A képen szöveg látható

Automatikusan generált leírásA képen szöveg, clipart látható

Automatikusan generált leírás

Tanszéki kurzuskiírási rendszer

**Projektmunka 2.**

(GKNB\_INTM005)

Zöllei Alex

CM6PBK

Varga Szabolcs

H7NG5S

Bogdán Ádám

XL6KK5

Szikonya Attila

HWBWRN

# Projekt tervezés

## 1.1 Feladatspecifikáció

A projekt célja egy tanszéki kurzuskiírási rendszer fejlesztése. Ebben megadhatónak kell lennie, milyen tárgyakat fog meghirdetni a tanszék a következő félévre, beleértve az előadás és gyakorlati kurzusok összes adatát (darabszám, oktatók hozzárendelése és terhelésaránya, óraszámok, hozzárendelt képzések, azokon a tárgy kötelező/diff/szabvál jellege, létszámkorlát, teremigény, egyéb oktatói igények, tárgyösszevonások). A megadott adatokat tartósan tárolni és visszakeresni kell tudni félévenként. Az elvégezhető tevékenységeknek felhasználói csoportonként eltérően szabályozhatónak kell lennie. Félévente és az utolsó két félév átlagában megállapíthatónak kell lennie az oktatói terheléseknek, amelyeket számszerűen és grafikusan is ábrázolni kell abszolút értékben és a besoroláshoz tartozó törvényi előírásokhoz mérten százalékosan is. A szoftvernek a weben elérhetőnek kell lennie.

## 1.2 Felépítés, szerkezet

Az oldalt belépés/ azonosítás után lehet megtekinteni. A rendszer kezdőlapja a kurzuskiírási oldal, ami egy listát tartalmaz. A weboldal fix elemei közé tartozik egy horizontális navigációs sáv az oldal tetején, ami a kurzuskiírás, statisztika, beállítások és egy kijelentkezés elemet tartalmaz.  Egy bal oldali vertikális navigációs sáv oldalanként eltérő elemeket tartalmaz.

**Kurzuskiírás:** A bal oldali navigációs sávnak a tetején jelenik meg kurzuskiírásnál a szűrési mező. Alatta ki lehet választani melyik félévnek a kurzusait és képzéseit akarjuk megtekinteni, de alapértelmezetten az oldal tartalma az aktuális félévben létrehozott tantárgyakat tartalmazza. Lejjebb a verzióelőzmény található.

Az 1-es kép szemlélteti a kurzuskiírás megjelenítését. A lista első mezejébe a tantárgy kódja kerül, a másodikba pedig a tantárgy neve. Az adott tantárgy mezejének háttérszíne alapból szürke színű és a tantárgy állapotát egy piros ’ X ’ jelöli. Amint megfelelővé válik, vagyis minden előzőleg meghatározott követelményeknek megfelel, az állapotnál egy zöld pipa jelenik meg, illetve a tantárgy háttérszíne zöldre vált. Az utolsó mezők tartalmazzák a szerkesztési lehetőséget, a tantárgy állapotát, hogy megfelelt-e minden elvárásnak és bekerült-e a neptun rendszerbe. Végül minden sor végén egy nyilat láthatunk, amire rákattintva lenyílik egy további ablak. A listák lenyitható füllel rendelkeznek, ahol több információt tudhatunk meg az aktuálisan kiválasztott tantárgyról. Itt megtaláljuk, hogy hány óra tartása van előírva az adott tantárgyhoz, ezen belül az is, hogy hány előadást és gyakorlatot fognak tartani. Az óraszámok alatt láthatunk egy újabb listát, ahol az oktatók neveit látjuk, illetve minden hozzájuk és a kurzushoz kapcsolódó adatokat látunk egy táblázatban. Például a tanárok terhelését, teremszámot, kurzuskódot, heti óraszámot, valamint mindegyik kurzushoz megjegyzés is fűzhető.



1. ábra: Tervezett kurzuskiírás megjelenítése

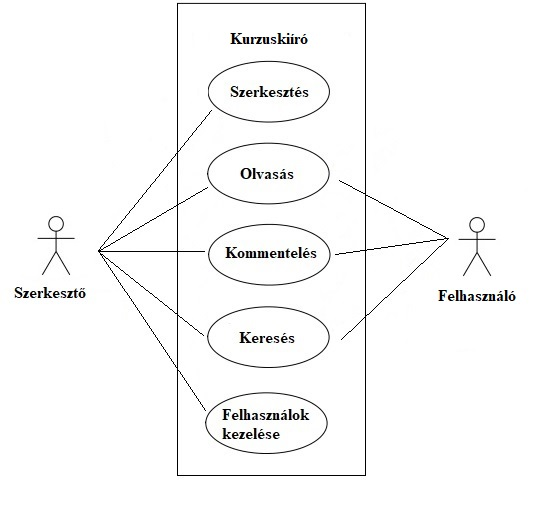
**Mintatantervek:** Ahhoz, hogy a felhasználó tárgyat és hozzá kurzust tudjon kiírni a tárgynak szerepelnie kell a tanszék egyik mintatantervjében. A mintatatanterveket egy külön oldalon lehessen kezelni, hozzáadni és törölni.

**Statisztika:** Külön oldal az oktatók leterheltségének az összesítésére/ ábrázolására nappali és levelező óraszámokkal. Itt százalékosan ábrázolja az oldal, hogy a kötelező minimumhoz képest mit teljesített az oktató. (táblázatos forma, tortadiagram). További statisztikák megjelenítése: Utolsó két félév átlagában  hány órát tartott az adott oktató.

**Beállítások:** Ezen a felületen a szerkesztő be tudja állítani a többi felhasználó jogosultságát, valamint tudja törölni az accountokat is. A saját felhasználói fiókjához tartozó adatokat mindenki saját maga tudja szerkeszteni.

**Adminisztráció:** Ezen a felületen

## 1.3 Usecase



2.. ábra: Usecase ábra

## 1.4 Funkciók, szűrések

**Tárgyak felvétele és módosítása.** Új félév nyitásakor lehetőség van az előző évi (azonos félév) kurzuskiírást, betölteni, és csak a változásokat módosítani. Ez a funkció  jelentősen megkönnyíti a szerkesztő munkáját. A tárgyakat manuálisan is fel lehet vinni.Egy tárgyat csak akkor lehet felvenni, ha az a hozzárendelt mintatantervben szerepel, ezzel meggátolva azt, hogy esetleg egy nem létező tárgy kerüljön felvételre. A rendszer a tárgykódnak megfelelően automatikusan kitölti a tárgy nevét, a gyakorlati és elméleti óraszámot. A mintatervből az adott tárgyhoz lekérve a szakokat egy listát hoz létre, amiből aztán a felhasználó választja ki melyik szakoknak legyen meghirdetve a tárgy. A kurzuskódot és a terhelést a felhasználó gépeli be. Ezeknél szűrésekkel megakadályozható, hogy rossz adatokat adjon meg a felhasználó. A kurzuskódnál csak pozitív egész számokat 20-ig, terhelésnél 100-ig tartó értéket lehet megadni. Ezen felül az azonos kurzuskódoknál ellenőrizve van, hogy a terhelések összege 100%, ha nem akkor figyelmezteti a felhasználót. Nem enged olyan állapotot elfogadni, ami ellentmondást tartalmaz például 0 gyakorlati óránál nem lehet gyakorlati kurzust felvinni. Mivel az oktatók is ismertek, ezért korlátozásként csak olyan oktatót lehet egy kurzushoz felvenni, amelyiket már előtte felvettünk a rendszerbe. Ezt nagy valószínűséggel az oktatókhoz elkészített statisztikánál lehet majd megtenni. Terem- és szoftverigényhez tetszőleges szöveg adható meg. A létszám módosításakor csak 0 és 1000 közötti egész számokat lehet megadni.

**Szűrés, keresés:** Oktató szerinti keresést lehet végezni, ami visszaadja az általa oktatott tárgyakat, a leterheltséget, valamint, hogy a héten hány órája van az összes képzésen.

**Verziókövetés, előzménytörténet:** Lehet követni ki módosított min és mikor. Korábbi verzióra visszaállítás is lehetséges. Ha visszaállít valamit a felhasználó az egy újabb bejegyzésként jelenik meg az előzményekben.

**Felhasználók kezelése:** Kétféle felhasználói szint bevezetése. A szerkesztő minden jogosultságot megkap. Szerkeszteni tudja az egész adatbázist, valamint fel tud venni új felhasználókat a második felhasználói szintbe. A szerkesztőknek van joga manuálisan regisztráltatni új felhasználót a rendszerbe. Ehhez egy külön beállítások oldalon van lehetősége. A második felhasználói körbe tartozók azok, akik valamilyen módon érintettek a táblázat tartalmában. Csak megtekinteni és megjegyzést fűzni tudnak a kiírt kurzusokhoz.(legalább a saját adatait). A statisztikákat, a beállítások oldalon a jogosultságkezelési felületet és az verzió előzményeket nem látják, valamint a szerkesztéshez szükséges elemeket sem.

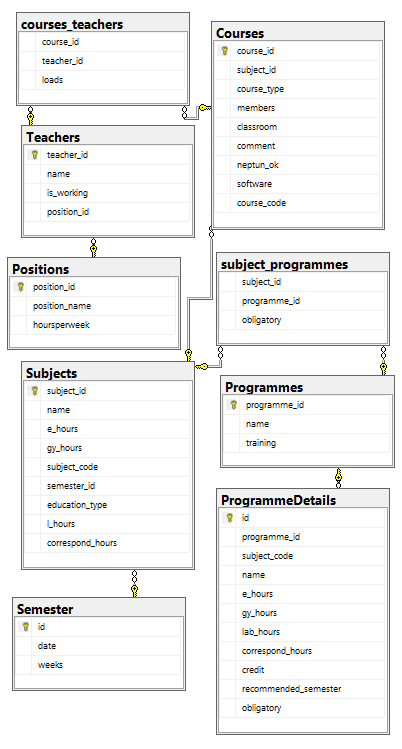
## 1.5 Feladatfelosztás

|  |  |
| --- | --- |
| **NÉV** | **FELADATKÖR** |
| Zöllei Alex | Új adatok felvitelének és módosításának megvalósítása, oldal megjelenésének tervezése, CRUD műveletek |
| Bogdán Ádám | Jogosultsági körök, statisztika elkészítése |
| Varga Szabolcs | SQL adatbázis tervezés és adatkezelés, bejelentkezés megvalósítása, authorization, CRUD műveletek |
| Szikonya Attila | Szűrés elkészítése, verziókövetés |

# Projekt fejlesztése

## Adatbázis felépítése

Az adatbázis megtervezésekor igyekeztünk lefedni a régi rendszererben (excel) előforduló minden rekordot. A kurzuskiíráshoz megtervezett adatbázis végső verzióját az alábbi diagramm szemlélteti.



3. ábra: Adatbázis felépítése

## Szemeszter

A „**Semester**” táblában tároljuk a szemeszterre vonatkozó adatokat. Mivel minden szemeszterben más tantárgyak és más kurzusok kerülnek kiírásra, így ez az adatbázis egyik fő eleme. A *date* mező tárolja magát a szemeszter nevét, a *weeks* mező az adott szemeszterben a szorgalmi heteket reprezentálja. Az alkalmazás kezdő oldalán lehet majd választani az itt letárolt szemeszterek közül.

## Tantárgyak

A „**Subjects**” táblában tároljuk a tantárgyakra vonatkozó adatokat. A tantárgy nevét a *name*, tantárgykódját a *subject\_code*, elméleti óraszámát a *e\_hours*, gyakorlati óraszámát a *gy\_hours*, a laborok óraszámát a *l\_hours* mezőben tároljuk. A levelezős szak miatt bevezettük a *correspond\_*hours mezőt is, hiszen itt csak összesített órákkal kell dolgozni. Az *education\_type* mezőben tárolva van az, hogy milyen a tantárgy oktatási típusa. A *semester\_id* idegen kulccsal kapcsolódik a *Semester* táblához. Több a többhöz kapcsolattal kapcsolódik a *Programmes* táblához.

## Mintatantervek

A „**Programmes**” táblában tároljuk le a tanszéken a mintatantervek neveit. A *name*  mezőben a mintatanterv nevét (évszámmal eggyütt) és a képzési szintet is letároljuk, a *training*-ben a képzési formát. Ez a tábla a **subject\_programmes** kapcsolótáblával van összekapcsolva a *subject* táblához. Ebben az idegenkulcsokon kívül az is le van tárolva, hogy kötelező-e azon a szakon a meghírdetett tantárgy (*obligatory*). Ez lehet szinkron, asszinkron, és hagyományos.

## Mintatanterv adatok

A „**ProgrammeDetails**” tábla tartalmazza a Programmes-hez tartozó adatokat. Ezekhez a mezőket az it.sze.hu –n elérhető mintatantervek adatai alapján vettük fel.

## Kurzusok

A „**Courses**” tábla több az egyhez kapcsolattal kapcsolódik a *Subjects*-hez, mivel minden tantárgyhoz több kurzus tartozhat. A kurzus típusát a *course\_type*  mezőben, a meghírdetett létszámot a *members*, a termet, ahol a kurzus megtartásra kerül a *classroom*, a szoftverigényt a *software*, a kurzus óraszámát a *hours*, a kurzus kódját, pedig a *course\_code* mezőben kerülnek tárolásra. A *neptun\_ok* mező jelzi, hogy a kurzus be lett-e már írva a neptun, vagy még nem. A kiírandó kurzusokhoz kommentet is lehet fűzni, ezt a *comment* tárolja. Ezen kívül a tábla több a többhöz kapcsolatban van a *Teachers* táblával, mivel egy oktatónak több kurzusa van, és egy kurzust is több oktató taníthat. A kapcsolótáblában a *loads* mezőben tárolva van, hogy egy tanár egy kurzusnál hány százalékban van leterlhelve.

## Tanárok

A „**Teachers**” tábla tartalmazza a oktatók adatait. A *name* mezőben az oktató nevét, az *is\_working* mezőben pedig az tároljuk, hogy az oktató jelenleg dolgozik-e a tanszéken. A tábla a **courses\_teachers** kapcsolótáblával van összekapcsolva a *Courses* táblával. A kapcsolótáblában le van tárolva, hogy az adott oktatót hány százalékban terheli az adott tantárgy.

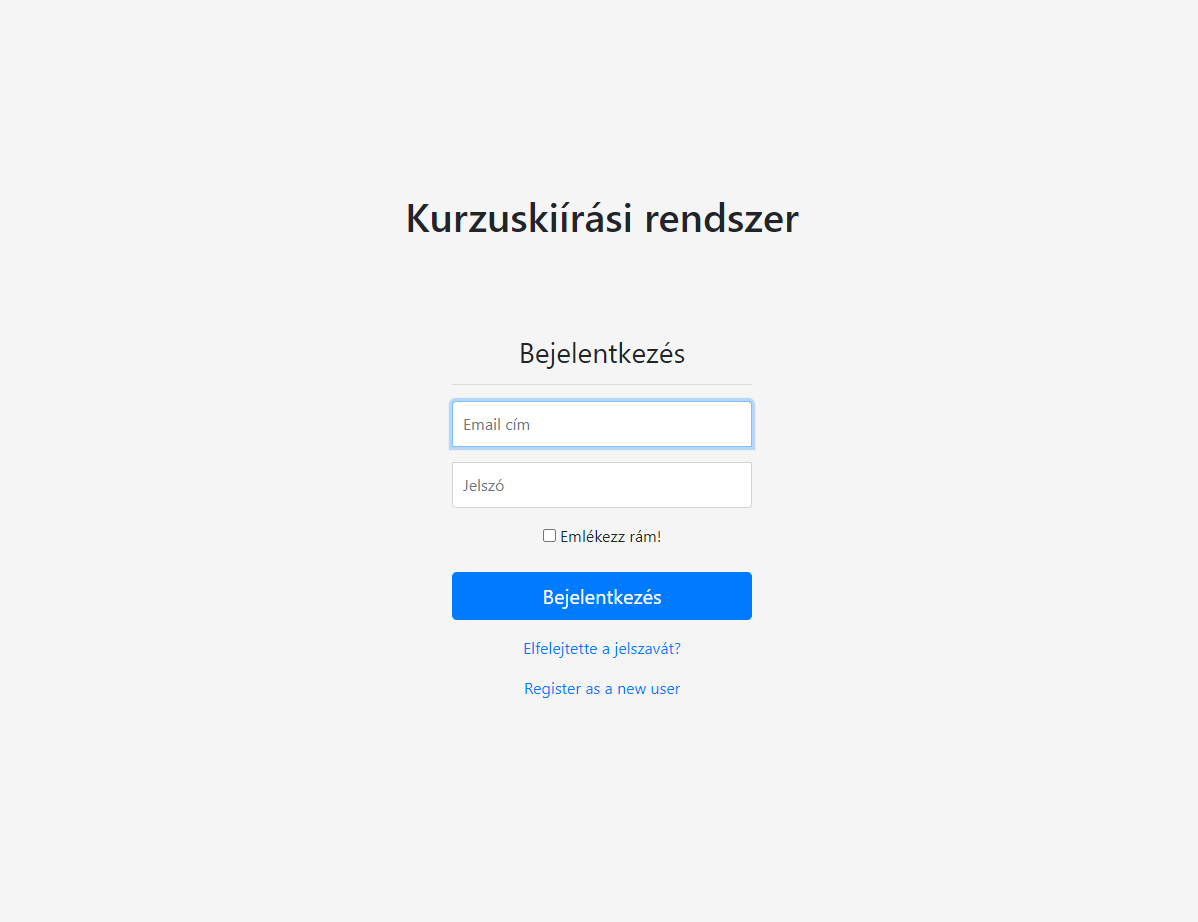
## Oktatói pozíciók

A „**Position**” táblába tartalmazza az oktatók lehetséges beosztásait. Itt tárolásra kerül a pozíció neve (*position\_name*) és az ehhez a beosztáshoz tartozó kötelező heti minimum óraszám(*hoursperweek*). Egy a többhöz kapcsolattal kapcsolódik a *Teachers* táblához.

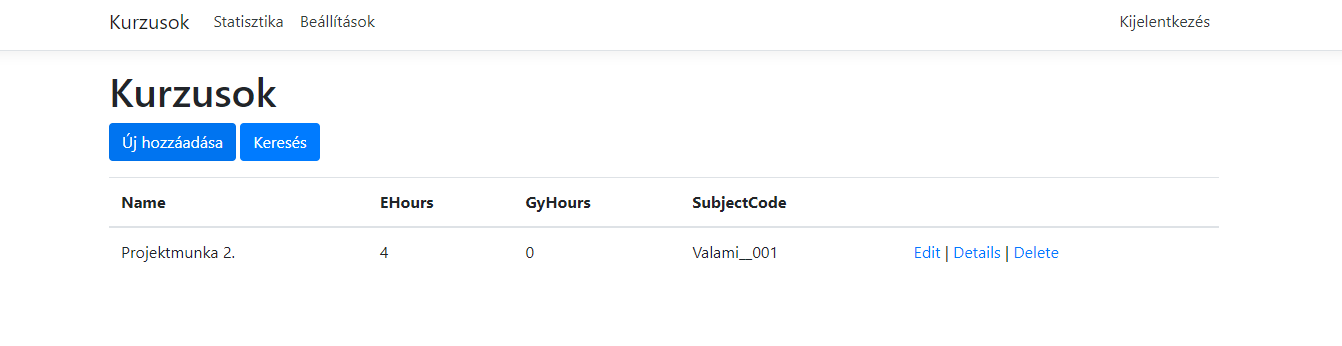
## 2.2 Első beszámoló, Szeptember 1 – Szeptember 5.

Csapatunk eddigi elért eredményeihez tartozik a keretrendszerek és használni kívánt programok kiválasztása és az ezekkel való ismerkedés. Számos technológiát kipróbáltunk, végül az ASP.NET Core MVC architektúrája mellett tettük le a voksunkat. Ezt full-stack fejlesztésre használjuk a frontend funkciót a beépített Razor látja el. Megterveztük és létrehoztuk az adatbázist, ahol a tantárgyak, a kurzusok, a tanárok és a szakok tárolódnak.

Az alkalmazás működő funkciója a tantárgyak megjelenítése az oldalon az adatbázisból. Tárgyak felvétele, módosítása az adatbázisba még csak tesztelés szintjén van megvalósítva. (CRUD funkciók), későbbiekben ezek teljesen átírásra kerülnek. Ehhez létrehoztunk egy szűrést is viszont ez még csak a tárgynév szerint dolgozik. Valamint létrehoztuk a kezdetleges statisztika funkciót, ahol a tanárok terheltségét lehet majd nyomon követni. A bejelentkezési felület is működik, azonban a regisztráció folyamatát még nem teljesen tisztáztuk, nem tudjuk, hogy kelljen-e validáltatni egy admin userrel. Amennyiben az adatbázisban még nem szerepel a belépni kívánt személy, regisztrálni kell. A regisztráció e-mailt kér, majd olyan jelszót, ami megfelel a kritériumoknak (nagybetű, hosszúság, különleges karakter). A kurzuskiírási oldal részeit (maga a kurzusok listája, navigációs sáv, beállítások, statisztika stb.) akkor érhető el, ha a felhasználó be van jelentkezve, különben csak a bejelentkezési oldal jelenik meg bármilyen URL elérés esetén. Az alkalmazás tokeneket használ, így érzékeli, ha valaki már be van jelentkezve. A beállítások menüpont alatt egyelőre csak a felhasználói fiókkal kapcsolatos dolgok találhatók, (e-mail cím, jelszó változtatás) az adminisztrációs felület még nem került implementálásra.



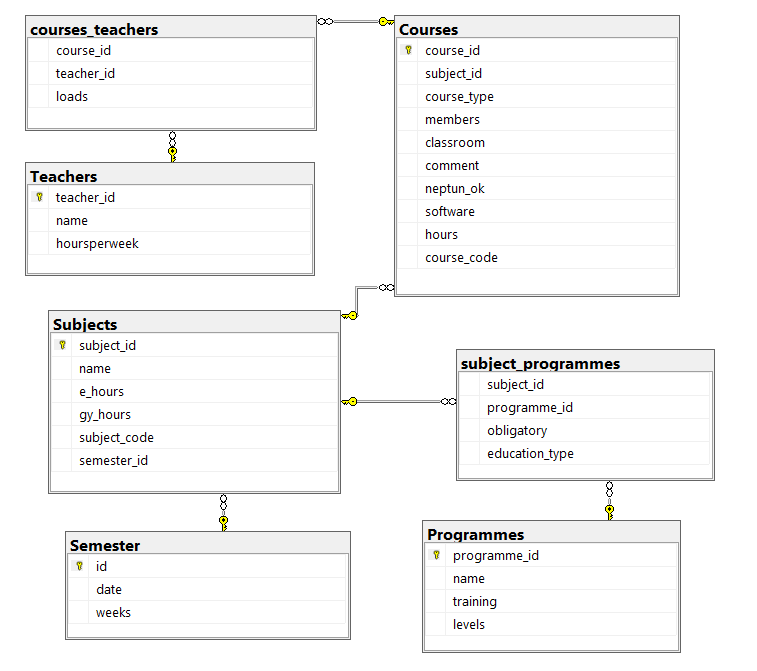
4.. ábra: Bejelentkezés oldal



5.. ábra: Kezdetleges tárgy kiírás

A webalkalmazás kinézete folyamatos fejlesztés alatt van.

A megtervezett adatbázis még a 6. ábra szerint néz ki. Ez későbbiekben, a fejlesztés során még változtatatásra kerülhet.



6. ábra: Adatbázis első verziójának a felépítése

## 2.3 Második beszámoló, Szeptember 16 – 29.

Az oldalunkhoz hozzáadtunk egy oldalsó navbar-t, ahol a keresési mezőt és a szemeszter (félévek) választási lehetőséget helyeztük el. Új szemesztert is hozzá lehet adni a rendszerhez, ami magától, a következő szemesztert hozza létre. Ez a navigációs sáv nem fix helyzetbe valósítottuk meg, akkor jelenik meg, ha a felhasználó a weboldal olralára húzza a kurzort. Azt a funkciót, hogy az oldalon való navigálás után a kezdőoldalra visszatérve ugyanarra a szemeszterre jussunk vissza a felhasználó, mint amelyiken legutóbb dolgozott session-al oldottuk meg.

Megvalósítottuk a jogosultságok kezelését, amit az admin az adminisztráció oldalon éri el. Ennek az oldalnak a vezérlését az AdministrationController kezeli. Két felhasználó szintet hoztunk létre, egy adminisztrátort és egy sima felhasználói szintet. Az adminisztrátornak minden jogosultsága megvan, beleértve az adatok teljes körű szerkesztését, törlését, hozzáadását, valamint a userek felhasználó szintjének a módosítását. A felhasználó csak megtekinteni tudja az adatokat, valamint kommentet fűzni hozzá.

A tárgyakat és kurzusokat jelenleg táblázatos módban jelenítjük meg, ez a későbbiekben még módosulni fog. A táblázatot és az oldal többi elemét is reszponzívan valósítjuk meg.

## 2.4 Harmadik beszámoló, Szeptember 30 – Október 6.

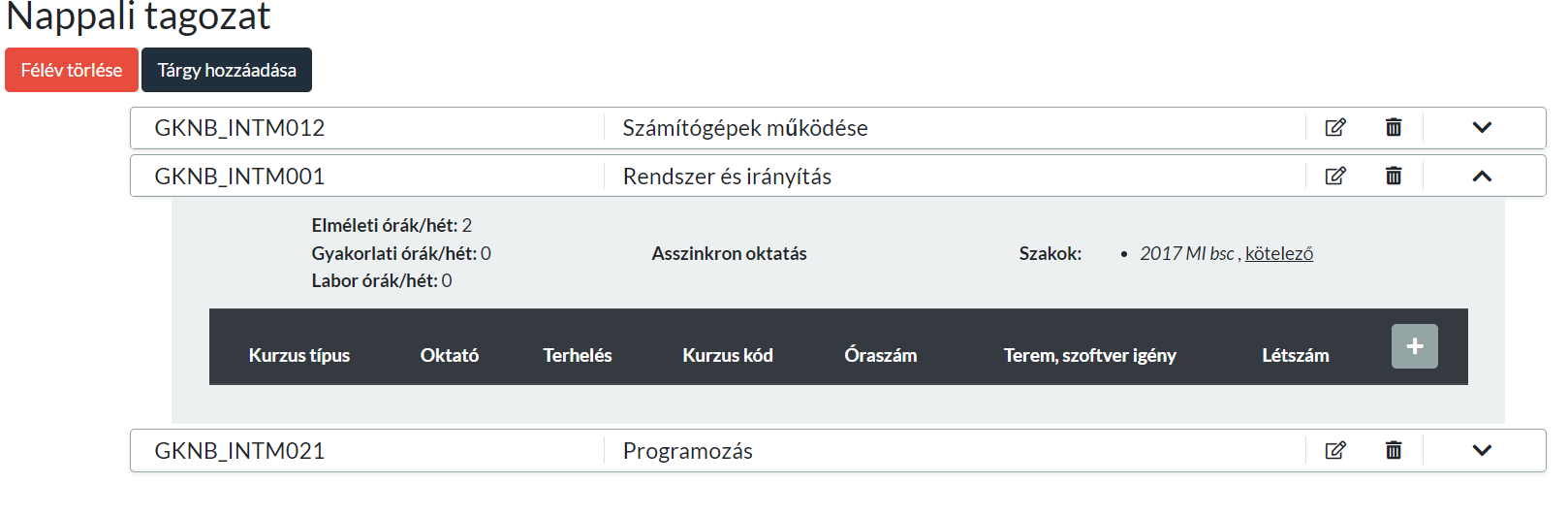
Mivel szerettünk volna egy modern és reszponzív kinézetet, viszont igény miatt egy táblázatos megjelenítést is meg szerettük volna hagyni, ezért kétféle adat megjelenítést eszközöltünk.

Az teljesen táblázatos kinézetet a 7. ábra szerint valósítottuk meg.



7. ábra: Adatok megjelenítése táblázatos formában

A reszponzív, átláthatóbb felületet collapse html elemekkel és bootstrap grid systemmel oldottuk meg. A kurzusok itt már nem a tantárgyak mellett, hanem azok alatt a tantárgyak legördülő részleteiben. Az így kapott kinézetet a 8. ábrán látható.



8. ábra: Adatok megjelenítése áttekinthetőbb formában

Az oldal közötti váltás megtartását újratöltésnél, valamint azt, hogy a webalkalmazásban történő navigáció után is a „kibontott” részletek szintúgy megmaradjanak a böngésző local storage-vel oldottuk meg.

Elkészítettük a tantárgyak és a kurzusok létrehozását és törlését. A létrehozást Ajax kérésekkel valósítottuk meg, egy felugró „modal” ablakon tudja a felhasználó felvinni az adatokat, és létrehozni a tantárgyat /kurzust. Ha a felhasználó törölni szeretne szintúgy egy felugró ablakon meg kell erősítenie azt.

## 2.5 Negyedik beszámoló, Október 7 – Október 13.

Mind a tantárgyakhoz, mind a kurzusokhoz hozzáadtuk a szerkesztési lehetőséget, ezzel együtt véglegesítettük a törlési és a létrehozási műveleteket is, így most már, ha a felhasználó hibás adatot szeretne bevinni már figyelmeztetést fog visszakapni.

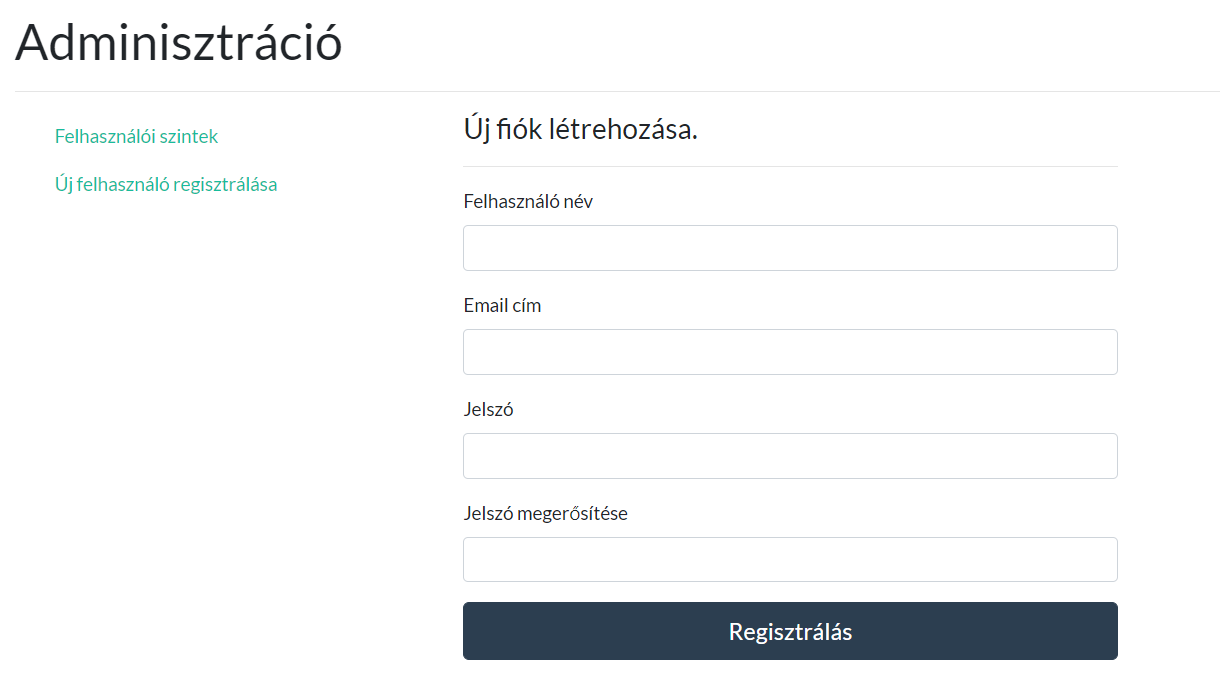
Kommentezési lehetőséget adtunk hozzá a 9. ábra szerint, így a rendszerbe felregisztrált oktatók megjegyzéseket tudnak fűzni ha valamilyen változtatást szeretnének kérni. Ezt a megjegyzés ikonra kattintva tudja megtenni a felhasználó. Alatta mindig kijeleztetjük, hogy ki szerkesztette utoljára, valamint ha az adott kommentre ráviszi a felhasználó az egerét, akkor megjelenik, hogy ki írta azt az adott megjegyzést. Az írt kommentet az adatbázisban letároljuk a kurzusok táblában. Kommentet szerkeszteni már nem lehet, törlési lehetőséget csak az admin jogosultságú felhasználónál állítottuk be.



9. ábra: Kommentek hozzáfűzése

## 2.6 Ötödik beszámoló, Október 14 - Október 20.

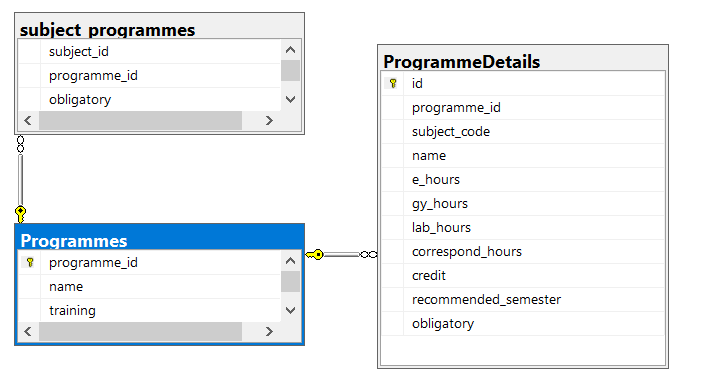
Tovább fejlesztettük az Adminisztráció oldal funkcióit. Átláthatóbb kinézetre váltottunk, valamint hozzáadtuk az „Új felhasználók regisztrálása” funkciót. Ezzel az adminisztrátor tudja regisztráltatni a felhasználót a rendszerbe. Felhasználó nevet, e-mail címet, és jelszót kell beállítania az új felhasználónak. (A jelszót természetesen később az újonnan regisztrált felhasználó meg tudja változtatni.) Az adminisztráció az MVC architektúrában egy önálló kontrollerkénk jelenik meg, független a rendszer gerincét alkotó HomeController-től.



10. ábra: Adminisztrációs felület

## 2.7 Hatodik beszámoló, Október 21 - November 3.

Hozzáadtuk a „Mintatantervek” felületet az alkalmazásunkhoz. Ebben a funkcióban lehet az egyes tantervekhez (Programme tábla) tantárgyakat adni. Erre azért volt szükség, hogy a szemeszterre ne is lehessen olyan tantárgyat kiírni, amely nem szerepel az adott mintatantervben. Mivel ezt a funkciót utólag tettük bele, így változtatnuk kellett az adatbázis szerkezetén is. Az adatbázisban hozzáadtunk egy „ProgrammeDetails” táblát a mintatanterv tárgyaihoz tartozó adatokkal. Ezzel egyidejűleg kivettük a „Programmes” táblából a *levels,*  ezt összevontuk a  *name*  mezővel. Valamint áthelyeztük az *education\_type* mezőt a „Subjects” táblába.



11. ábra: Változtatás az adatbázis felépítésében

Ezek után frissítettük az alkalmazás modelljeit az EF használatával, és létrehoztunk egy új kontrollert SyllabusController néven. Az adatok megjelenítést, a kurzuskiíráshoz hasonlóan csináltuk, viszont itt már csak a táblázatos kinézetet alkalmaztuk. A navigációs sávot a *Kurzusok* oldalhoz hasonló képpen készítettük el.

## 2.8 Hetedik beszámoló. November 4 – November 17.

Tovább fejlesztettük a mintatantervek funkciót. A felületen elhelyeztünk egy beviteli mezőt, ami az it.sze.hu-ról várja a mintatanterv url linkjét, elküldés után kinyerjük az információkat a Széchenyi István egyetem oldalán található mintatanterv táblázatból és letároljuk őket az adatbázisban. Így nem muszáj manuálisan az adatokat felvennie a felhasználónak.

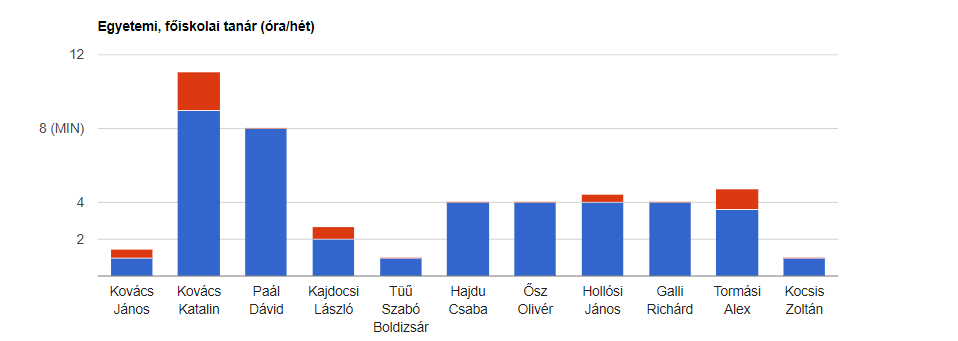
Az adminisztrációs felülethez hozzáadtunk a felhasználó törlése funkciót, így az admin jogosultságú felhasználó már el is tudja távolítani a sima usereket az alkalmazásból. Az adatok törlésével véglegesen eltávolítja a fiókokat, és azokat nem lehet visszaállítani.

Eszközöltünk még számos apróbb változtatást, hibajavítást a rendszerben, például megváltoztattuk a bejelentkezési felületet, a kurzusok felületen már egy gombnyomással ki lehet bontani és össze lehet csukni a legördülő tartalmakat.

## 2.9 Nyolcadik beszámoló. November 18 – November 30.

A mintatantervekhez létrehoztuk az összes alapműveletet, így most már manuálisan is lehet törölni, szerkeszteni és hozzáadni tantárgyakat egy mintatantervhez.

Ugyan úgy a Statisztikánál, vagyis a TeachersControllerben is elkészítettük, a tanárok hozzáadását. Az egyetemi beosztások miatt a *Positions* táblával kibővítettük az adatbázist. Ha a tanár inaktív státuszban van, vagyis éppen nem dolgozik az egyetemen, akkor nem szükséges kitörölni. Ezt külön be lehet állítani. Csak azokat a tanárokat lehet hozzárendelni egy újonnan elkészített kurzushoz, akik aktív státuszban vannak. A tanárok neve mellett megjelenítettük, hogy az utolsó két félévben összeessen mennyi órájuk volt. Ezen felül statisztikákat hoztunk létre. Minden egyetemi beosztásnál külön oszlopdiagramon jelenítettük meg, hogy az egyes tanárok hetente a legutolsó félévben mennyi órát vállalnak a kötelező heti óraszámhoz képest. Ezt a 12. ábra szemlélteti.



12. ábra: Tanárok heti óraszám statisztikája

További megvalósított elemek:

* Email küldése elfelejtett jelszónál.
* Figyelmeztetési funkció, ha a tantárgyhoz még nincsen elegendő kurzus kiírva.
* Kurzust csak úgy enged felvenni, ha a hozzárendelt tanárok összes terhelése kiadja a 100%-ot.
* Pozitív visszajelzés küldése, ha a tantárgyhoz az összes kurzus már be van írva a neptunba és a tárgyakkal nincsen egyéb probléma.
* További kinézeti változtatások.

# Összegzés

Úgy érezzük, hogy a kiírt feladatot sikerült teljesítenünk, és egy komplett, működő rendszert létrehoznunk. Egyedül a verziókövetést nem valósítottuk meg, hiszen a projekt fejlesztése során rájöttünk, hogy mivel csak egy tényleges szerkesztője lesz a rendszernek, erre nem lesz szükség. Ezt konzultáció alkalmával megbeszéltük a projektvezető oktatóval. Az elkészített rendszer fejlesztése során sok tapasztalatot sikerült szereznünk, hiszen arra vállalkoztunk, hogy egy számunkra eddig még ismeretlen technológiával dolgozzunk. Habár sokszor ütköztünk nagyobb problémába, ezeket mélyebb utánajárás után mindig sikerült megoldanunk. Reméljük a megszerzett tudást a jövőben kamatoztatni tudjuk.