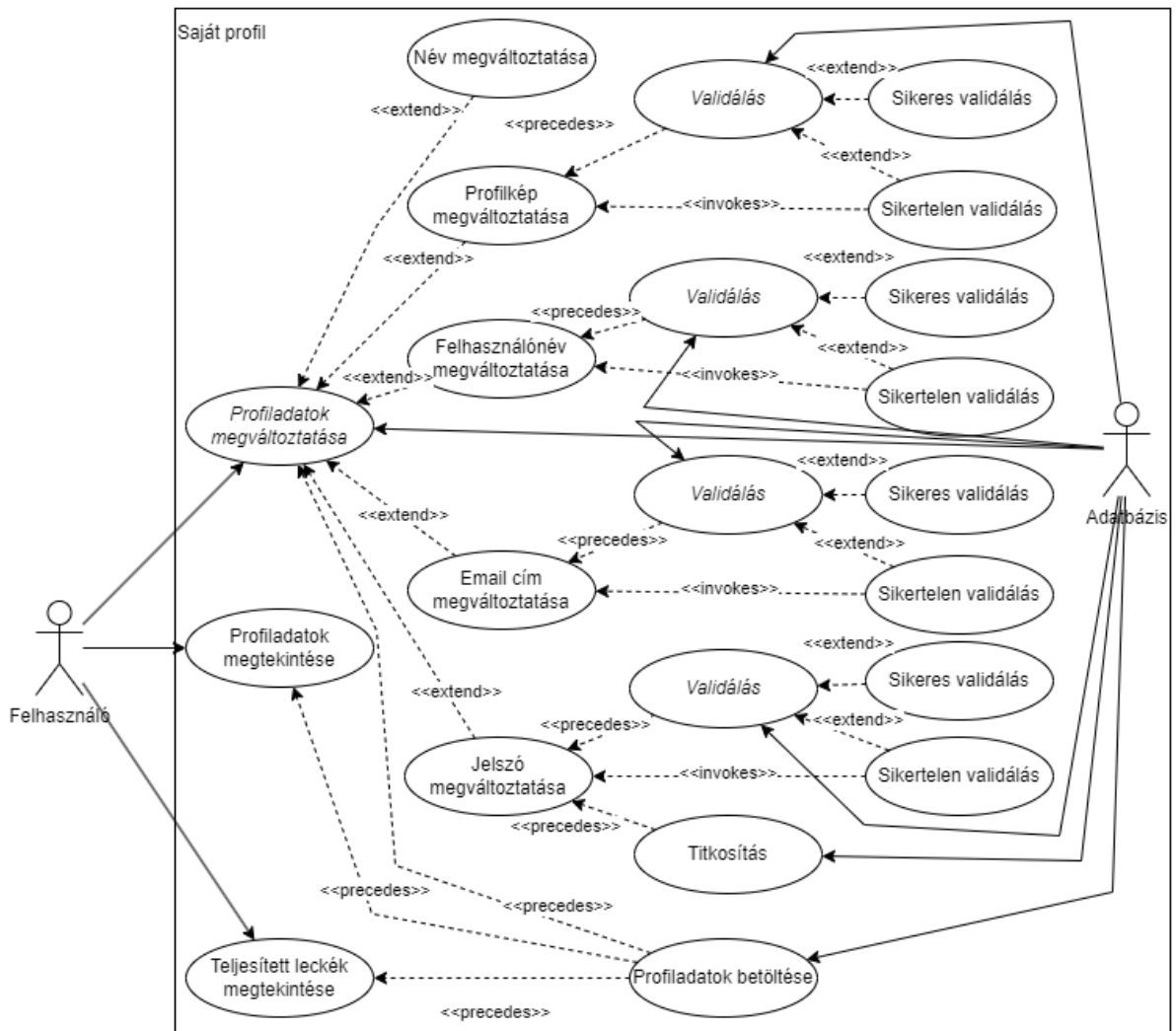
45. ábra - Használati eset diagram, autentikáció



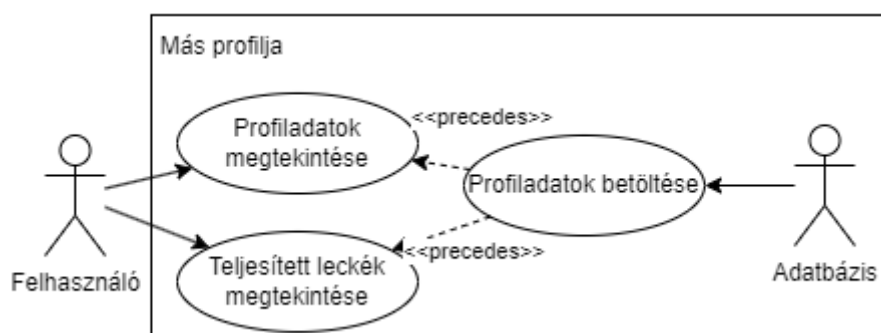
46. ábra - Használati eset diagram, vezérlőpult



47. ábra - Használati eset diagram, navigációs sáv

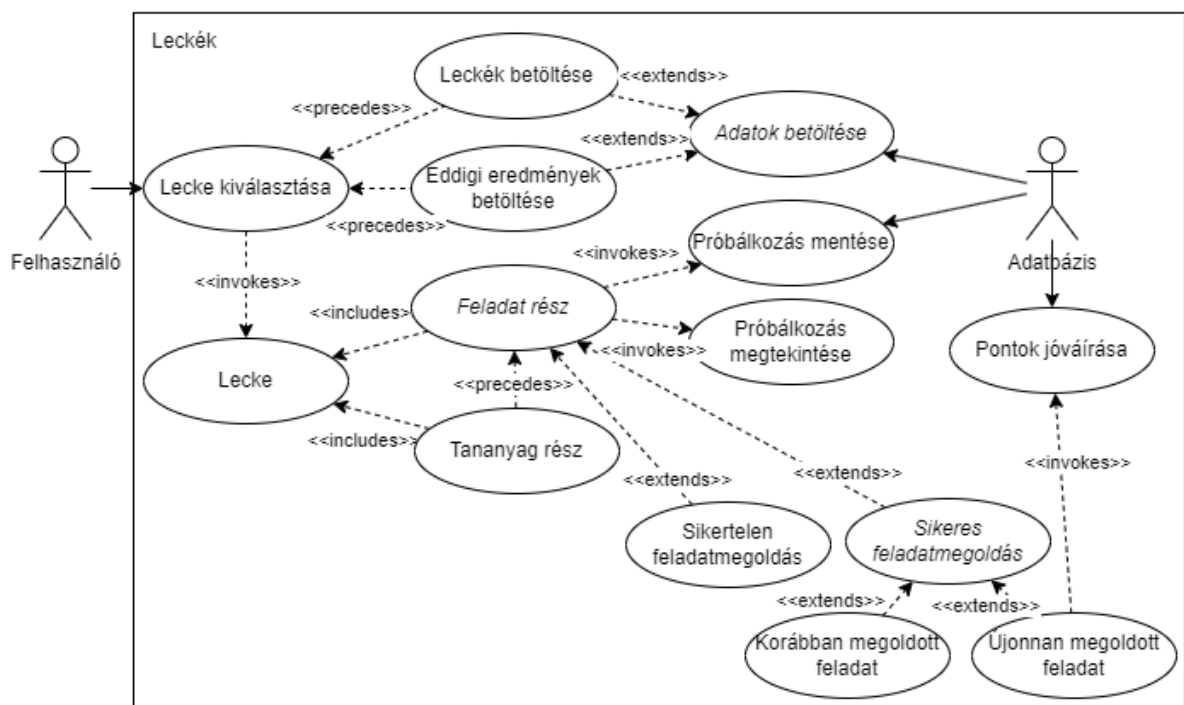


48. ábra - Használati eset diagram, saját profil



49. ábra - Használati eset diagram, más felhasználó profilja



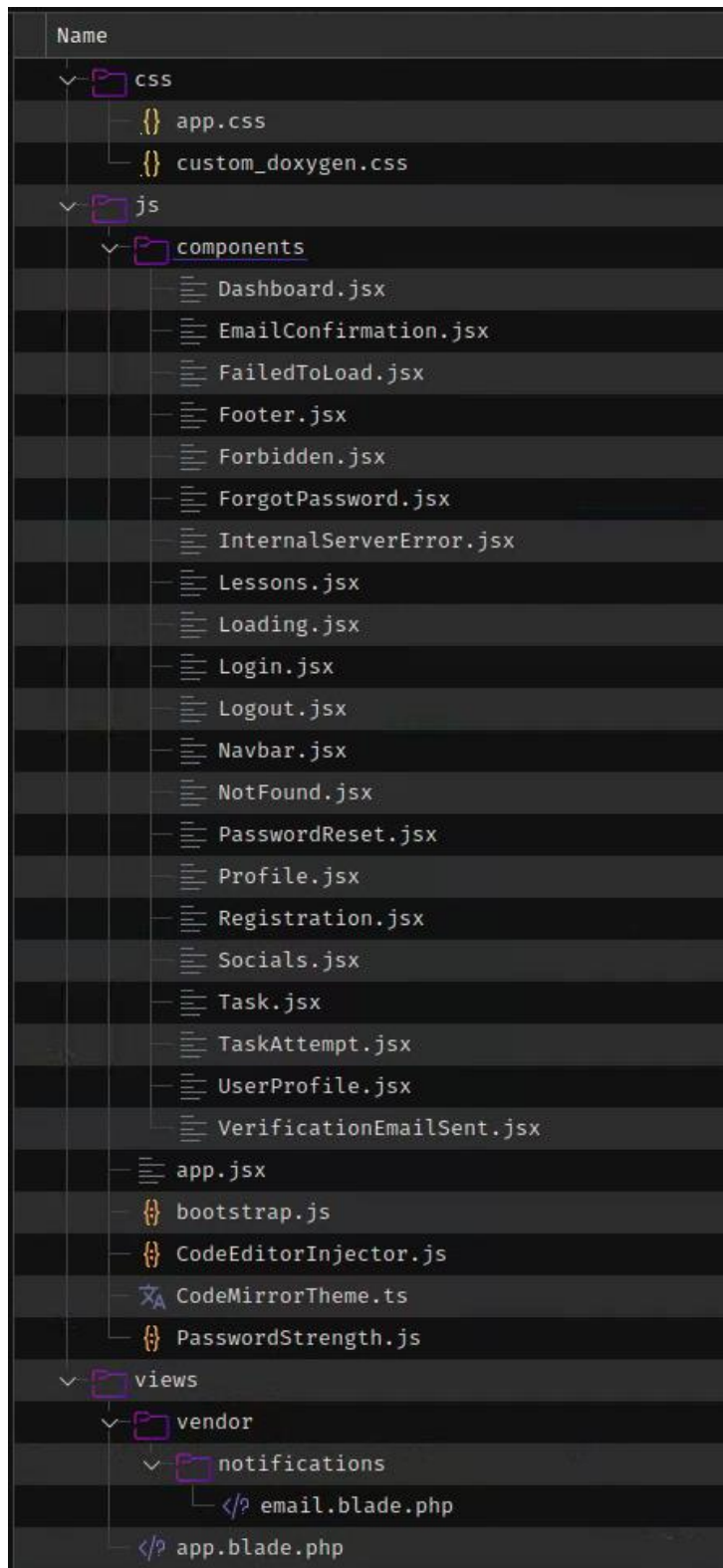


51. ábra - Használati eset diagram, leckék és feladat rész



### 3.3.4. A View bemutatása

A felhasználó felületet megjelenítő fájlok a projektmappa 'resources' könyvtárában találhatóak (lásd az alábbi ábrát).



52. ábra - Megjelenítésért felelős resources könyvtár

#### 3.3.4.1. CSS

A css mappában található az `app.css`, illetve a `custom_doxygen.css`. Előbbi felel az Algoritmizátor stíluslapjaiért, illetve itt vannak beimportálva a TailWindCSS könyvtárai is. Utóbbi a Doxygen által generált dokumentáció stílusáért felel.

#### 3.3.4.2. Views

A `resources/views` mappában találhatóak a megjelenítésért felelős Blade template-ek. Itt csak két template található, mivel az Inertia használatából kifolyólag a szerver részéről csak az `app.blade.php` felel a megjelenítésért, a tényleges oldalak React komponensekből jönnek, amik a `resources/js/components` mappában találhatóak. Összesen két dolgot importál be, az `app.css`-t, illetve az `app.jsx`-et. Az `app.jsx`-ről, valamint a komponensek rendereléséről lentebb írok bővebben. A másik Blade template szolgál a kimenő e-mailek alapjául.

#### 3.3.4.3. JS

Az oldalak renderelésének menete a következő:

- A szerver a beérkező GET request-re, ami lekér egy oldalt, visszaküld egy válaszüzenetet (erről a 3.3.7-es szekcióban írok bővebben).
- Ez először az `app.blade.php`-ba fut be, onnan pedig az `app.jsx`-be.
- Itt a kapott válasz alapján az Inertia megkeresi és lerendereli a `components` mappából a megfelelő React komponenst.

A szerverrel való oda-vissza kommunikáció HTTP requestek segítségével történik, leginkább GET, illetve POST requestekkel, attól függően, hogy történik-e adatmódosítás a backend-en. Ehhez az axios keretrendszer segítségét vettem igénybe.

Egy React komponens felépítése és működése általában a következő:

- Az oldal betöltésekor a state és ref változók (ha vannak) inicializálódnak. Fontos tudni, hogy minden esetben, amikor egy változó megváltozik, újrenderelődik az oldal.
- Ezután, ha szükséges, akkor egy (vagy akár több) GET requesttel lekérődnek az adatok, majd ezekben a state változóknak eltárolásra kerülnek.
- A komponens megjelenítéséért felelős részében, ezen adatok alapján lesz legenerálva a tényleges nézet, amit a felhasználó látni fog.
- Persze betöltés után is küldődhetnek requestek, a felhasználó interakcióira válaszképpen.

#### 3.3.4.4. *Route-ok, middleware*

Még mielőtt rátérnénk teljes mértékben a backend-re, kitérek a backend belépési és kilépési pontjára a frontend felől.

A Laravel úgynevezett 'Route'-okban kezeli a frontend felől érkező HTTP requesteket. Ezek a route-ok a projektmappa 'routes' könyvtárában találhatóak. Az Algoritmizátorban két route fájl van érdemben használatban, a 'web.php' és a 'test.php'. Előbbiben vannak definiálva azok a route-ok, amiket a felhasználó is használ, utóbbiban pedig az egységtesztekhez használt route-ok.

A web.php-ben lévő route-ok működése a következő:

- Amikor jön a frontend felől egy HTTP request egy adott címre, az ide fut be. Az egészet úgy kell elképzelni, mint egy nagyon nagy switch-case-t.
- A route, ahova az adott típusú és adott címre érkező request befut, meghívja a megfelelő kontrollerből a megfelelő függvényt, hogy az a requestből érkező adatok feldolgozása után visszaküldjön egy választ a frontendre.
- Van lehetőség a route-okhoz middleware-t csatolni, ezek funkciója nagyon sok oldalú, de lényegük, hogy megnézhetik, manipulálhatják, vagy akár vissza is utasíthatják a requestet, még mielőtt az az adott kontrollerbe beérkezne.

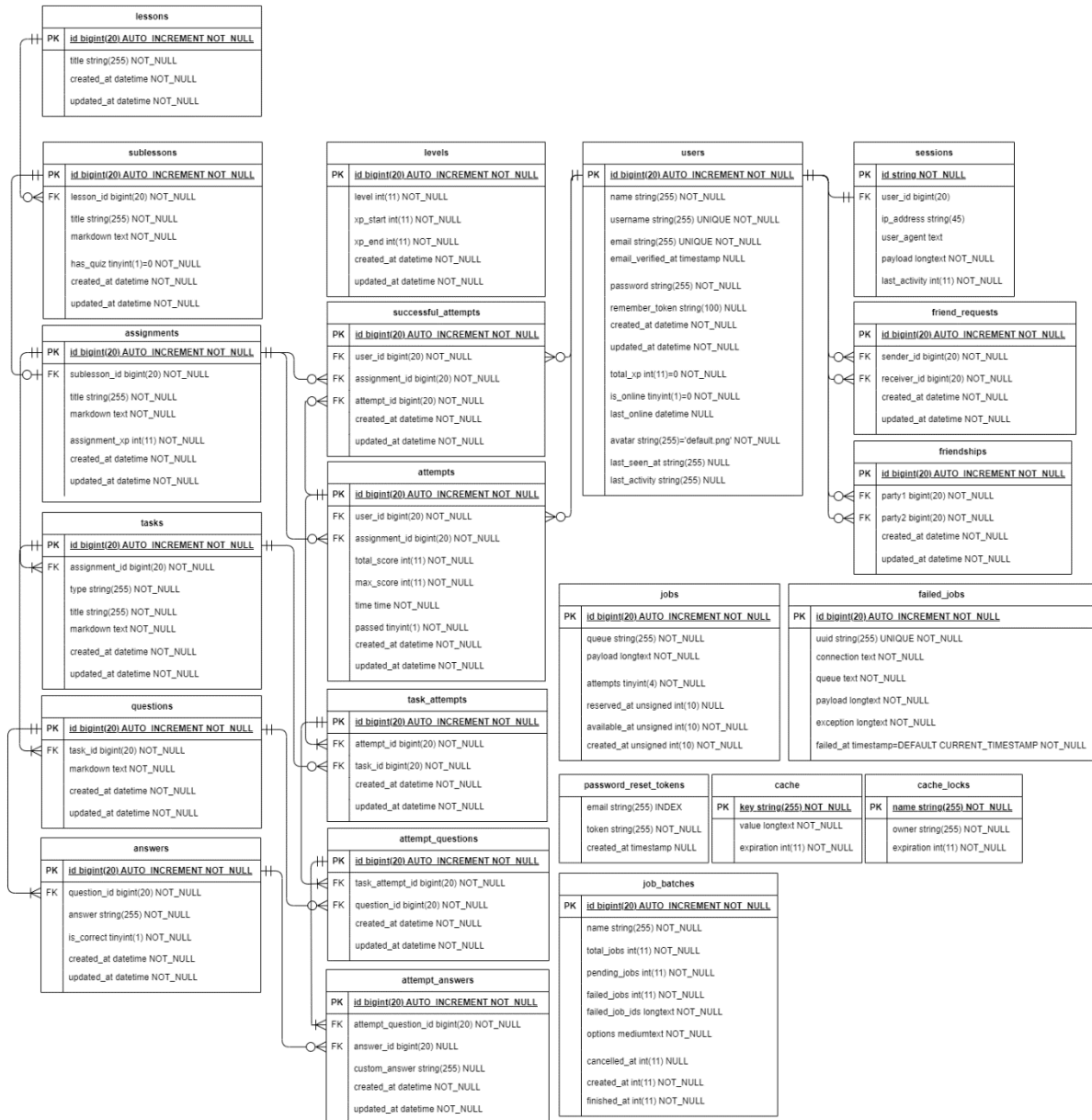
### 3.3.5. Az adatbázis felépítése

#### 3.3.5.1. *Migration*

A Laravel az adatbázisok változásainak követésére úgynevezett 'migration'-öket használ. Egy-egy migration fájl `up()` függvényében például meg lehet adni, hogy milyen táblák legyenek létrehozva, milyen adatokkal legyenek feltöltve. Ezeket a migration-öket végre lehet hajtani a konzolból (de akár a kódból is), ekkor meghívódik az `up()` függvényük és megtörténnek az adatbázisban a változtatások. A migration fájlokban található egy `down()` függvény is, ennek kitöltésével és meghívásával lehet 'rollback'-elni a változtatásokat. A projektmappa 'database/migrations' mappájában elérhető egy migration fájl. Ebben a fájlban vannak tárolva az adatbázis kezdeti táblái, illetve a leckék adatai. Ezt a fájlt futtatva lehet inicializálni az adatbázist az üzembe helyezés során.

### 3.3.5.2. Séma

Az adatbázis sémája a következő diagram szerint épül fel. Itt definiálva vannak a Laravel működéséhez használt táblák, illetve az Algoritmizátor saját táblái is.



53. ábra - Az adatbázis sémája

A model szorosan kapcsolódik az adatbázishoz az Eloquent ORM segítségével. Az Object Relational Mapping használatával a kontrollerekből a modellen keresztül függvényhívások segítségével lehet lekérdezni az adatbázisból, vagy az abban lévő adatokat módosítani.

The diagram illustrates the 'AppModels' package, which contains several classes and interfaces. The classes are organized into a hierarchical structure, with 'Lesson' at the top and 'Answer' at the bottom. The 'User' class is a central entity, interacting with 'Level', 'Friendship', and 'FriendRequest'. The 'Task' and 'TaskAttempt' classes are also central, with 'TaskAttempt' extending 'Task'. The 'Question' and 'Answer' classes are related to 'TaskAttempt' and 'AttemptQuestion' and 'AttemptAnswer' respectively. The 'User' class has a 'password' attribute and a 'remember\_token' attribute. The 'Level' class has a 'xp\_end' attribute. The 'Friendship' and 'FriendRequest' classes have 'party1' and 'party2' attributes. The 'Lesson' class has a 'sublessons()' method. The 'Assignment' class has a 'sublessons()' method. The 'Attempt' class has a 'successfulAttempts()' method. The 'Task' class has a 'questions()' method. The 'TaskAttempt' class has a 'questions()' method. The 'Question' class has an 'answers()' method. The 'Answer' class has an 'attemptAnswers()' method. The 'User' class has a 'getLevelAttribute()' method. The 'User' class has a 'sendEmailVerificationNotification()' method. The 'User' class has a 'sendPasswordResetNotification()' method. The 'User' class has a 'findById()' method. The 'User' class has a 'senders()' method. The 'User' class has a 'receivers()' method. The 'User' class has a 'friendTo()' method. The 'User' class has a 'friends()' method. The 'User' class has a 'successfulAttempts()' method. The 'Level' class has a 'findLevelByXp()' method. The 'Friendship' class has a 'party1()' method. The 'Friendship' class has a 'party2()' method. The 'FriendRequest' class has a 'sender()' method. The 'FriendRequest' class has a 'receiver()' method. The 'Lesson' class has a 'sublessons()' method. The 'Assignment' class has a 'sublessons()' method. The 'Attempt' class has a 'successfulAttempts()' method. The 'Task' class has a 'questions()' method. The 'TaskAttempt' class has a 'questions()' method. The 'Question' class has an 'answers()' method. The 'Answer' class has an 'attemptAnswers()' method. The 'User' class has a 'getLevelAttribute()' method. The 'User' class has a 'sendEmailVerificationNotification()' method. The 'User' class has a 'sendPasswordResetNotification()' method. The 'User' class has a 'findById()' method. The 'User' class has a 'senders()' method. The 'User' class has a 'receivers()' method. The 'User' class has a 'friendTo()' method. The 'User' class has a 'friends()' method. The 'User' class has a 'successfulAttempts()' method. The 'Level' class has a 'findLevelByXp()' method. The 'Friendship' class has a 'party1()' method. The 'Friendship' class has a 'party2()' method. The 'FriendRequest' class has a 'sender()' method. The 'FriendRequest' class has a 'receiver()' method.

41

Egy kis információ a Laravel modellek működéséről:

A modellek neveinek is követniük kell egy konvenciót, a 'User' model az adatbázisban a 'users' táblához kapcsolódik, a 'Lesson' model a 'lessons' táblához, a 'FriendRequest' model a 'friend\_requests' táblához és így tovább. A 'HasMany', 'BelongsTo', 'hasOne' típusú függvények kapcsolásokat ábrázolnak a táblák között, továbbá a User model az e-mail megerősítés funkcionalitás miatt még tartalmaz plusz függvényeket.

Alább bemutatom a User modelt és a leckék felépítését részletesen:

- User model:

A User model tartja nyilván a felhasználókat. A pontos felépítése a következő:

- name: A felhasználó neve
- username: A felhasználó felhasználóneve, egyedi.
- email: A felhasználó e-mail címe.
- email\_verified\_at: Ez a dátum az e-mail cím megerősítésének dátuma, ha nincs megerősítve az e-mail cím, akkor itt null szerepel.
- password: A felhasználó jelszava, hashelve.
- total\_xp: A felhasználó eddig összegyűjtött összes xp-je, ez alapján van a user összekapcsolva a levels táblával, és így nyeri ki a megfelelő levelt.
- is\_online: Éppen online-e a felhasználó, ezt a snoop és a logout middleware kezeli.
- last\_online: Mikor volt legutóbb online a felhasználó, ezt a snoop middleware kezeli.
- avatar: A felhasználó profilképének a neve. A profilképek egy fix mappában vannak eltárolva, lekérdezésükkor ez a név lesz az útvonalhoz hozzáillesztve.
- last\_seen\_at: Melyik oldalon járt a felhasználó utoljára, a snoop middleware kezeli.
- last\_activity: Mit csinált utoljára a felhasználó, a snoop middleware kezeli.
- remember\_token: Ez az 'Emlékezz rám' funkcionalitáshoz használt token. Ha a felhasználó nem pipálta be belépés előtt, akkor null az értéke.

- Leckék, tananyag, feladatok felépítése:

A leckék és a feladatok hierarchikusan és modulárisan lettek megtervezve. A Lesson model egyfajta kategória a Sublesson-öknek. Ez reprezentálja a Tananyag oldalon lévő navigációs sávon lévő fő kategóriákat. Ez alá vannak beosztva a Sublesson-ök, ahol már egy picit több adat tárolódik az adatbázisban. Ezeknek is van címük, de itt már található egy markdown mező is. Ez maga a leckének a törzse. Ezek a markdownok a storage/markdowns mappában találhatóak meg, és az inicializáló migration innen seedeli bele az adatbázisba. A leckékben lévő kódszerkesztők működéséhez viszont a kódot úgy kell elhelyezni a markdownban, hogy azok közvetlenül egymás alatt legyenek, ha azt szeretnénk, hogy a programnyelv váltogatók fülek működjenek. Található még a Sublesson modelben egy has\_quiz bool mező. Ez indikálja, hogy a leckéhez tartozik-e kvíz vagy sem.

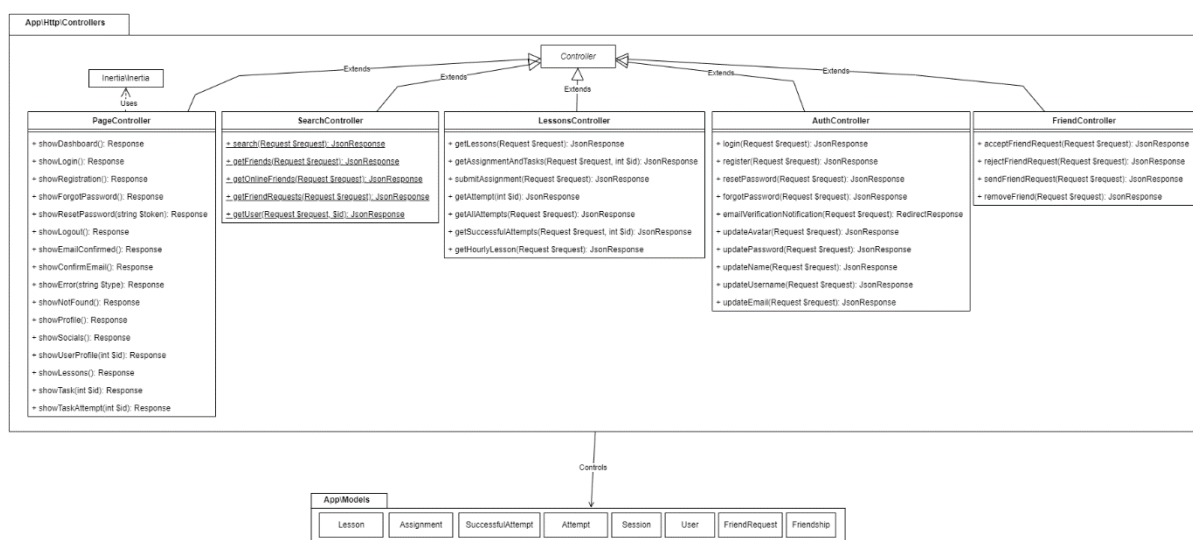
A feladatok is hasonlóképpen lettek megtervezve, ezek legnagyobb részt az inicializáló migrationben vannak nyílvántartva, csak egy pár nagyobb feladat kapott saját markdown fájlt. Ezeket is a nagyfokú modularitás miatt szabadon lehet módosítani, új feladatokat kitalálni, meglévőket módosítani.



### 3.3.7. A Controller bemutatása

Az Algoritmizátor kontroller részében 5 kontroller van (lásd: 55. ábra):

- A PageController felel a frontenden megjelenítendő oldalak rendereléséért. A web.php-ban lévő route bejegyzések innen hívják meg a megfelelő függvényt, ami visszaadja a lerenderelendő oldalt.
- A SearchController felel a különböző keresések lebonyolításáért. A search() függvény hívódik meg például a Közösség oldalon a Keresés fülön, miközben keresgélünk a felhasználók között.
- A LessonsController felel a megfelelő leckék lekérdezéséért, miközben navigálunk a Tananyag oldalon, illetve ez a kontroller kezeli a feladatok adatainak kinyerését, a feladatbeküldést, pontozást és visszaadást is. A vezérlőpulton középén megjelenő aktuális óra leckéjét is ez a kontroller adja vissza.
- Az AuthController felelős az összes autentikációval kapcsolatos lekérdezésekért és módosításokért, illetve ezen kontrolleren keresztül tudja a felhasználó az adatait megváltoztatni. A login() és a register() függvények ezeken kívül pedig munkamenetet is kezelnek.
- A FriendController felelős a barátkérélmek kezeléséért, illetve a barátok esetleges eltávolításáért is.

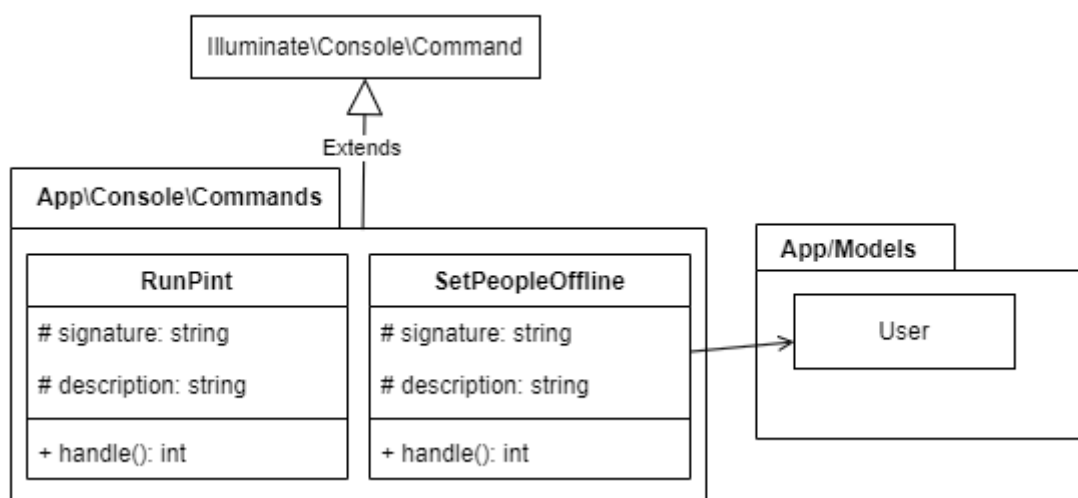


55. ábra - A Controller osztálydiagramja

### 3.3.8. A backend egyéb részeinek bemutatása

#### 3.3.8.1. Console commands

Az itt lévő osztályok konzol utasításokat futtatnak. A RunPint osztály a Laravel Pint segédtool-t futtatja, ami kijavítja a kódban az esetleg előforduló esztétikai hibákat. Ez leginkább fejlesztés során hasznos, production környezetben nincs sok értelme. A SetPeopleOffline osztály már hasznosabb, ez az osztály egy adatbázisműveletet futtat le, ami az inaktív felhasználókat offline állapotba rakja. Mivel van 'Emlékezz rám' funkció, így a bejelentkezés nem feltétlenül jelenti azt, hogy egy felhasználó aktívan használja is az oldalt.

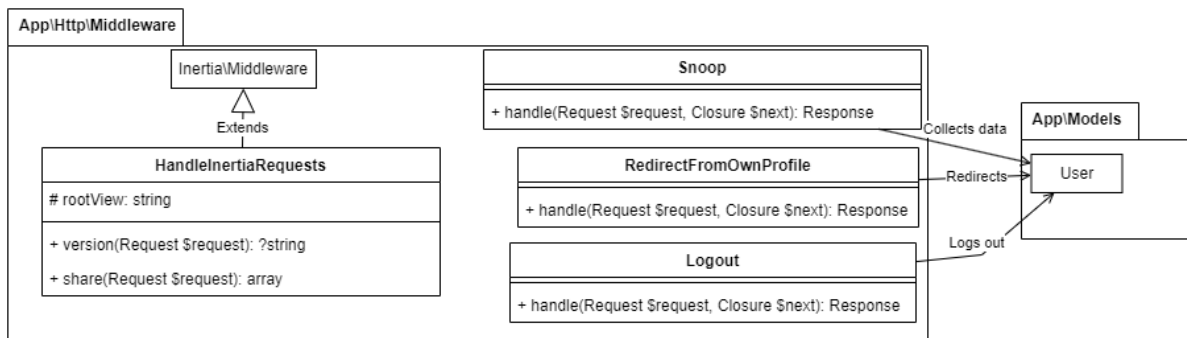


56. ábra - A Console/Command mappában lévő osztályok osztálydiagramja

### 3.3.8.2. Middleware

A következő Middleware osztályok közül a HandleInertiaRequests osztály a legfontosabb, ugyanis ez a middleware kezeli le azokat a requesteket, amiket az Inertia-nak kell lekezelnie. A Snoop osztály minden oldalt lekérdező route-on rajta van, ez információt gyűjt<sup>2</sup> a felhasználóról. Ennek segítségével jelenik meg a felhasználók legutóbbi tevékenysége a Közösség oldalon, illetve ennek az osztálynak a segítségével van követve, hogy egy felhasználó éppen aktív-e.

A RedirectFromOwnProfile egy biztonsági middleware, arra az esetre, ha valaki azon az URL-en keresztül szeretné megnézni a saját profilját, ami egyébként a többi felhasználó profiljának van fenntartva.



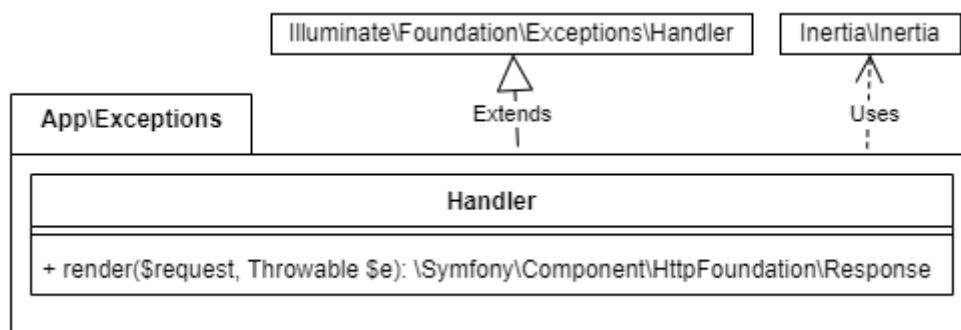
57. ábra - A Middleware osztályok osztálydiagramja

---

<sup>2</sup> Nem kell megijedni, semmi szenzitív információt nem gyűjt. Mindössze annyit gyűjt, hogy mikor volt legutóbb aktív a felhasználó, illetve milyen oldalon járt utoljára.

### 3.3.8.3. Exceptions

Az Exceptions-ben lévő Handler osztály a rossz requestek megjelenítésére van fenntartva. Tehát ha olyan oldalt szeretne megnyitni a felhasználó, ami nem létezik, akkor ez a Handler osztály fog visszaküldeni egy 404 oldalt az Inertia-nak.



58. ábra - Az Exception handler osztálydiagramja

## 3.4. Tesztelés

### 3.4.1. Egység tesztek

A tesztelés során a backend függvényeit vettem alapul. Az alábbi táblázatokban bemutatom a tesztelési tervet. Felhívom rá a figyelmet, hogy mivel az inicializáló migration minden teszt előtt lefut, ezért a tesztek egy teszt adatbázist igényelnek, ha a phpunit.xml fájlban nincs beállítva külön adatbázis, akkor a production adatbázisban fognak végrehajtódni a tesztek, ami az összes adat törlésével jár! Továbbá a tesztek más route-okat használnak, mint a production környezet, ezek a route-ok a routes/test.php fájlban találhatóak.

A projektmappa .github mappájában található továbbá egy test.yml fájl, ez minden git push-nál létrehoz egy pipeline-t és lefuttatja az alábbi teszteket.

#### 3.4.1.1. Model tesztek

#	Model	Leírás	Eredmény
1.1.1	Answer	Answer létrehozható érvényes adatokkal.	OK
1.1.2		Az answer kérdéshez tartozik.	OK
1.1.3		Answer nem hozható létre kérdés nélkül.	OK

<b>1.1.4</b>		Answer nem hozható létre answer szöveg nélkül.	OK
<b>1.1.5</b>		Answer is_correct flag lehet igaz.	OK
<b>1.1.6</b>		Answer is correct flag lehet hamis.	OK
<b>1.2.1</b>	Assignment	Assignment létrehozható érvényes adatokkal.	OK
<b>1.2.2</b>		Az assignment sublesson-höz tartozik.	OK
<b>1.2.3</b>		Az assignment-nek sok task-ja van.	OK
<b>1.2.4</b>		Az assignment-nek sok attempt-je van.	OK
<b>1.2.5</b>		Az assignment-nek sok sikeres attempt-je van.	OK
<b>1.2.6</b>		Assignment nem hozható létre sublesson nélkül.	OK
<b>1.3.1</b>	Attempt answer	Attempt answer létrehozható érvényes adatokkal.	OK
<b>1.3.2</b>		Az attempt answer attempt question-höz és answer-höz tartozik.	OK
<b>1.3.3</b>		Attempt answer nem hozható létre attempt question vagy answer nélkül.	OK
<b>1.4.1</b>	Attempt question	Attempt question létrehozható érvényes adatokkal.	OK
<b>1.4.2</b>		Az attempt question task attempt-hez és question-höz tartozik.	OK
<b>1.4.3</b>		Attempt question nem hozható létre task attempt vagy question nélkül.	OK
<b>1.5.1</b>	Attempt	Attempt létrehozható érvényes adatokkal.	OK
<b>1.5.2</b>		Az attempt felhasználóhoz és assignment-hez tartozik.	OK

<b>1.5.3</b>		Attempt nem hozható létre felhasználó vagy assignment nélkül.	OK
<b>1.6.1</b>	Friend request	Friend request létrehozható érvényes adatokkal.	OK
<b>1.6.2</b>		A friend request sender-hez és receiver-höz tartozik.	OK
<b>1.6.3</b>		Friend request nem hozható létre sender nélkül.	OK
<b>1.6.4</b>		Friend request nem hozható létre receiver nélkül.	OK
<b>1.7.1</b>	Friendship	Friendship létrehozható érvényes adatokkal.	OK
<b>1.7.2</b>		A friendship party1-hez és party2-höz tartozik.	OK
<b>1.7.3</b>		Friendship nem hozható létre party1 nélkül.	OK
<b>1.7.4</b>		Friendship nem hozható létre party2 nélkül.	OK
<b>1.8.1</b>	Lesson	Lecke létrehozható érvényes adatokkal.	OK
<b>1.8.2</b>		A cím alapján keresett lecke a helyes leckét adja vissza.	OK
<b>1.8.3</b>		A cím alapján keresett lecke null-t ad vissza, ha a cím nem létezik.	OK
<b>1.8.4</b>		Sublesson-ök hozzáadhatók a leckéhez.	OK
<b>1.9.1</b>	Level	Level létrehozható érvényes adatokkal.	OK
<b>1.9.2</b>		A xp alapján keresett level a helyes level-t adja vissza.	OK
<b>1.9.3</b>		A xp alapján keresett level null-t ad vissza, ha az xp kívül esik a tartományon.	OK
<b>1.10.1</b>	Question	Question létrehozható érvényes adatokkal.	OK

<b>1.10.2</b>		A question task-hoz tartozik.	OK
<b>1.10.3</b>		A question-nek sok answer-je van.	OK
<b>1.10.4</b>		Question nem hozható létre task nélkül.	OK
<b>1.11.1</b>	Sublesson	Sublesson létrehozható érvényes adatokkal.	OK
<b>1.11.2</b>		A sublesson lesson-höz tartozik.	OK
<b>1.11.3</b>		Sublesson nem hozható létre lesson nélkül.	OK
<b>1.11.4</b>		Sublesson nem hozható létre cím nélkül.	OK
<b>1.11.5</b>		Sublesson nem hozható létre markdown nélkül.	OK
<b>1.11.6</b>		Sublesson has_quiz flag-e boolean típusú.	OK
<b>1.12.1</b>	Successful attempt	Successful attempt létrehozható érvényes adatokkal.	OK
<b>1.12.2</b>		A successful attempt felhasználóhoz, assignment-hoz és attempt-hez tartozik.	OK
<b>1.12.3</b>		Successful attempt nem hozható létre felhasználó, assignment vagy attempt nélkül.	OK
<b>1.13.1</b>	Task attempt	Task attempt létrehozható érvényes adatokkal.	OK
<b>1.13.2</b>		A task attempt attempt-hez és task-hoz tartozik.	OK
<b>1.13.3</b>		Task attempt nem hozható létre attempt vagy task nélkül.	OK
<b>1.14.1</b>	Task	Task létrehozható érvényes adatokkal.	OK
<b>1.14.2</b>		A task assignment-hez tartozik.	OK
<b>1.14.3</b>		A task-nak sok question-je van.	OK

<b>1.14.4</b>		Task nem hozható létre assignment nélkül.	OK
<b>1.15.1</b>	User	Felhasználó létrehozható érvényes adatokkal.	OK
<b>1.15.2</b>		A felhasználó szintje helyesen számított.	OK
<b>1.15.3</b>		A felhasználó friend request-et küldhet.	OK
<b>1.15.4</b>		A felhasználó friend request-et fogadhat.	OK
<b>1.15.5</b>		A felhasználónak lehetnek sikeres attempt-jei.	OK
<b>1.15.6</b>		A felhasználónak lehetnek attempt-jei.	OK

#### 3.4.1.2. Controller tesztek

#	Controller	Leírás	Eredmény
<b>2.1.1</b>	Auth	Sikeres login érvényes hitelesítő adatokkal.	OK
<b>2.1.2</b>		Sikertelen login érvénytelen hitelesítő adatokkal.	OK
<b>2.1.3</b>		Sikeres registration érvényes adatokkal.	OK
<b>2.1.4</b>		Sikertelen registration érvénytelen adatokkal.	OK
<b>2.1.5</b>		Sikeres password reset érvényes tokennel.	OK
<b>2.1.6</b>		Sikertelen password reset érvénytelen tokennel.	OK
<b>2.2.1</b>	Friend	Friend request elfogadható.	OK
<b>2.2.2</b>		Friend request elutasítható.	OK
<b>2.2.3</b>		Friend request küldhető.	OK
<b>2.2.4</b>		Friend eltávolítható.	OK
<b>2.3.1</b>	Lessons	A leckék helyesen lettek lekérve.	OK



<b>2.3.2</b>		Az assignment és tasks helyesen lettek lekérve.	OK
<b>2.3.3</b>		Az assignment beadása sikeres.	OK
<b>2.3.4</b>		Az attempt helyesen lett lekérve.	OK
<b>2.4.1</b>	Search	A felhasználók helyesen lettek keresve.	OK
<b>2.4.2</b>		A friends helyesen lett lekérve.	OK
<b>2.4.3</b>		Az online friends helyesen lett lekérve.	OK
<b>2.4.4</b>		A friend requests helyesen lett lekérve.	OK
<b>2.4.5</b>		A felhasználó nem lett lekérve.	OK
<b>2.4.6</b>		A felhasználó nem található hibát ad vissza.	OK

#### 3.4.1.3. Egyéb tesztek

#	Típus	Leírás	Eredmény
<b>3.1.1</b>	Set people offline middleware	A set people offline command az embereket offline állapotba helyezi.	OK
<b>3.1.2</b>		A set people offline command nem érinti azokat az embereket, akik az elmúlt 20 percben online voltak.	OK
<b>3.2.1</b>	Exception handler middleware	A handler 404 nézetet jelenít meg a 'not_found' http kivétel esetén.	OK
<b>3.2.2</b>		A handler más kivételek esetén a szülőt jeleníti meg.	OK
<b>3.3.1</b>	Custom reset notification	Custom reset notification létrehozható érvényes tokennel.	OK

<b>3.3.2</b>		A custom reset notification helyes csatornákat ad vissza.	OK
<b>3.3.3</b>		A custom reset notification helyes verification URL-t generál.	OK
<b>3.3.4</b>		A custom reset notification helyes mail üzenetet ad vissza.	OK
<b>3.4.1</b>	Custom verify email	Custom verify email létrehozható.	OK
<b>3.4.2</b>		A custom verify email helyes csatornákat ad vissza.	OK
<b>3.4.3</b>		A custom verify email helyes verification URL-t generál.	OK
<b>3.4.4</b>		A custom verify email helyes mail üzenetet ad vissza.	OK

#### 3.4.2. Felhasználói felület tesztelése

A felhasználói felületet a fejlesztés közben és után is, én, illetve kedves barátaim manuálisan teszteltük. A tesztelések során felmerülő problémák ki lettek javítva, a vizuális elemek tovább lettek csiszolva.

### 3.5. Üzembe helyezés, karbantartás

#### 3.5.1. Konfigurálás

Az üzembe helyezéshez szükséges egy webszerver, egy MySQL adatbázisszerver és egy e-mail szerver. Ha az e-mail szerver nem elérhető, vagy nem megoldható a felállítása, akkor sajnos az Algoritmizátor funkciói nagyban sérülnek, mivel e-mail megerősítés nélkül a webalkalmazás funkciói nem elérhetőek. Erre egy kiskapu, ha egy regisztrált felhasználónak az adatbázisban közvetlenül beállítunk az 'email\_verified\_at' mezőhöz egy dátumot.

Ha a fenti követelmények teljesülnek, akkor a következők szerint helyezhető üzembe az Algoritmizátor. Erősen javaslom linux alapú szerver használatát, de megoldható az üzembe

helyezés más platformon is. A következőekben én feltételezem, hogy az üzembe helyezés linux környezetben történik.

1. Egy tetszőleges mappába ki kell csomagolni a forrásfájlokat a webszerveren.
2. Telepíteni kell a Composer és a NodeJS csomagokat a következő parancsokkal:
  - a. `composer install`
  - b. `npm install`
3. Ezután létre kell hozni egy `.env` fájlt a `.env.example` fájl alapján a következőképpen:
  - a. `cp .env.example .env`
4. Generálni kell egy alkalmazáskulcsot a következőképpen:
  - a. `php artisan key:generate`
5. Be kell állítani a megfelelő környezeti változókat, az adatbázis kapcsolatot és az e-mail szerver konfigurációját a `.env` fájlban
6. Ezután migrálni kell az adatbázis táblákat a következő paranccsal:
  - a. `php artisan migrate`
7. Ezután lebuildeljük a frontend asseteket a következő paranccsal:
  - a. `npm run build`
8. El kell indítani a Laravel szerveret a következő paranccsal:
  - a. `php artisan serve`
9. Majd legutolsó sorban létre kell hozni egy 'algoritmizator' mappát a webszerver gyökerében és egy symlinket beállítani a projekt 'public' mappája és ezen mappa között, továbbá még egy symlinket a projektkönyvtár public mappájából a projektkönyvtár storage/app/public mappájába storage néven.
10. A SetPeopleOffline funkcionalitáshoz még egy cron bejegyzést be kell állítani, hogy a Laravel scheduler működni tudjon:
  - a. `* * * * * cd /út-a-projekthez && php artisan schedule:run >> /dev/null 2>&1`

## 4. Összefoglalás

### 4.1. Skálázhatóság

A weboldal egészen jól skálázható, bővíthető újabb leckékkel, feladatokkal, algoritmus implementációkkal. Mivel a tananyag rész markdown-ból generálja a leckéket, így ezen a részen is nagyon sok a kiaknázatlan lehetőség.

### 4.2. További fejlesztési lehetőségek

A tervezés része volt, csak idő hiányában már nem volt lehetőség az implementálására, egy automatikus futtató és tesztkörnyezet, illetve olyan feladatok, ahol egy adott problémára kell leprogramozni egy algoritmust, adott időn belül.

## 5. Irodalomjegyzék

### 5.1. Felhasznált dokumentációk

- [1] „The PHP Framework for Web Artisans” [Online] Elérhető: <https://laravel.com/docs/11.x> [Hozzáférés dátuma: 2024. 05. 28.]
- [2] „Eloquent ORM” [Online] Elérhető: <https://laravel.com/docs/11.x/eloquent> [Hozzáférés dátuma: 2024. 05. 28.]
- [3] „Pint - An Opinionated Code Style Fixer for PHP” [Online] Elérhető: <https://laravel.com/docs/11.x/pint> [Hozzáférés dátuma: 2024. 05. 28.]
- [4] „Vite - Next Generation Frontend Tooling” [Online] Elérhető: <https://laravel.com/docs/11.x/vite> [Hozzáférés dátuma: 2024. 05. 28.]
- [5] „React - Learn” [Online] Elérhető: <https://react.dev/learn> [Hozzáférés dátuma: 2024. 05. 28.]
- [6] „Tailwind CSS - Documentation” [Online] Elérhető: <https://tailwindcss.com/docs> [Hozzáférés dátuma: 2024. 05. 28.]
- [7] „CodeMirror - Documentation” [Online] Elérhető: <https://codemirror.net/docs/> [Hozzáférés dátuma: 2024. 05. 28.]
- [8] „Inertia.js - The Modern Monolith” [Online] Elérhető: <https://inertiajs.com> [Hozzáférés dátuma: 2024. 05. 28.]

### 5.2. Felhasznált irodalom

- [9] „Array Data Structure Guide” [Online] Elérhető: <https://www.geeksforgeeks.org/array-data-structure-guide/> [Hozzáférés dátuma: 2024. 05. 28.]
- [10] „Array (data structure)” [Online] Elérhető: [https://en.wikipedia.org/wiki/Array\\_\(data\\_structure\)](https://en.wikipedia.org/wiki/Array_(data_structure)) [Hozzáférés dátuma: 2024. 05. 28.]
- [11] „Linked List Data Structure” [Online] Elérhető: <https://www.geeksforgeeks.org/linked-list-data-structure/> [Hozzáférés dátuma: 2024. 05. 28.]
- [12] „Linked list” [Online] Elérhető: [https://en.wikipedia.org/wiki/Linked\\_list](https://en.wikipedia.org/wiki/Linked_list) [Hozzáférés dátuma: 2024. 05. 28.]

- [13] „Stack Data Structure” [Online] Elérhető: <https://www.geeksforgeeks.org/stack-data-structure/> [Hozzáférés dátuma: 2024. 05. 28.]
- [14] „Stack (abstract data type)” [Online] Elérhető: [https://en.wikipedia.org/wiki/Stack\\_\(abstract\\_data\\_type\)](https://en.wikipedia.org/wiki/Stack_(abstract_data_type)) [Hozzáférés dátuma: 2024. 05. 28.]
- [15] „Queue Data Structure” [Online] Elérhető: <https://www.geeksforgeeks.org/queue-data-structure/> [Hozzáférés dátuma: 2024. 05. 28.]
- [16] „Queue (abstract data type)” [Online] Elérhető: [https://en.wikipedia.org/wiki/Queue\\_\(abstract\\_data\\_type\)](https://en.wikipedia.org/wiki/Queue_(abstract_data_type)) [Hozzáférés dátuma: 2024. 05. 28.]
- [17] „Tree Data Structure” [Online] Elérhető: <https://www.geeksforgeeks.org/tree-data-structure/> [Hozzáférés dátuma: 2024. 05. 28.]
- [18] „Tree (data structure)” [Online] Elérhető: [https://en.wikipedia.org/wiki/Tree\\_\(data\\_structure\)](https://en.wikipedia.org/wiki/Tree_(data_structure)) [Hozzáférés dátuma: 2024. 05. 28.]
- [19] „Graph Data Structure and Algorithms” [Online] Elérhető: <https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/> [Hozzáférés dátuma: 2024. 05. 28.]
- [20] „Graph (abstract data type)” [Online] Elérhető: [https://en.wikipedia.org/wiki/Graph\\_\(abstract\\_data\\_type\)](https://en.wikipedia.org/wiki/Graph_(abstract_data_type)) [Hozzáférés dátuma: 2024. 05. 28.]
- [21] „Hashing Data Structure” [Online] Elérhető: <https://www.geeksforgeeks.org/hashing-data-structure/> [Hozzáférés dátuma: 2024. 05. 28.]
- [22] „Hash table” [Online] Elérhető: [https://en.wikipedia.org/wiki/Hash\\_table](https://en.wikipedia.org/wiki/Hash_table) [Hozzáférés dátuma: 2024. 05. 28.]
- [23] „Insertion Sort” [Online] Elérhető: <https://www.geeksforgeeks.org/insertion-sort/> [Hozzáférés dátuma: 2024. 05. 28.]
- [24] „Insertion sort” [Online] Elérhető: [https://en.wikipedia.org/wiki/Insertion\\_sort](https://en.wikipedia.org/wiki/Insertion_sort) [Hozzáférés dátuma: 2024. 05. 28.]
- [25] „Selection Sort” [Online] Elérhető: <https://www.geeksforgeeks.org/selection-sort/> [Hozzáférés dátuma: 2024. 05. 28.]

- [26] „Selection sort” [Online] Elérhető: [https://en.wikipedia.org/wiki/Selection\\_sort](https://en.wikipedia.org/wiki/Selection_sort)  
[Hozzáférés dátuma: 2024. 05. 28.]
- [27] „Merge Sort” [Online] Elérhető: <https://www.geeksforgeeks.org/merge-sort/>  
[Hozzáférés dátuma: 2024. 05. 28.]
- [28] „Merge sort” [Online] Elérhető: [https://en.wikipedia.org/wiki/Merge\\_sort](https://en.wikipedia.org/wiki/Merge_sort)  
[Hozzáférés dátuma: 2024. 05. 28.]
- [29] „Quick Sort” [Online] Elérhető: <https://www.geeksforgeeks.org/quick-sort/>  
[Hozzáférés dátuma: 2024. 05. 28.]
- [30] „Quicksort” [Online] Elérhető: <https://en.wikipedia.org/wiki/Quicksort> [Hozzáférés dátuma: 2024. 05. 28.]
- [31] „Heap Sort” [Online] Elérhető: <https://www.geeksforgeeks.org/heap-sort/>  
[Hozzáférés dátuma: 2024. 05. 28.]
- [32] „Heapsort” [Online] Elérhető: <https://en.wikipedia.org/wiki/Heapsort> [Hozzáférés dátuma: 2024. 05. 28.]
- [33] „Bucket Sort” [Online] Elérhető: <https://www.geeksforgeeks.org/bucket-sort/>  
[Hozzáférés dátuma: 2024. 05. 28.]
- [34] „Bucket sort” [Online] Elérhető: [https://en.wikipedia.org/wiki/Bucket\\_sort](https://en.wikipedia.org/wiki/Bucket_sort)  
[Hozzáférés dátuma: 2024. 05. 28.]
- [35] „Counting Sort” [Online] Elérhető: <https://www.geeksforgeeks.org/counting-sort/>  
[Hozzáférés dátuma: 2024. 05. 28.]
- [36] „Counting sort” [Online] Elérhető: [https://en.wikipedia.org/wiki/Counting\\_sort](https://en.wikipedia.org/wiki/Counting_sort)  
[Hozzáférés dátuma: 2024. 05. 28.]
- [37] „Radix Sort” [Online] Elérhető: <https://www.geeksforgeeks.org/radix-sort/>  
[Hozzáférés dátuma: 2024. 05. 28.]
- [38] „Radix sort” [Online] Elérhető: [https://en.wikipedia.org/wiki/Radix\\_sort](https://en.wikipedia.org/wiki/Radix_sort) [Hozzáférés dátuma: 2024. 05. 28.]
- [39] „Graph Representation” [Online] Elérhető: <https://www.geeksforgeeks.org/graph-representation/> [Hozzáférés dátuma: 2024. 05. 28.]
- [40] „Breadth First Search or BFS for a Graph” [Online] Elérhető: <https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/>  
[Hozzáférés dátuma: 2024. 05. 28.]

- [41] „Breadth-first search” [Online] Elérhető: [https://en.wikipedia.org/wiki/Breadth-first\\_search](https://en.wikipedia.org/wiki/Breadth-first_search) [Hozzáférés dátuma: 2024. 05. 28.]
- [42] „Depth First Search or DFS for a Graph” [Online] Elérhető: <https://www.geeksforgeeks.org/depth-first-search-or-dfs-for-a-graph/> [Hozzáférés dátuma: 2024. 05. 28.]
- [43] „Depth-first search” [Online] Elérhető: [https://en.wikipedia.org/wiki/Depth-first\\_search](https://en.wikipedia.org/wiki/Depth-first_search) [Hozzáférés dátuma: 2024. 05. 28.]
- [44] „Shortest Path Algorithms” [Online] Elérhető: <https://www.geeksforgeeks.org/shortest-path-algorithms/> [Hozzáférés dátuma: 2024. 05. 28.]
- [45] „Dijkstra's algorithm” [Online] Elérhető: [https://en.wikipedia.org/wiki/Dijkstra%27s\\_algorithm](https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm) [Hozzáférés dátuma: 2024. 05. 28.]
- [46] „Bellman–Ford algorithm” [Online] Elérhető: [https://en.wikipedia.org/wiki/Bellman%E2%80%93Ford\\_algorithm](https://en.wikipedia.org/wiki/Bellman%E2%80%93Ford_algorithm) [Hozzáférés dátuma: 2024. 05. 28.]
- [47] „Pointers in C and C++ | Set 1 (Introduction, Arithmetic and Array)” [Online] Elérhető: <https://www.geeksforgeeks.org/pointers-in-c-and-c-set-1-introduction-arithmetic-and-array/> [Hozzáférés dátuma: 2024. 05. 28.]
- [48] „Pointers” [Online] Elérhető: <http://www.cplusplus.com/doc/tutorial/pointers/> [Hozzáférés dátuma: 2024. 05. 28.]
- [49] „Pointer (computer programming)” [Online] Elérhető: [https://en.wikipedia.org/wiki/Pointer\\_\(computer\\_programming\)](https://en.wikipedia.org/wiki/Pointer_(computer_programming)) [Hozzáférés dátuma: 2024. 05. 28.]
- [50] „Dynamic Memory Allocation in C using malloc(), calloc(), free() and realloc()” [Online] Elérhető: <https://www.geeksforgeeks.org/dynamic-memory-allocation-in-c-using-malloc-calloc-free-and-realloc/> [Hozzáférés dátuma: 2024. 05. 28.]
- [51] „Dynamic Memory” [Online] Elérhető: <http://www.cplusplus.com/doc/tutorial/dynamic/> [Hozzáférés dátuma: 2024. 05. 28.]



## 6. Ábrajegyzék

1. ábra - Oldaltérkép .....	4
2. ábra - Lábléc .....	5
3. ábra - Navigációs sáv bejelentkezés előtt .....	5
4. ábra - Navigációs sáv bejelentkezés után .....	5
5. ábra - Bejelentkezés .....	6
6. ábra - Elfelejtett jelszó.....	7
7. ábra - Jelszó visszaállítás e-mail .....	7
8. ábra - Jelszó visszaállítása .....	8
9. ábra - Regisztráció .....	9
10. ábra - E-mail cím megerősítése .....	10
11. ábra - Megerősítési e-mail .....	11
12. ábra - E-mail cím megerősítve.....	11
13. ábra - Vezérlőpult.....	12
14. ábra - Saját profil .....	13
15. ábra - Profiladatok módosítása .....	13
16. ábra - Profiladatok módosítása, teljes név.....	14
17. ábra - Profiladatok módosítása, e-mail cím .....	14
18. ábra - Profiladatok módosítása, felhasználónév .....	14
19. ábra - Profiladatok módosítása, jelszó .....	14
20. ábra – Közösség, barátok .....	15
21. ábra - Közösség, barátkérelmek .....	16
22. ábra - Közösség, keresés .....	17
23. ábra - Profil, más felhasználó .....	18
24. ábra - Tananyag.....	19
25. ábra – Tananyag, kódszerkesztő és feladat gomb .....	19
26. ábra - Feladat elindítása .....	20
27. ábra – Feladat, első kérdések.....	20
28. ábra - Feladat eredménye .....	22
29. ábra - Feladat eredménye, összpontszám, sikeresség .....	22
30. ábra - Kijelentkezés .....	23

31. ábra - Komponens diagram .....	26
32. ábra - Drótvázterv, bejelentkezés .....	27
33. ábra - Drótvázterv, regisztráció .....	27
34. ábra - Drótvázterv, saját profil .....	27
35. ábra - Drótvázterv, felhasználónév megváltoztatása .....	27
36. ábra - Drótvázterv, e-mail cím megváltoztatása .....	28
37. ábra - Drótvázterv, jelszó megváltoztatása .....	28
38. ábra - Drótvázterv, közösség fül, barátok .....	28
39. ábra - Drótvázterv, közösség fül, barátkérelmek .....	28
40. ábra - Drótvázterv, közösség fül, keresés .....	29
41. ábra - Drótvázterv, leckék fül .....	29
42. ábra - Drótvázterv, leckék fül, oldal alja .....	29
43. ábra - Drótvázterv, kvíz .....	29
44. ábra - Drótvázterv, kvíz eredmény .....	30
45. ábra - Használati eset diagram, autentikáció .....	31
46. ábra - Használati eset diagram, vezérlőpult .....	31
47. ábra - Használati eset diagram, navigációs sáv .....	32
48. ábra - Használati eset diagram, saját profil .....	33
49. ábra - Használati eset diagram, más felhasználó profilja .....	33
50. ábra - Használati eset diagram, közösség fül .....	34
51. ábra - Használati eset diagram, leckék és feladat rész .....	35
52. ábra - Megjelenítésért felelős resources könyvtár .....	36
53. ábra - Az adatbázis sémája .....	40
54. ábra - A model osztálydiagramja .....	41
55. ábra - A controller osztálydiagramja .....	44
56. ábra - A Console/Command mappában lévő osztályok osztálydiagramja .....	45
57. ábra - A Middleware osztályok osztálydiagramja .....	46
58. ábra - Az Exception handler osztálydiagramja .....	47

## 7. Felhasznált technológiák

### 7.1. Backend

#### 7.1.1. Production

- PHP: v8.2
- ext-pdo
- inertiajs/inertia-laravel v1.0
- laravel/framework v11.0
- laravel/tinker v2.9
- nyholm/psr7 v1.8
- symfony/psr-http-message-bridge v7.0

#### 7.1.2. Production és development

- PHP: v8.2
- ext-pdo
- inertiajs/inertia-laravel v1.0
- laravel/framework v11.0
- laravel/tinker v2.9
- nyholm/psr7 v1.8
- symfony/psr-http-message-bridge v7.0
- fakerphp/faker v1.23
- laravel/dusk v8.2
- laravel/pint v1.15
- laravel/sail v1.26
- laravel/telescope v5.0
- mockery/mockery v1.6
- nunomaduro/collision v8.0
- phpunit/phpunit v11.0.1
- spatie/laravel-ignition v2.4

## 7.2. Frontend

### 7.2.1. Production

- @codemirror/commands v6.3.3
- @codemirror/lang-cpp v6.0.2
- @codemirror/lang-java v6.0.1
- @codemirror/lang-javascript v6.2.2
- @codemirror/lang-python v6.1.6
- @codemirror/language v6.10.1
- @codemirror/state v6.4.1
- @codemirror/theme-one-dark v6.1.2
- @codemirror/view v6.26.3
- @fortawesome/fontawesome-svg-core v6.5.2
- @fortawesome/free-solid-svg-icons v6.5.2
- @fortawesome/react-fontawesome v0.2.0
- @inertiajs/react v1.0.16
- @lezer/highlight v1.2.0
- codemirror v6.0.1
- react-markdown v9.0.1
- react-modal v3.16.1
- react-tabs v6.0.2
- react-tooltip v5.26.3
- vite-plugin-laravel v0.3.1

### 7.2.2. Production és development

- @codemirror/commands v6.3.3
- @codemirror/lang-cpp v6.0.2
- @codemirror/lang-java v6.0.1
- @codemirror/lang-javascript v6.2.2
- @codemirror/lang-python v6.1.6
- @codemirror/language v6.10.1
- @codemirror/state v6.4.1
- @codemirror/theme-one-dark v6.1.2
- @codemirror/view v6.26.3
- @fortawesome/fontawesome-svg-core v6.5.2
- @fortawesome/free-solid-svg-icons v6.5.2
- @fortawesome/react-fontawesome v0.2.0
- @inertiajs/react v1.0.16
- @lezer/highlight v1.2.0
- codemirror v6.0.1
- react-markdown v9.0.1
- react-modal v3.16.1
- react-tabs v6.0.2
- react-tooltip v5.26.3
- vite-plugin-laravel v0.3.1
- @babel/plugin-proposal-class-properties v7.18.6
- @babel/plugin-syntax-dynamic-import v7.8.3
- @babel/preset-react v7.24.1
- @vitejs/plugin-react v4.2.1
- autoprefixer v10.4.19
- axios v1.6.4
- babel-loader v9.1.3
- laravel-vite-plugin v1.0.2
- postcss v8.4.38
- react v18.2.0

- react-dom v18.2.0
- resolve-url-loader v5.0.0
- sass v1.74.1
- sass-loader v12.6.0
- tailwindcss v3.4.3
- vite v5.2.10

### 7.3. Felhasznált programok, IDE-k

- PHPStorm 2024.1
- Doxygen v1.10.0
- draw.io v24.1.0
- Github Desktop v3.3.6-linux3 (x64)