

# **Reszponzív web alkalmazások tervezése, megvalósítása és tesztelése**

Konzulens: Ekler Péter

# **Analízis**

Gema Szabolcs  
Husztai Martin

BAE7J  
OCUW5I

# 1. Bemutatása

Alkalmazásunk a BME Analízis c. tantárgy weboldalát hivatott modernebbé alakítani, reszponzív web felület, és korszerű adatbázis kezelés segítségével. Egyaránt használható, diákként, tanárként, vagy éppen tárgyfelelős(admin) -ként is.

**Mérnök informatikus szak, analízis (1), BSc (BMETE90AX21, BMETE90AX04)**

[Követelményrendszer \(Kötelező olvasmány!!!\)](#)

[Tételesor, tematika, tematika heti bontásban](#)

[Régi zárthelyi és vizsga feladatsorok](#)

[Mateking \(egy gyakorló oldal\), Mateking ismertető](#)

---

[Zárthelyi eredmények és aláírás \(név és Neptun kód nélkül\)](#)

Mindenki saját korábbi pontjainak ismeretében jó eséllyel be tudja azonosítani magát.  
Jegyhatárok: 40%, 55%, 65%, 80%.

---

[Előadások és gyakorlatok órarendi adatai](#)

---

**Előadások**

Komplex számokhoz anyag: [Barabás B., Fulop O.: Építészek matematikája II.](#), 2. fejezet.  
Előadásjegyzet: [színes változat](#), [fekete-fehér változat](#)

A gyakorlatanyag és az előadásanyag kölcsönösen hivatkozik egymásra. A hivatkozás akkor működik, ha pdf file-ok azonos könyvtárba, változatlan névvel kerülnek elmentésre, vagy a böngészőbe ágyazott pdf nézegetőt használunk.

---

**Gyakorlatok**

[A gyakorlatokra tervezett feladatok](#)  
Komplex számokhoz feladatok  
Gyakorló feladatok: [színes változat](#), [fekete-fehér változat](#)  
IMSC (emelt szintű) gyakorlatra szánt feladatok (folyamatosan bővülő pildatár).  
Nagy Ilona [Matematika A](#) honlapján található [gyakorlatanyagot](#) is ajánljuk gyakorlásra, különösen a 10. feladat sorban található szöveges szélsőérték-feladatokat, melyekhez tartozik [megoldás](#) is.

A gyakorlatanyag és az előadásanyag kölcsönösen hivatkozik egymásra. A hivatkozás akkor működik, ha pdf file-ok azonos könyvtárba, változatlan névvel kerülnek elmentésre, vagy a böngészőbe ágyazott pdf nézegetőt használunk.

---

**Zárthelyik**

[0. Zárthelyi és pótlásai](#)

Hírek
Számonkérés
Információ
Feladatok
Követelmény
Profil

Login
Adatok
Jelszó változtatás
Chat
Diákok
Új hír közzététele
Új felhasználó
Követelmények
Hír törlése
Új gyakorlat
Új számonkérés
Gyakorlat változtatás
Kijelentkezés

Diákok
Új jegy beírása

MARTIN:	Elso Vizsga	100 pont	5
	Elso Vizsga	101 pont	6
	Elso Vizsga	1 pont	1
	ANALIZIS VISZAG2	45 pont	2
	ANALIZIS VISZAG2	200 pont	10
	Elso Vizsga	32 pont	32
GEMA:	asds	999 pont	12
	Elso Vizsga	12 pont	1
	Elso Vizsga	0 pont	0
		587 pont	57
student:	asds	23 pont	2
	Elso Vizsga	23 pont	2
	ANALIZIS VISZAG2	444 pont	444

## 2. Fejlesztés

# Verziókezelés

- **Git** - [Github](#)

Verziókezeléshez GitHub-ot használtunk. Legelején egy master, illetve egy react-try branchen dolgoztunk, majd a későbbiekben áttértünk arra, hogy minden új feladatra külön branchet csináltunk és pull request-el mergeltük össze a fő ággal.



Feb 3, 2019 – May 14, 2019

Contributions: Commits ▾

Contributions to master, excluding merge commits



## **Backend technológiák**

- **IntelliJ Idea**

Backend fejlesztő környezetének leginkább az *IntelliJ-t* használtunk, amivel könnyen szerkeszthettük a Java forrásfájlokat, illetve futtathattuk a *Spring-Boot* applikációinkat.



- **Spring Boot**

Ezen keretrendszeren alapszik backend szolgáltatásunk. Ehhez is sok elérhető tutorial oldal volt található az interneten.



- **Apache Maven**

Programunk menedzselésére használtuk. Különböző dependency-eket injektáltunk, illetve saját build folyamatot írtunk, hogy össze tudjuk csomagolni a frontendet a backenddel.



- **Project Lombok**

Lombok Java könyvtár segítségével egyszerűen adhattunk gettereket, settereket Java forráskódunkhoz, boilerplate kódok elkerülése, és fejlesztésünk felgyorsítása végett.



- **MongoDB - MongoDB Atlas**

Konzulensünk ajánlásával MongoDB-t használtunk adatbázisként. Ez egy teljesen új platform volt számunkra, de könnyen bele lehet jönni, és egyszerűnek éreztük a használatát. Saját MongoDB Atlas szolgáltatását használtuk ami egy felhőbe integrált adatbázis, így, bárholnan elérhető.



- **Postman**

Segítségével egyszerűen tudtuk tesztelni az API-t, így gyorsabban kiderült, hogy mi és hogyan működik vagy éppen nem működik. Nem kellett frontend részen kódolni teszteléshez.



- **Swagger**

Segítségével írhatjuk le, és gyűjthetjük össze az API kéréseinket. Deployoltuk a Swagger-UI-t is Maven segítségével, így elérhető publikusan az összes adható REST kérés a backendhez, leírással ellátva, modellekkel együtt.



- **Clean Code RESTful API**

Próbáltuk a REST kéréseket clean code elvek szerint elkészíteni, így átláthatóan használjuk a REST összes metódusát (GET, PUSH, DELETE...), a (pl. /api/news, vagy /api/students).

**{ REST }**

news-controller Hírek kezeléséért felelős controller		
GET	/api/news	Összes hírt vissza adja
POST	/api/news	Új hírt ad hozzá
DELETE	/api/news	Adott hírt töröl Id alapján. News modellt vár

News ▾ {	
description:	Hírek modell
id	string Generált Id. Nem kell kitölteni
text	string Hír szövege
title	string Hír címe
}	

# **Frontend Technológiák**

- **Visual Studio Code**

A frontendet VS Code segítségével valósítottuk meg. Ez szerintünk jobban támogatja a JavaScript nyelvet, illetve React-hez használt kiterjesztéseket, így a fejlesztés gyorsabban ment vele, és szebben indentált kódot készíthettünk.



- **Figma**

A design-t terveztük meg vele. Ez egy eléggé jól ismert vektorgrafikus design-t készítő felület, ahol logikusan lehet felépíteni a frame-eket, illetve lehet anchor pointokat használni, így más tervezés közben kiderülhet, hogy mutat a design. Akár egyszerre is lehet rajta dolgozni, hiszen elérhető a felhőben.



- **InVision**

A Figmás projektünket tudtuk "live"-ra kapcsolni, így interaktívan tudtuk bemutatni tervünket. Külön frame-line-t lehet készíteni vele, kommentekkel, illetve reaktív felületekkel tűzdelve, hogy egy valós élményt adjon vissza.



- **React**

Segítségével építhettük egyszerűen a User Interfészt. Szintén egy teljesen új technológia volt számunkra, de sok tutorial oldal segítségével sikerült belejönni.



- **Node Packages**

- reactstrap

Leginkább Bootstrap-es Gombokat, Alerteket, Selecteket használtunk vele.

- material-ui

Ikonokat innen importáltunk.

- axios - react-talk - react-stomp  
Segítségével működtetjük a chat felületünket.
- react-router  
Ezzel a modullal navigálunk az oldalak között.

## **Continuous Integration**

- **Travis CI - [Travis Link](#)**

Travissal automatizáljuk a buildet.  
Lépések:



1. Maven segítségével összecsomagolja a frontendet a backenddel
2. Felrakja Docker Hub-ra a Dockerfile segítségével
3. Végül Heroku-ra is deployol a Procfile segítségével

- **Heroku - [Heroku Link](#)**

Segítségével sikerült felrakni az internetre a teljes web alkalmazásunkat. Míg a Netlify esetén csak a frontend, addig Heroku-nál a backend is működött egyszerre.



- **Docker - [Docker Repository Link](#)**

Docker image-ként is elérhető az alkalmazásunk, így aki rendelkezik docker-el könnyűszerrel tudja futtatni azt.



- **Netlify**

Segítségével sikerült felrakni a frontendet a világhálóra. Sajnos az backendet nem érte el, ezért hamar el kellett vetni ezt a megoldást.



- **Codacy** - [Codacy Link](#)

Codacy ellenőrzi a kódunkat többféle szempontból is, amit mi megpróbálunk minimálisan tartani.



## **Egyéb**

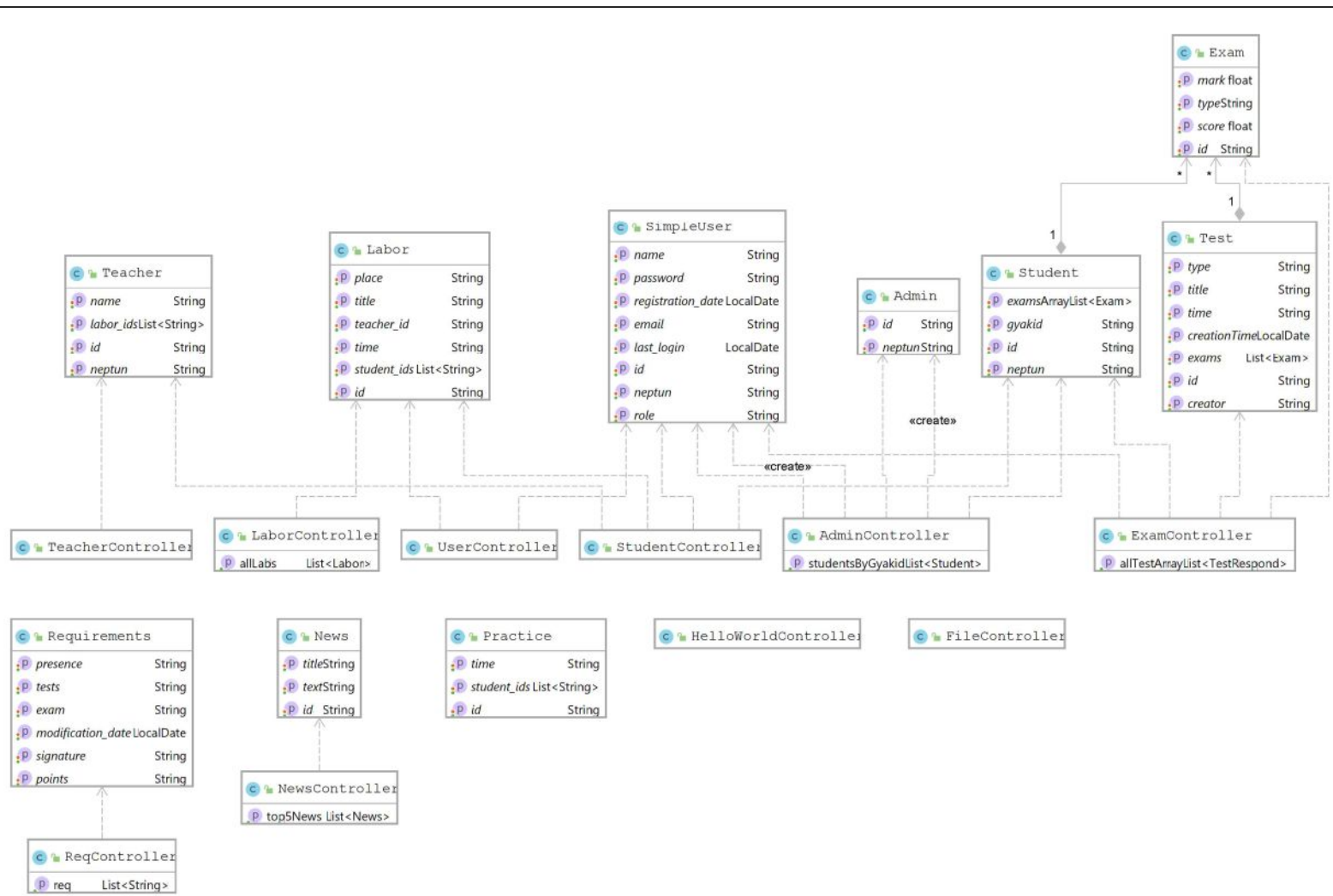
- **Adobe Illustrator**

Az oldal logóját terveztük meg vele. A későbbi felhasználhatóság szempontja végett, vektor grafikus logóban gondolkodtunk.





### **3. Adatmodell**



## 4. Funkciók

### **Bejelentkezés**

Alkalmazásunk egy bejelentkezési lehetőséggel köszönt minket. Innen nem tudunk elnavigálni másik oldalra, mert a react router visszairányít a bejelentkezés formra. Alertdialog-al jelzi ha hibás a megadott Neptun kód vagy a jelszó.

The screenshot shows the application's navigation bar with links: Hírek, Számonkérés, Információ, Feladatok, Követelmény, and Profil. Below the bar, a pink error message box states: "Sikertelen bejelentkezés! Rossz neptun kód vagy jelszó." Below this is the "Bejelentkezés" form. It contains two input fields: "Neptun kód:" with the text "rossz" and "Jelszó:" with masked characters "\*\*\*\*\*". There is a checkbox labeled "Maradjak bejelentkezve" and a "Belépés" button.

### **Bejelentkezés után**

Bejelentkezés után, a profilba találjuk magunkat, ahol különböző menük között választhatunk attól függően hogy milyen privilégiummal rendelkező felhasználóval léptünk be (student, teacher, admin).

### **Hírek**

Az admin felvehet híreket, amik megjelennek a hírek oldalon. Ha rákattintunk egy popupba megjelenik nagyba a hír. Ezeket a híreket természetesen törölni is tudja, külön menüpontba szedve

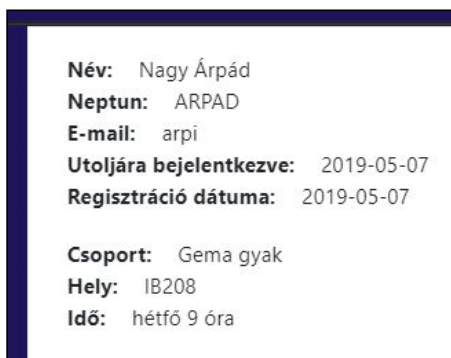
The screenshot shows the "Hírek" form. It has two input fields: "Hír címe" (News title) with the text "Előadás Információ" and "Hír szövege" (News text) with the text "Elmarad a következő előadás. Kellemes pihenést! Martin". There is a "Feltöltés" (Upload) button.

The screenshot shows the "Hírek" list. It has a table with two columns: "Előadás Információ" and "Első hír". The first row shows "Elmarad a következő előadás. Kellemes pihenést! Martin ..." under the first column and "A szöveg ..." under the second column.

The screenshot shows a modal dialog titled "Előadás Információ". It contains the text "Elmarad a következő előadás. Kellemes pihenést! Martin".

## Adatok

Bármelyik felhasználó meg tudja nézni saját adatait. Specifikusan, ha mondjuk egy oktató nyitja meg, akkor láthatja a saját gyakorlatait is amit ő tart.



Név: Nagy Árpád  
Neptun: ARPAD  
E-mail: arpi  
Utoljára bejelentkezve: 2019-05-07  
Regisztráció dátuma: 2019-05-07  
  
Csoport: Gema gyak  
Hely: IB208  
Idő: hétfő 9 óra

## Jelszó változtatás

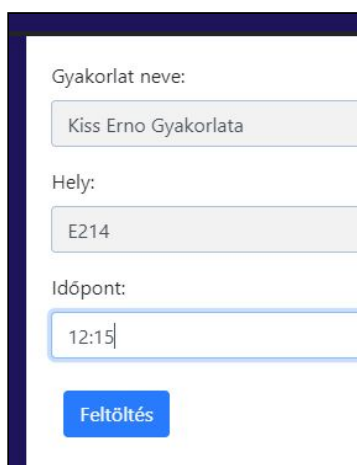
A menüben lehetőségünk van megváltoztatni jelszavunkat. Ellenőrzi, hogy a két jelszó egyezik-e és csak ezután küldi el a szervernek.



Régi jelszó: .....  
Új jelszó: .....  
Megerősítés: .....  
  
Jelszó változtatás

## Gyakorlat felvétele

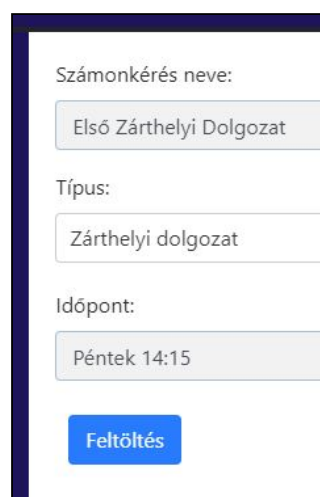
Admin felvehet új gyakorlatot aminek megadhatja a nevét, helyét és idejét. A felvétel eredményét majd látja a felvevő.



Gyakorlat neve:  
Kiss Erno Gyakorlata  
  
Hely:  
E214  
  
Időpont:  
12:15  
  
Feltöltés

## Számonkérés felvétele

Admin felvehet új számonkérést is aminek megadhatja a nevét, típusát, és idejét. A felvétel eredményét majd látja a felvevő.



Számonkérés neve:  
Első Zárthelyi Dolgozat  
  
Típus:  
Zárthelyi dolgozat  
  
Időpont:  
Péntek 14:15  
  
Feltöltés

## Új felhasználó felvétele

Adminként, illetve gyakorlati vezetőként új diákot vehetünk fel akinek megadhatjuk nevét, e-mail címét, neptunját illetve hogy milyen gyakorlatra fog járni.

Felvehetünk még Gyakorlatvezetőt adminként, akinek megadhatjuk azt, hogy milyen gyakorlatot fog vezetni. Itt többet is választhatunk.

Melyik gyakorlatot tartja:

Kiss Erno Gyakorlata ✕ Második Gyakorlat ✕

Regisztrálás

Természetesen admint is felvehetünk, mint új felhasználót.

Felhasználó felvétele:

Diák

Diák neve:

Husztí Martin

E-mail:

martinka@gmail.com

Neptun:

MARTIN

Gyakorlat:

Kiss Erno Gyakorlata

Regisztrálás

Tallózás...

## Diákok

Minden magasabb rangú felhasználónak van egy olyan menüpontja hogy "Diákok". Itt láthatja a gyakorlataihoz rendelt diákjait (Admin mindent lát).

Diákok				Új jegy beírása
MARTIN:	Első Zárthelyi Dolgozat	80 pont	5	
	Első Zárthelyi Dolgozat	32 pont	2	
BALAZS:	Első Zárthelyi Dolgozat	2 pont	1	
	1. Pót Zh	99 pont	5	
lencsi:	1. Pót Zh	99 pont	5	
balazs:	1. Pót Zh	1 pont	1	

## Követelmények

Tudjuk változtatni a követelményeket is plain text formájában.

Hírek Számonkérés Információ Feladatok Követelmény

Követelmények

**Jelenlét:**  
Maximum 2x hiányzás a gyakorlatról

**Számonkérések:**  
A félév során 3 db

**Aláírás:**  
Feltétel a 2 sikeres ZH

**Pontszámítás:**  
20-tól kettes

**Vizsga:**  
40 pontos, teljesíteni kell

Teljes követelmény letöltése

Jelenlét

Maximum 2x hiányzás a gyakorlatról

Aláírás

Feltétel a 2 sikeres ZH

Vizsga

40 pontos, teljesíteni kell

Számonkérések

A félév során 3 db

Pontszámítás

20-tól kettes

Változtatás

\*Csak lényeges, rövid infók kerüljenek a szövegdobozba

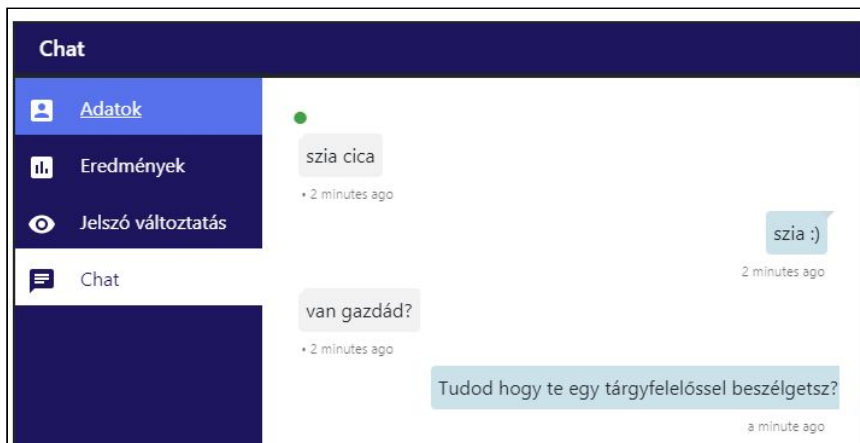
## Gyakorlat változtatás

Lehetőségünk van egy diákot áthelyezni más gyakorlatra egyszerűen.

Diák Neptun:
MARTIN
Gyakorlat:
Második Gyakorlat
<b>Változtatás</b>

## Chat

Lehetőségünk van chatelésre. Alapvetően egy TalkBox libraryt alkalmazunk, így könnyen ki lehetett váltani a megvalósítást



## **5. Jövőbeli tervek, továbbfejlesztési lehetőségek**

### **Token alapú autentikáció:**

- Token alapú autentikációt is bevezethetünk, mely segítségével biztonságosabb kommunikációt biztosíthatunk a backend és frontend között. Felmerült a JWT, illetve az OAuth.

### **Password hash:**

- Az egyszerű implementáció miatt a jelszavakat egyszerű plain textben tároljuk az adatbázisban. A későbbiekben a biztonság érdekében hashelve lenne célszerűbb tárolni, saltingot, illetve stretchinget alkalmazva rajta.

### **Szerver Limit:**

- Sok adat esetén túlterhelnek a klienst, illetve a szerveret az adatokkal (mondjuk 1000 vagy több diák), így beiktathatnánk egy limit változót amit a szerver felé küldhetünk, így csak annyi adatot adna vissza amennyi szükséges.

### **FTP szerver:**

- Továbbiakban akár lehetőség van FTP szervert kötni az alkalmazáshoz. Ezzel könnyebben lehetne kezelni a fájl kezeléseket, nevezetesen, ha egy jogosult felhasználó szeretne valamilyen dokumentumot fel, vagy letölteni

### **EMAIL:**

- Lehetne egy központi e-mail cím, ami az adott felhasználónak értesítést küld, ha valamilyen adat-a megváltozna, vagy új adatkötésben venne részt. illetve ezt lehetne használni, hogy pl: hírekről értesítést adjon mindenkinek.
- Regisztrációval e-mailes megerősítést is lehetne bevezetni.

### **JUNIT:**

- UI, illetve backend tesztelésre lehetne JUnit teszteket írni, egy elkerülnénk különböző error-okat, hibákat, amire alapesetben nem biztos hogy gondolnánk.

### **Dinamikus Téma:**

- Felhasználó tudná állítani a témáját, attól függően melyet szeretne alkalmazni. Az ilyen adatokat lehetne cookie-ban tárolni.

**Felhasználói szintek:**

- Attól függően, hogy egy embernek milyen jogosultsága van, más jelenne meg. Ez abban különbözne a mostanitól, hogy, ha valaki pl egyszerre oktató, és diák is, akkor neki mindkettő jogosultsághoz tartozó menük, opciók megjelennének, és nem kéne 2 külön felhasználó

**Feladat megoldó:**

- Lehetne új menü pont, ahova oktató online-interaktív feladatokat tehetne közzé, gyakorlás, illetve házi feladat kapcsán, amit diákok utána kitölthetnek, és esetleg ennek eredménye megjelenne a profiljukban.

**Neptun alapú szinkronizáció:**

- Neptunnal lehetne szinkronizálni. Tehát minden engedett neptunkódhoz tartozna felhasználó, ami az alapvető adatokat onnan szedné le, illetve visszafelé is működne. Így pl.: Jegyfelvételnél megjelenne neptunba is automatikusan az eredmény.

**Saját fiókban jegyzet:**

- Diáknak lenne saját tárhelye ahova bizonyos formátumú fájlokat tölthetne fel, és érhetne el a későbbiekben.

**Stream:**

- Ha későbbiekben stream-elve lenne az óra, vagy csak videóra véve, akkor el lehetne érni az anyagot, és közvetlen megtekintani.

**Forum:**

- Üzenőfal, ahol kérdéseket lehetne felrakni, illetve más tudna rá válaszolni. Ez későbbiekben segítené feladat megoldás illetve készülés terén a diákoknak, és lehetőséget adna, hogy különböző anyagokat beszélhessenek át. Ez abban különbözne a chat-től, hogy lehetne témákat, topicokat nyitni, és utána is megmaradna.

**Summary**

- bevezetése, így ha nagyobb projektre nőné ki magát, akkor könnyebb lenne kezelni, fejleszteni, hisz megkönnyítené a fejlesztők munkáját.

- Mindemellett lehetne custom hibakódokat készíteni.