



EÖTVÖS LORÁND TUDOMÁNYEGYETEM

Informatikai Kar

Média- és Oktatásinformatikai Tanszék

ELTE IK órarendtervező webalkalmazás

Dr. Abonyi-Tóth Andor
adjunktus

Laczik Tamás
programtervező informatikus szak

Budapest, 2019

TARTALOMJEGYZÉK

Bevezetés	3
Motiváció	3
Célok.....	3
Felhasználói dokumentáció.....	5
A megoldott probléma	5
A felhasznált módszerek	5
Rendszerkövetelmények	5
A program használata	6
Telepítés.....	6
A program használata	9
Fejlesztői dokumentáció	15
A probléma specifikációja	15
Funkcionális követelmények	15
Képernyőtervek.....	19
A felhasznált módszerek	21
A felhasznált technológiákról általánosságban.....	21
Backend	21
Frontend.....	22
A program logikai és fizikai szerkezete.....	22
A TTK TO adatbázis elérése	23
Könyvtárszerkezet	24
Adatbázisdiagram	25
Backend végpontok.....	26
Osztálydiagram	30
Tesztesetek.....	32

Egységtesztek.....	32
Manuális tesztek	35
Összegzés.....	40
Továbbfejlesztési lehetőségek	40
Irodalomjegyzék	42

BEVEZETÉS

Motiváció

Szakedolgozatom témájának egy vizuális órarendkészítő webalkalmazás fejlesztését választottam. Saját tapasztalatból tudom, hogy a hallgatóktól sok időt és energiát vesz el félév kezdete előtt az ideális órarend megalkotása. Mindenkinek más és más preferenciái vannak oktatók és időbeosztás tekintetében, melyek alapján végül el kell jutni egy megfelelő kompromisszumhoz. A tanrend böngészésére használt oldal ugyan minden szükséges információt megad ehhez, azonban az általa használt táblázatos formátumot nehéz átlátni, így a hallgatóknak maguknak kell megrajzolniuk órarendjüket. Erre a problémára szeretnék egy egyszerűbb megoldást nyújtani.

Az általam fejlesztett webalkalmazás közvetlenül a TTK TO online adatbázisából kéri le a tanórák adatait, így mindig naprakész információkkal szolgál a felhasználóknak. A projektnek menet közben az idő görög istene után a „Chronos” nevet adtam. Ezt a fantázianevet a szakdolgozat témájának bejelentése után találtam ki, ezért nem szerepel annak címében, a dolgozat további részében, illetve az alkalmazásban azonban rendszeresen használom.

Célok

Egy felhasználóbarát, könnyen kezelhető, és hasznos alkalmazást szeretnék kiadni a kezeim közül. Az alkalmazás webes mivolta is a széleskörű elterjedést és az egyszerű használatbavételt hivatott elősegíteni, hisz így a végfelhasználónak nincs más dolga, mint begépelni az oldal címét, és máris megkezdheti az alkalmazás használatát. Míg mobil környezetben többnyire feladat-specifikus alkalmazások telepítése a kényelmes, az asztali- és hordozható számítógépek világában inkább az azonnal használható, telepítést és konfigurálást nem igénylő programok használata áll kézre.

Személyes céloom továbbá a legkorszerűbb webes technológiák használatának gyakorlása is, melyek már fiatal koromtól kezdve érdekeltek. Akkoriban az fogott meg bennük, hogy segítségükkel viszonylag gyorsan és egyszerűen készíthetünk kézzel fogható, használható és esztétikus honlapokat, illetve internetes alkalmazásokat. Az évek során egyre többet

tanultam róluk az iskolában, valamint önszorgalomból. Jelenlegi és előző munkahelyemen is hasonló projekteken dolgoztam, így elmondhatom magamról, hogy az informatika számos területe közül ezen vagyok a legjáratosabb. Egy ennyire gyorsan fejlődő környezetben azonban az ember folyamatosan új módszerekkel és irányzatokkal találja szemben magát, így a tanulás sosem ér véget.

FELHASZNÁLÓI DOKUMENTÁCIÓ

A megoldott probléma

A program az ELTE Lágymányosi kampuszán tanuló hallgatóknak hivatott segítséget nyújtani órarendjük összeállításában. A TTK TO adatbázisának segítségével órarendeket építhetünk fel, melyek vizuálisan jelennek meg, ezzel segítve az előttünk álló félév megtervezését. Ha szeretnénk, az órarendeket megoszthatjuk ismerőseinkkel is.

A felhasznált módszerek

A szoftver megálmodásakor fontos szempont volt a minél több felhasználó kiszolgálása, ezért a hordozhatóságot szem előtt tartva webalkalmazás fejlesztése mellett döntöttem. Köszönhetően a modern böngészők szabványkövető magatartásának, a jól megírt kód manapság gyakorlatilag bármely környezetben ugyanolyan kiszámíthatóan és megbízhatóan fut.

Mindezen pozitívumok ellenére sajnos előfordulhatnak böngészőspecifikus problémák, így alkalmazásomat érdemes a széleskörű elterjedtségnek örvendő Google Chrome program legfrissebb verziójával használni.

Rendszerkövetelmények

A program futtatásához szükségünk van egy webszerverre. Ez lehet egy távoli gép, vagy ha nincs ilyenhez hozzáférésünk, akkor akár a saját számítógépünk is. A webszervernek az alábbi technológiákat kell támogatnia:

- PHP
- MySQL

A program futtatható a XAMPP nevű, platformfüggetlen webszerver-szoftvercsomag segítségével, mely többek között tartalmazza a széleskörűen elterjedt Apache webszervert, illetve támogatja a fenti technológiák használatát is.

A program használata

Telepítés

A XAMPP letöltése és beüzemelése

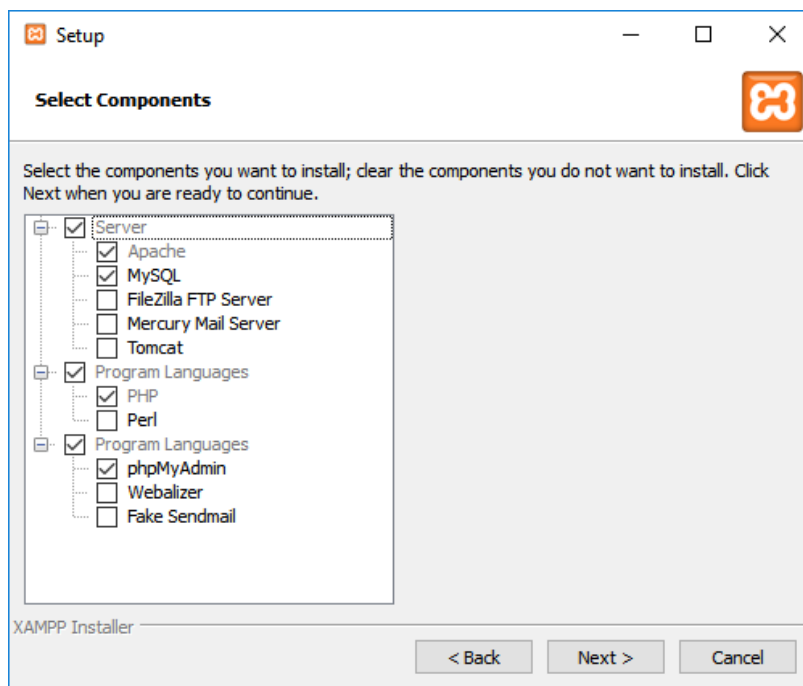
A XAMPP webserverver-szoftvercsomag ingyenesen letölthető a <https://www.apachefriends.org/hu/index.html> címen elérhető weboldalról, az általunk használt operációs rendszer kiválasztásával. Ebben az útmutatóban a Windows-rendszerhez készült, 7.2.11-es verzió telepítését ismertetem.



1. ábra: A XAMPP webserverver-szoftvercsomag honlapja

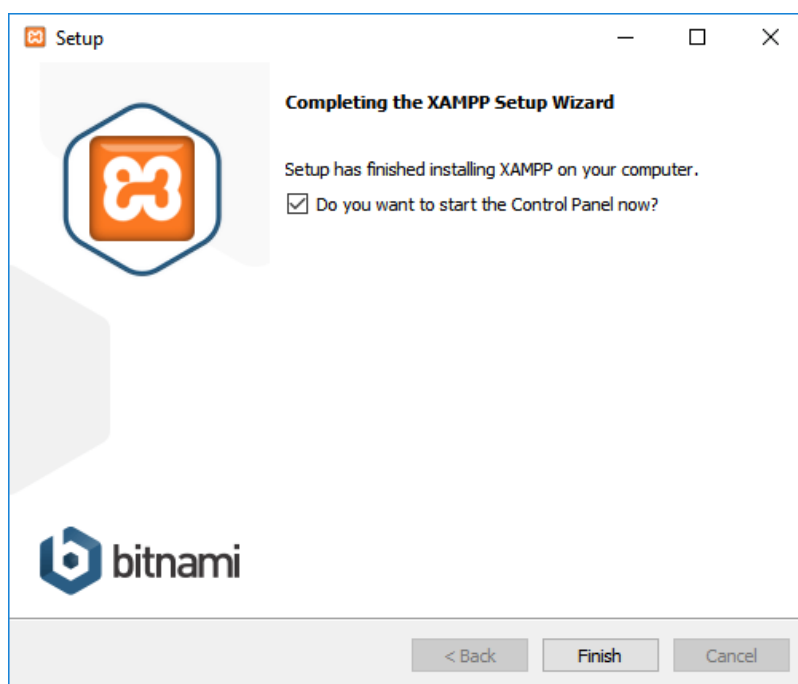
A letöltött telepítőfájlt futtatva haladjunk végig értelemszerűen a varázsló lépésein. A Chronos órarendtervező futtatásához elegendő az Apache, MySQL, valamint PHP modulokat telepíteni. Ajánlott azonban a phpMyAdmin modul installálása is, ugyanis ezen eszközzel később könnyen kezelhetjük adatbázisunkat.

A varázsló alapértelmezetten a `C:\xampp` mappába telepíti a programot, amit szabadon megváltoztathatunk. Azonban arra a telepítő is felhívja a figyelmünket, hogy más könyvtár használata esetén felmerülhetnek fájlhozzáférési problémák.



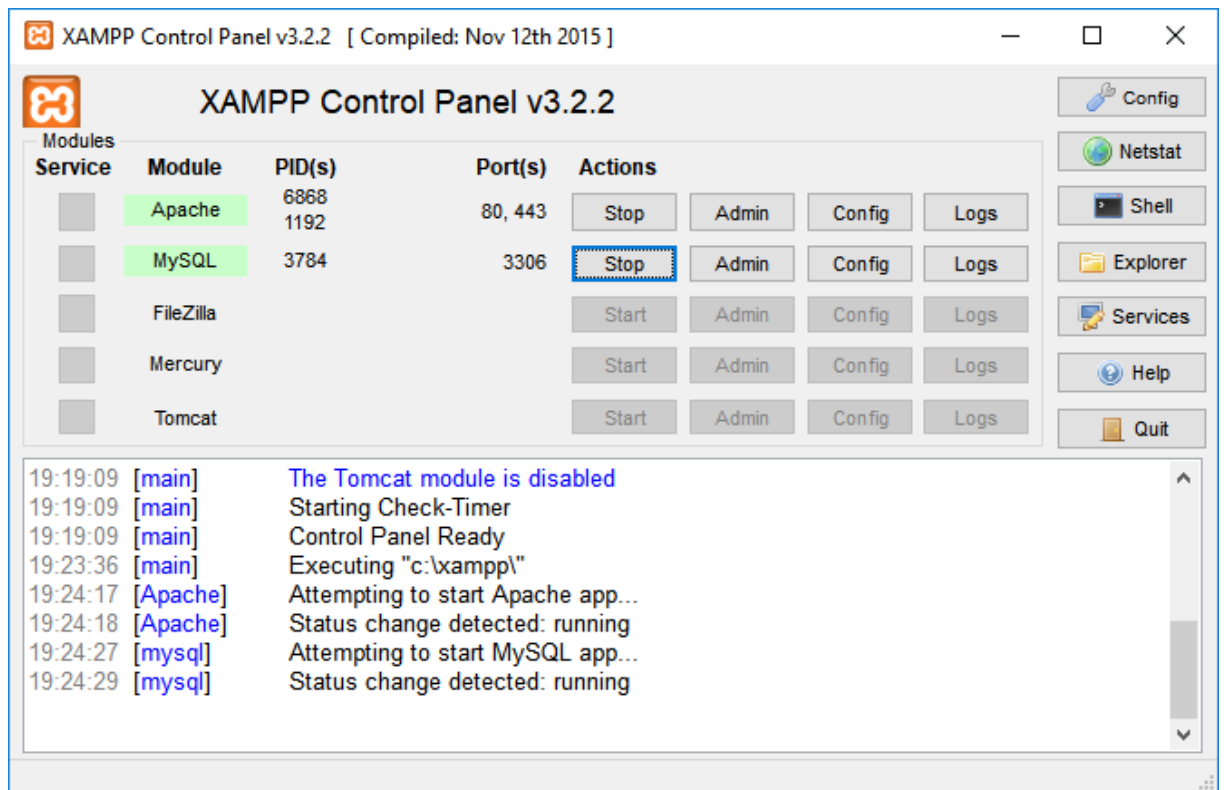
2. ábra: Ajánlott modulok a Chronos órarendtervező alkalmazáshoz

A telepítés végén lehetőségünk van egyből elindítani a XAMPP vezérlőpultját, melyet érdemes megtennünk a folytatás érdekében. A telepítést követően kapcsoljuk be az Apache és a MySQL modulokat a „Start” gombok segítségével. Ha minden jól ment, a modulok neve rövidesen zöld kiemelést kap, és megjelennek az általuk használt portok számai is.



3. ábra: A XAMPP telepítésének befejezése

Ekkor egy tetszőleges böngészővel megnyitva a `http://localhost` címet, a számítógépünkön futó Apache webszerver üdvözlőoldalát tekinthetjük meg. A `http://localhost/phpmyadmin` címen pedig a phpMyAdmin felülete fogad bennünket, ahol adatbázisunkat tudjuk kezelni.



4. ábra: A XAMPP vezérlőpult az Apache és a MySQL modulok sikeres elindítását követően

A Chronos órarendtervező alkalmazás telepítése

XAMPP szoftvercsomag használata esetén a Chronos órarendtervező telepítéséhez a program összes állományát a telepítési könyvtárban lévő `htdocs` mappába kell másolnunk a könyvtárszerkezet megtartásával. Dönthetünk úgy, hogy először kiürítjük a `htdocs` mappát, majd ide helyezzük el a fájlokat; teljes törlés helyett azonban létrehozhatunk egy új mappát is, melybe telepíthetjük a Chronos órarendtervezőt. Én az előbbi megoldást választom, hogy a gyökérkönyvtárból elérhető legyen a program.

A fájlok átmásolása után meg kell nyitnunk a `database.php` fájlt, ahol be kell állítanunk az adatbázisunk eléréséhez használt paramétereket. Erre a Database osztály `host`, `username`, `password`, `database` változóinak megfelelő értékekre állításával van lehetőségünk. Biztonsági okokból célszerű létrehozunk egy új felhasználót a

phpMyAdmin segítségével a Chronos számára, aki kizárólag SELECT, INSERT, UPDATE, DELETE jogokkal rendelkezik az órarendtervező által használt adatbázison, majd ezen felhasználó hozzáférési adatait megadni a Database osztályban.

```
class Database {
    private $conn;

    private $host = 'localhost';
    private $username = 'Chronos';
    private $password = 'h8qqAL4cWG2vzOwo';
    private $database = 'chronos';

    function __construct() {
```

5. ábra: Beállítások egy frissen létrehozott felhasználói fiókhoz

A program használatba vétele előtti utolsó teendőnk az adatbázisstruktúra kialakítása. Ehhez keressük fel a <http://localhost/phpmyadmin> oldalt, ahol lépünk be megfelelő jogosultságokkal. A XAMPP alapértelmezetten jelszó nélkül hozza létre a root felhasználót, így bejelentkezés nélkül is teljes irányításunk van az adatbázisunk felett. Ezen később érdemes változtatni biztonsági szempontok miatt.

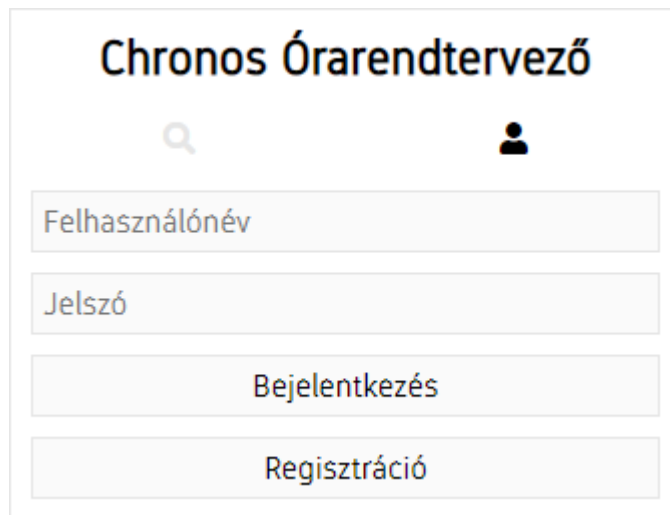
Az importálás menüpontra kattintva lehetőségünk van megnyitni a `data.sql` fájlt, melynek futtatásával létrehozuk a `chronos` adatbázist a megfelelő táblákkal. Ha mindezzel megvagyunk, készen állunk a program használatba vételére.

Előfordulhat, hogy távoli kiszolgáló használata esetén nincs jogosultságunk új adatbázist létrehozni. Ekkor ki kell szednünk az adatbázist létrehozó sort a `data.sql` fájlból, valamint a táblákat a meglévő adatbázisban kell létrehoznunk.

A program használata

A felhasználói felület

A program alapvető funkciói bejelentkezés nélkül is használhatók. Az alkalmazás megnyitásakor egy üres órarend fogad minket, a képernyő bal oldalát pedig egy multifunkciós oldalsáv foglalja el. Az oldalsávon két nézet között választhatunk: alapértelmezetten a felhasználó-piktogrammal jelölt, bejelentkezéshez illetve regisztrációhoz használt űrlapot látjuk, azonban a nagyító-ikonra kattintva átválthatunk a tanóra-keresőre.



Chronos Órarendtervező

🔍 👤

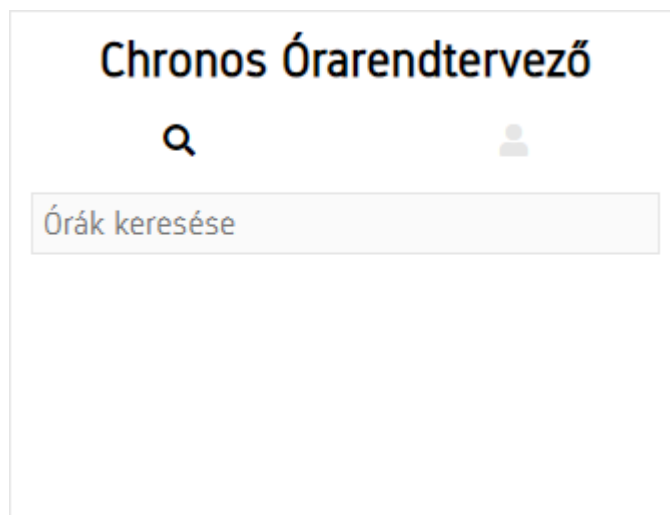
Felhasználónév

Jelszó

Bejelentkezés

Regisztráció

6. ábra: Az oldalsáv felhasználói nézete bejelentkezés előtt



Chronos Órarendtervező

🔍 👤

Órák keresése

7. ábra: Az oldalsáv keresési füle

A tanóra-kereső

Az utóbbi nézetben található keresőmezőbe tetszőleges kifejezést írva a rendszer automatikusan lekéri az adott kulcsszavakra illeszkedő tanórákat a TTK TO adatbázisából. Az ekkor megjelenő találati listában tantárgyak szerint csoportosítva láthatjuk az egyes tanórák időpontjait. A tanórák lehetnek előadások, gyakorlatok, illetve konzultációk is.



8. ábra: A keresési találatok listája

Tetszőleges tanóra kattintva hozzáadhatjuk azt az órarendünkhöz. Amennyiben nem vagyunk bejelentkezve, a rendszer megjelenít egy figyelmeztetést, mely szerint az általunk készített órarend az oldal elhagyásával el fog veszni, hacsak nem mentjük el regisztrációval vagy bejelentkezéssel. Erre akkor is felhívja a figyelmünket egy felugró ablak, ha be szeretnénk zárni az oldalt, esetleg frissíteni, vagy másik oldalra menni.

⚠ Az órarend elmentéséhez regisztrálj, vagy jelentkezz be!

9. ábra: Az órarend felett megjelenő figyelmeztető üzenet

Az órarendhez csak olyan tanórákat adhatunk hozzá, melyeknek ismert a kezdő- és befejező időpontja, illetve az, hogy a hét melyik napján vannak. Ezen adatok jellemzően levelező tagozatos tanórák esetén hiányoznak, érthető okból. Ha egy olyan tanórára kattintunk, mely már szerepel az aktuális órarendben, ezzel eltávolítjuk onnan.

Analízis 1. GY. (BSc,08,A) IP-08aAN1G



1. gyakorlat
Hétfő 12:00-14:00
Chripkó Ágnes
Déli Tömb 0-220 Kárteszi Ferenc terem

2. gyakorlat
Szarvas Kristóf

3. gyakorlat
Bognár Gergő

10. ábra: Egy, már az órarendhez hozzáadott, illetve hiányos információval rendelkező tanórák




Bejelentkezés és regisztráció

Az oldalsáv másik nézetén található űrlapot használhatjuk bejelentkezéshez és regisztrációhoz egyaránt, hisz mindkettőhöz csupán egy felhasználónévre és jelszóra van szükség. Ha sikeresen belépünk felhasználói fiókunkba vagy regisztrálunk és nem mentett változások vannak az órarendben, akkor a rendszer lehetőséget ad annak elmentésére tetszőleges néven.

Felhasználói fiók kezelése

Bejelentkezés után az előbb említett űrlap helyét egy felhasználói panel veszi át. Itt láthatjuk, hogy milyen néven léptünk be. Lehetőségünk van kijelentkezésre, jelszavunk megváltoztatására, illetve felhasználói fiókunk törlésére. Utóbbi esetben minden általunk készített órarend is törlődik, melyek visszaállítására nincs lehetőség. Emiatt a rendszer először figyelmeztet minket, és megerősítést kér szándékunkról.

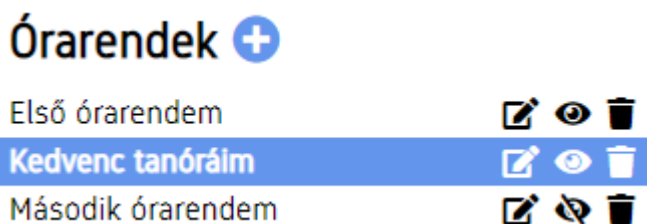
tomlaczik

-  KIJELENTKEZÉS
-  JELSZÓ MÓDOSÍTÁSA
-  FELHASZNÁLÓ TÖRLÉSE

11. ábra: A felhasználói fiók kezelésére szolgáló menü

Saját órarendek

Ezen rész alatt megtekinthetjük saját órarendjeink listáját. Az órarend nevére kattintással meg is nyithatjuk azt. Lehetőségünk van továbbá az órarend átnevezésére, törlésre, illetve láthatóságának megváltoztatására. Amennyiben úgy döntünk, hogy egyik órarendünket láthatóvá tesszük, ez esetben a rá mutató hivatkozás segítségével bárki megtekintheti azt, akár bejelentkezés nélkül is. Ez remek lehetőség, ha barátainkkal közösen szeretnénk tervezni, hisz így ők is láthatják, hogy éppen hogyan alakul a féléves órarendünk. A láthatóságot természetesen bármikor visszaállíthatjuk rejtettre.





12. ábra: Saját órarendek listája, miután a középsőt megnyitottuk


Tanóra részletei


Ha az órarendünkre tekintve egy tanóráról bővebb információkkal szeretnénk gazdagodni, ezt megtehetjük egyetlen kattintással a szóban forgó tanórán. Ennek hatására megjelenik egy felugró ablak, melyben a tanóra összes jellemzőjét láthatjuk. Lehetőségünk van továbbá az ablak bezárására, a tanóra törlésére, illetve alternatív időpontok megjelenítésére. Ha utóbbit választjuk, az ablak bezáródik, és az oldalsávon megjelenik a kereső, mely az adott tanóra nevével megegyező találatokat mutat.


Analízis 1. GY. (BSc,08,A)


 IP-08aAN1G

 1. gyakorlat

 Hétfő

 12:00-14:00

 Chripkó Ágnes

 Déli Tömb 0-220 Kárteszi Ferenc terem

 TÖRLÉS  ALTERNATÍV IDŐPONTOK  VISSZA

13. ábra: A tanóra adatait részletező ablak

FEJLESZTŐI DOKUMENTÁCIÓ

A probléma specifikációja

Funkcionális követelmények

A program funkcionális követelményeit a felhasználói történetek alábbi táblázata tartalmazza.

Funkció	Regisztráció üres órarenddel
Leírás	A felhasználó úgy regisztrál magának fiókot, hogy az aktuálisan megnyitott órarendje üres, vagy valaki másét tekinti meg
Előfeltétel	<ul style="list-style-type: none">• A felhasználó nincs bejelentkezve• A megnyitott órarend üres, vagy nem a felhasználóé• A felhasználó beírta a kívánt felhasználónevet és jelszót• A megadott felhasználónév még nincs használatban
Kiváltja	A felhasználó a regisztráció gombra kattint
Eredmény	<ul style="list-style-type: none">• Az adatbázisban létrejön a felhasználóhoz tartozó rekord• A felhasználót bejelentkezteti a rendszer• A felhasználó nevében készít egy üres órarendet a rendszer, majd megjeleníti azt

Funkció	Bejelentkezés üres órarenddel
Leírás	A felhasználó úgy jelentkezik be, hogy az aktuálisan megnyitott órarendje üres, vagy valaki másét tekinti meg
Előfeltétel	<ul style="list-style-type: none"> • A felhasználó nincs bejelentkezve • A megnyitott órarend üres, vagy nem a felhasználóé • A felhasználó beírta a felhasználónevét és jelszavát • A felhasználónév és jelszó páros helyes
Kiváltja	A felhasználó a bejelentkezés gombra kattint
Eredmény	<ul style="list-style-type: none"> • A felhasználót bejelentkezteti a rendszer • A rendszer megjeleníti a felhasználó legutóbb megnyitott órarendjét, vagy készít neki egy újat, amennyiben nincs ilyen

Funkció	Kijelentkezés
Leírás	A felhasználó kijelentkezik
Előfeltétel	<ul style="list-style-type: none"> • A felhasználó be van jelentkezve
Kiváltja	A felhasználó a kijelentkezés gombra kattint
Eredmény	<ul style="list-style-type: none"> • A felhasználót kijelentkezteti a rendszer • Egy üres órarend jelenik meg

Funkció	Tanóra keresése
Leírás	A felhasználó tanórákat keres az adatbázisban, melyeket hozzáadhat az órarendjéhez
Előfeltétel	<ul style="list-style-type: none"> • A felhasználó nem egy másik ember órarendjét nézi
Kiváltja	A felhasználó beír egy keresőkifejezést a keresőmezőbe
Eredmény	<ul style="list-style-type: none"> • A keresőkifejezésre illeszkedő nevű tanórák megjelennek az oldalsávon

Funkció	Tanóra hozzáadása
Leírás	A felhasználó hozzáadja a kiválasztott tanórát az órarendjéhez
Előfeltétel	<ul style="list-style-type: none"> • A felhasználó nem egy másik ember órarendjét nézi • A felhasználó beírt egy keresőkifejezést, melyre kapott illeszkedő tanórákat • A kiválasztott tanóra nem szerepel a megnyitott órarendben • A kiválasztott tanórához elérhető nap, valamint kezdő- és befejező időpont
Kiváltja	A felhasználó a tanórára kattint
Eredmény	<ul style="list-style-type: none"> • A kiválasztott tanóra megjelenik az aktuális órarendben • A kiválasztott tanóra bekerül az adatbázisba az aktuális órarendhez kapcsolva

Funkció	Óra törlése
Leírás	Egy tanóra törlése az aktuális órarendből
Előfeltétel	<ul style="list-style-type: none"> • A felhasználó nem egy másik ember órarendjét nézi • A tanóra szerepel az aktuális órarendben
Kiváltja	A felhasználó a tanóra törlése gombra kattint, vagy a keresési találatok között egy olyan tanórát választ ki, mely szerepel a megnyitott órarendben
Eredmény	<ul style="list-style-type: none"> • A tanóra eltűnik az aktuális órarendből • A tanóra törlődik az adatbázisból az aktuális órarend alól

Funkció	Órarend létrehozása
Leírás	A felhasználó új órarendet hoz létre
Előfeltétel	<ul style="list-style-type: none"> • A felhasználó be van jelentkezve
Kiváltja	A felhasználó az új órarend létrehozása gombra kattint
Eredmény	<ul style="list-style-type: none"> • A megadott névvel egy üres órarend kerül elmentésre az adatbázisba a felhasználóhoz • Az újonnan készített, üres órarend megjelenik

Funkció	Órarend láthatóságának megváltoztatása
Leírás	A felhasználó meg szeretné változtatni az órarendjének mások által való láthatóságát
Előfeltétel	<ul style="list-style-type: none"> • A felhasználó be van jelentkezve • A felhasználónak van legalább egy órarendje
Kiváltja	A felhasználó a láthatóság megváltoztatása ikonra kattint
Eredmény	<ul style="list-style-type: none"> • Az adott órarend láthatósági állapota az ellenkezőjére változik

Funkció	Órarend átnevezése
Leírás	A felhasználó meg szeretné változtatni egy órarendjének nevét
Előfeltétel	<ul style="list-style-type: none"> • A felhasználó be van jelentkezve • A felhasználónak van legalább egy órarendje
Kiváltja	A felhasználó az átnevezés gombra kattint
Eredmény	<ul style="list-style-type: none"> • A kiválasztott órarend neve a megadottra módosul

Funkció	Órarend törlése
Leírás	A felhasználó törölni szeretné egy órarendjét
Előfeltétel	<ul style="list-style-type: none"> • A felhasználó be van jelentkezve • A felhasználónak van legalább egy órarendje
Kiváltja	A felhasználó az órarend törlése gombra kattint
Eredmény	<ul style="list-style-type: none"> • A kiválasztott órarend törlődik az adatbázisból • A rendszer betölti a legutóbb létrehozott órarendet

Funkció	Órarend megnyitása
Leírás	A felhasználó megnyitja egyik órarendjét
Előfeltétel	<ul style="list-style-type: none"> • A felhasználó be van jelentkezve • A felhasználónak van legalább egy órarendje • A kiválasztott órarend nincs megnyitva
Kiváltja	A felhasználó az egyik órarendjének nevére kattint a listában
Eredmény	<ul style="list-style-type: none"> • Az oldal újratöltődik a kiválasztott órarenddel

Képernyőtervek

A program felhasználói felületének vázlatos tervei alább láthatóak.

Órarend neve					
	Hétfő	Kedd	Szerda	Csütörtök	Péntek
8:00					
9:00					
10:00	Tanóra 1				
11:00			Tanóra 2		
12:00					
13:00					
14:00			T.óra 3		

14. ábra: Az alkalmazás átfogó felületi terve

Keresés	[Profil]
Felhasználónév	
Jelszó	
Bejelentkezés	
Regisztráció	

15. ábra: Az oldalsáv bejelentkezésre szolgáló része

Keresés	[Profil]
Felhasználónév	
-Kijelentkezés	
-Jelszó módosítása	
-Fiók törlése	
Órarendek	
-Órarend 1	Átnev./Láth./Törl.
-Órarend 2	Átnev./Láth./Törl.

16. ábra: Az oldalsávon lévő felhasználói menü bejelentkezés után

[Keresés]	Profil
Analízis	
Analízis 1.	
- 1. előadás	
Időpont	
Tanár neve	
Helyszín	
- 2. előadás	
Időpont	
Tanár neve	
Helyszín	
- 1. gyakorlat	
Időpont	
Tanár neve	
Helyszín	
Analízis 2.	
- 1. előadás	
Időpont	
Tanár neve	
Helyszín	

17. ábra: Keresési eredmények megjelenítése az oldalsávon

Tárgy neve

- Tárgykód
- X. gyakorlat/előadás/konzultáció
- Nap
- Időpont (-tól-ig)
- Tanár neve
- Helyszín

Törlés

Alternatívák

Vissza

18. ábra: A tanórára kattintva megjelenő, részleteket tartalmazó ablak

A felhasznált módszerek

A felhasznált technológiákról általánosságban

A program szerveroldali részét PHP nyelven fejlesztettem, MySQL adatbázist használva. A kliensoldali kódot JavaScriptben írtam, az oldal struktúrájának leírásáról HTML, a megjelenítés definiálásáról pedig CSS gondoskodik. A fejlesztésnél nem használtam külső könyvtárakat, illetve keretrendszereket. Ezt a döntést azért hoztam meg, mert teljes mértékben én akartam irányítani a kód működését.

A projekt kliensoldali részét egyfajta kísérletnek is szántam, melynek keretein belül igyekeztem a JavaScript legújabb funkcióit kiaknázni. Emiatt régebbi böngészők használata esetén előfordulhatnak hibák, azonban a lehető legtisztább és legmodernebb kódra törekedtem, a visszafelé kompatibilitás nem volt cél. Úgy gondolom, hogy ezen céloom szakmailag is sokat segített, hiszen ennek köszönhetően rengeteg újat tanultam korszerű JavaScript-technológiákról, mely tudást később egész biztosan kamatoztatni tudom.

Backend

PHP

„A PHP egy általános szerveroldali szkriptnyelv dinamikus weblapok készítésére. Az első szkriptnyelvek egyike, amely külső fájl használata helyett HTML oldalba ágyazható. A kódot a webszerver PHP feldolgozómodulja értelmezi, ezzel dinamikus weboldalakat hozva létre.”^[1]

MySQL

„A MySQL egy többfelhasználós, többszálú, SQL-alapú relációs adatbázis-kezelő szerver. A MySQL az egyik legelterjedtebb adatbázis-kezelő, aminek egyik oka lehet, hogy a teljesen nyílt forráskódú LAMP (Linux–Apache–MySQL–PHP) összeállítás részeként költséghatékony és egyszerűen beállítható megoldást ad dinamikus webhelyek szolgáltatására.”^[2]

Frontend

JavaScript

„A JavaScript programozási nyelv egy objektumalapú, prototípus alapú szkriptnyelv, amelyet weboldalakon elterjedten használnak. Eredetileg Brendan Eich, a Netscape Communications mérnöke fejlesztette ki; neve először Mocha, majd LiveScript volt, később „JavaScript” nevet kapott, és szintaxisa közelebb került a Sun Microsystems Java programozási nyelvéhez. A JavaScriptet először 1997–99 között szabványosította az ECMA „ECMAScript” néven. A jelenleg is érvényes szabvány az ECMA-262 Edition 3 (1999. december), ami a JavaScript 1.5-nek felel meg. Ez a szabvány egyben ISO szabvány is.”^[3]

HTML

„A HTML (angolul: HyperText Markup Language=hiperszöveges jelölőnyelv) egy leíró nyelv, melyet weboldalak készítéséhez fejlesztettek ki, és mára már internetes szabvánnyá vált a W3C (World Wide Web Consortium) támogatásával. Az aktuális változata az 5, mely az SGML általános jelölőnyelv egy konkrét alkalmazása (azaz minden 5-ös HTML dokumentum egyben az SGML dokumentumszabványnak is meg kell, hogy feleljen). Ezt tervek szerint lassan kiszorította volna az XHTML (ami a szintén SGML alapú XML leíró nyelven alapul.”^[4]

CSS

„A CSS (Cascading Style Sheets, magyarul: egymásba ágyazott stíluslapok) a számítástechnikában egy stílusleíró nyelv, mely a HTML vagy XHTML típusú strukturált dokumentumok megjelenését írja le. Ezen kívül használható bármilyen XML alapú dokumentum stílusának leírására is, mint például az SVG, XUL stb.”^[5]

A program logikai és fizikai szerkezete

Az alkalmazás két élesen elkülönülő része a kliens-, illetve a szerveroldali kód. Az alkalmazás logikájának nagy része egy úgynevezett „vastag kliensben” kapott helyet, a szerveroldal csupán az oldal vázának generálásáért és az adatbáziskezelésért felel. Utóbbit egy REST-stílusú API-n keresztül lehet elérni a kliensoldalról.

Az adatbázissal a Database osztály tart közvetlen kapcsolatot, ezen osztály metódusaiban találhatóak az alkalmazás által használt SQL-kódok. Ez fontos karbantarthatósági szempontból, hiszen így a kód többi részéből az adatbázist beszédes nevű metódusok segítségével kezelhetjük, deklaratív módon. Amennyiben a program mögötti adatbázis felépítése, vagy típusa bármilyen formában megváltozna, kizárólag a Database osztály belsejét kell átalakítani. A REST API végpontjai egy másik, `rest.php` nevű fájlban találhatóak, procedurális módon leírva. A HTML kód az `index.php` fájlban található, egyszerű, a megjelenítést irányító logikával együtt.

A kliensoldali JavaScriptet osztályokra tagoltam, melyek az oldal különböző részeinek irányításáért felelősek. Ez a megközelítés hasonlít a közkedvelt, Angular nevű keretrendszer logikájához. Ezeket az osztályokat egyesével importálok, majd példányosítom az `app.js` fájlban. Az osztályok közötti kommunikációt egy „dependency injection” néven ismert módszerrel oldottam meg.

A TTK TO adatbázis elérése

Sajnos a TTK TO adatbázisának nincs saját API-ja, így a felhasználókat normál esetben kiszolgáló végpontot kellett felhasználnom. A Google Chrome fejlesztői eszközeinek segítségével megfigyeltem az oldal működését, ez alapján pedig meg tudtam azt szólítani saját programomból.

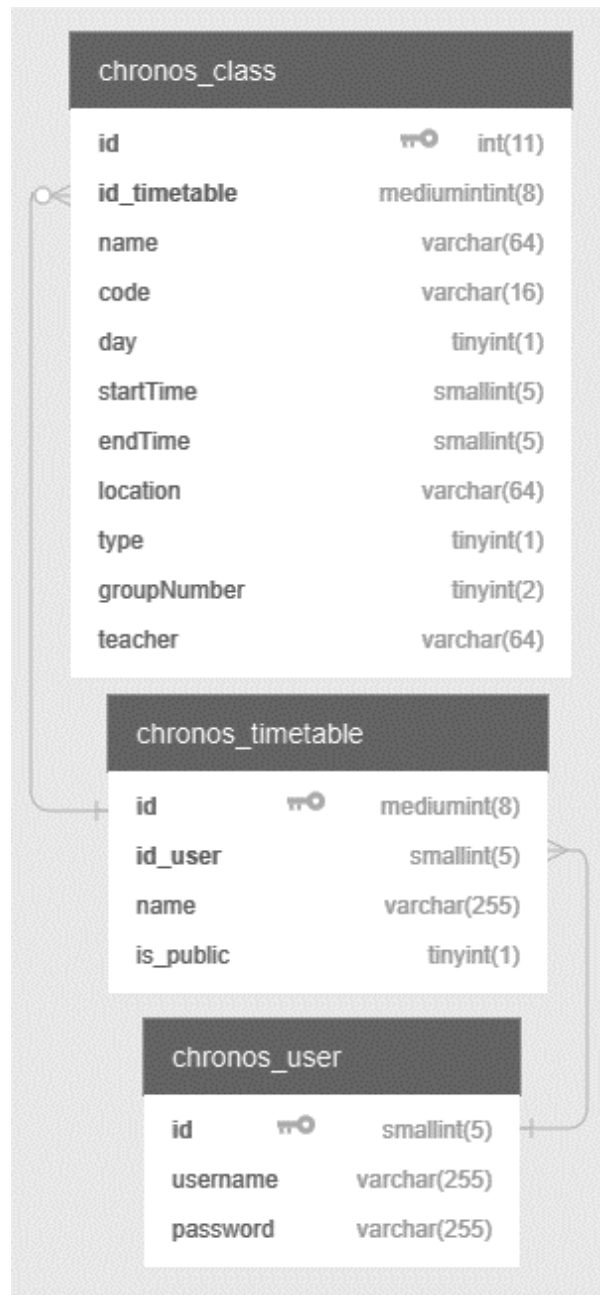
Mivel a TTK TO oldala másik domainen található, a CORS megkötések miatt nem tudtam felé közvetlenül az alkalmazásom Javascript kódjából AJAX hívást indítani. Ezt a problémát egy PHP szkript segítségével küszöböltem ki, mely egyfajta gateway-ként működik. A Javascript kódom ezt a PHP fájlt szólítja meg, majd pedig ez a PHP fájl intézi a lekérést a TTK TO szervere felé. A PHP környezetben nincs tiltva a CORS, így a lekérés sikeresen végbemegy.

Mivel ez nem egy általános felhasználásra tervezett API, a lekérésekre a válasz a TTK TO általi megjelenítésre szánt, HTML táblázatos formátumban érkezik meg. A fentebb említett PHP szkript felelős ezen válasz szabványos JSON formátumba történő konvertálásáért, melyet visszaküld a kliensoldali Javascript kód számára. A formátum konvertálása mellett más transzformációkat is elvégez a szkript, mint például a tanórák tantárgyak szerint történő csoportosítása, vagy a hibák kiszűrése, és jelzése a kliensoldali program felé.

Könyvtárszerkezet

- /
 - /font
 - Gidole-Regular.ttf
 - /script
 - app.js
 - lesson.js
 - popup.js
 - search.js
 - timetable.js
 - timetable_management.js
 - user_management.js
 - /style
 - general.css
 - popup.css
 - search_results.css
 - sidebar.css
 - timetable.css
 - /test
 - index.php
 - template.php
 - test.php
 - .htaccess
 - data.php
 - data.sql
 - database.php
 - index.php
 - rest.php

Adatbázisdiagram



19. ábra: Az alkalmazás által használt adatbázist leíró diagram

Az adatbázis tartalmaz továbbá minden táblához egy `test_` előtaggal ellátott másolatot is, mely táblákat az egységtesztek futtatásakor használja a rendszer, egyébként üresen tartja azokat.

Backend végpontok

A backend végpontjai a `/api/` útvonalon érhetőek el. Megvalósításuk az `api.php` fájlban található, mely az adatbázis eléréséhez a `database.php` fájlban lévő `Database` osztály metódusait használja. Az alábbi listában részletezem a végpontokat, HTTP-metódusok szerint csoportosítva. Az API nem használ GET metódusú végpontokat, mert az adatok csak oldalfrissítéskor kerülnek olvasásra, ekkor azonban a PHP szkript a végpontok használata nélkül képes azokat kinyerni az adatbázisból. Ha egy, az alábbi felsorolásban nem szereplő végpontot szólítunk meg, a szerver 404 NOT FOUND válasszal tér vissza. Amennyiben egy itt felsorolt végpontot egy általa nem támogatott metódussal szólítunk meg, a szerver 405 METHOD NOT ALLOWED választ ad. Nem várt, a kiszolgálón fellépő hiba esetén 500 INTERNAL SERVER ERROR választ kapunk.

- POST
 - `/sessions`
 - Új felhasználói munkamenet létrehozása bejelentkezéskor. A munkameneteket a PHP kezeli HTTP-sütik segítségével, így nincs szükség a munkamenet azonosítójának visszaadására.
 - Lehetséges válaszok
 - 400 BAD REQUEST: Hiányzó felhasználónév, illetve jelszó esetén.
 - 401 UNAUTHORIZED: Hibás felhasználónév, vagy jelszó esetén.
 - 200 OK: Sikeres azonosítás történt, a felhasználói munkamenet létrejött.
 - `/timetables`
 - Új órarend létrehozása az éppen bejelentkezett felhasználó számára. Kötelező megadni egy nevet, illetve opcionálisan tanórák is elhelyezhetőek közvetlenül a létrehozás után, amennyiben a kérésbe kerültek.
 - Lehetséges válaszok
 - 401 UNAUTHORIZED: Amennyiben a felhasználó nincs bejelentkezve.

- 400 BAD REQUEST: Amennyiben a kérés törzse nem tartalmazza a létrehozandó órarend nevét.
 - 201 CREATED: Az órarend sikeresen létrejött a megadott névvel, illetve tanórákkal. A válasz tartalmazza az új órarend azonosítóját.
- /classes
 - Tanóra létrehozása és beszúrása egy létező órarendbe. Kizárólag a bejelentkezett felhasználó saját órarendjébe lehetséges órát beszúrni, kötelező megadni az órarend azonosítóját.
 - Lehetséges válaszok
 - 401 UNAUTHORIZED: Amennyiben a felhasználó nincs bejelentkezve.
 - 400 BAD REQUEST: Amennyiben hiányzik az órarend azonosítója.
 - 403 FORBIDDEN: A felhasználó nem a saját órarendjének azonosítóját adta meg.
 - 201 CREATED: A tanóra bekerült az adatbázisba a kért órarend alá, a válasz törzse tartalmazza a létrehozott tanóra azonosítószámát.
- /users
 - Új felhasználói fiók létrehozása regisztráció folyamán. Szükséges megadni egy felhasználói nevet és jelszót, továbbá az adott felhasználói névnek egyedinek kell lennie.
 - Lehetséges válaszok
 - 400 BAD REQUEST: Hiányzik vagy a felhasználónév, vagy a jelszó.
 - 409 CONFLICT: A megadott névvel már létezik felhasználói fiók az adatbázisban.
 - 201 CREATED: A kért fiók sikeresen létrejött a megadott felhasználónévvel és jelszóval.

- DELETE
 - /sessions/current
 - Az éppen megnyitott felhasználói munkamenet törlése kijelentkezéskor.
 - Lehetséges válaszok
 - 401 UNAUTHORIZED: A felhasználó nincs bejelentkezve a rendszerbe.
 - 200 OK: A nyitott munkamenet sikeresen törlésre került, ezzel felhasználó kijelentkezett a rendszerből.
 - /timetables/{id}
 - A megadott azonosítószámú órarend törlése az adatbázisból. A felhasználónak be kell jelentkeznie, a megadott órarendnek pedig a saját tulajdonában kell lennie.
 - Lehetséges válaszok
 - 401 UNAUTHORIZED: Amennyiben a felhasználó nincs bejelentkezve.
 - 403 FORBIDDEN: A megadott azonosítószámú órarend nem az aktuálisan bejelentkezett felhasználó tulajdonában áll.
 - 200 OK: A megadott azonosítószámú órarend sikeresen törlésre került az adatbázisból.
 - /classes/{id}
 - A megadott azonosítószámú tanóra törlése az adatbázisból. A felhasználónak be kell jelentkeznie, a megadott tanórának pedig a saját órarendjében kell szerepelnie.
 - Lehetséges válaszok
 - 401 UNAUTHORIZED: Amennyiben a felhasználó nincs bejelentkezve.
 - 403 FORBIDDEN: A megadott azonosítószámú tanóra nem a felhasználó órarendjében szerepel.
 - 200 OK: A megadott azonosítószámú tanóra sikeresen törlésre került az adatbázisból.

- /users/me
 - Az éppen bejelentkezett felhasználó törlése az adatbázisból. Felhasználói fiók megszüntetésekor használt végpont.
 - Lehetséges válaszok
 - 401 UNAUTHORIZED: Amennyiben a felhasználó nincs bejelentkezve.
 - 200 OK: A felhasználó fiókja sikeresen törlésre került a rendszerből, az aktív munkamenet lezárult.
- PATCH
 - /timetables/{id}
 - A felhasználó saját órarendjének módosítására szolgáló végpont. Az órarend neve és láthatósága változtatható meg segítségével.
 - Lehetséges válaszok
 - 401 UNAUTHORIZED: Amennyiben a felhasználó nincs bejelentkezve.
 - 404 NOT FOUND: Amennyiben a megadott azonosítószerű órarend nem létezik.
 - 403 FORBIDDEN: A felhasználó valaki más órarendjét próbálja módosítani.
 - 200 OK: Az órarend sikeresen módosításra került az adatbázisban.
 - /users/me
 - A felhasználó saját adatainak frissítésére szolgáló végpont. Elég csak a megváltozott adatokat elküldeni a szervernek.
 - Lehetséges válaszok
 - 401 UNAUTHORIZED: Amennyiben a felhasználó nincs bejelentkezve.
 - 200 OK: A felhasználó adatai sikeresen frissítésre kerültek az adatbázisban.

Osztálydiagram

Szerveroldalon

Database
conn : Integer
<<create>> __construct() <<destroy>> __destruct() createUser(username : String,password : String) : Integer userExists(username : String) : Boolean getHashedPassword(username : String) : String getUserId(username : String) : Integer getTimetables(user_id : Integer) getLatestTimetableId(user_id : Integer) : Integer updateUser(id : Integer,data) : Boolean deleteUser(id : Integer) : Boolean createTimetable(user_id : Integer,name : String) : Integer readTimetable(id : Integer) readTimetableLessons(id : Integer) getTimetableOwner(timetable_id : Integer) : Integer getTimetableOwnerByLesson(lesson_id : Integer) : Integer updateTimetable(id : Integer,data) : Boolean insertClasses(timetable_id : Integer,days) deleteTimetable(id : Integer) : Boolean createLesson(timetable_id : Integer,name : String,code : String,day : Integer,startTime : Integer,endTime : Integer,location : String,type : Integer,groupNumber : Integer,teacher : String) : Integer deleteLesson(id : Integer) : Boolean

20. ábra: A Database osztály felépítése

Kliensoldalon

Lesson
typeName dayNames name : String code : String day : Integer startTime : Integer endTime : Integer location : String type : Integer groupNumber : Integer teacher : String id : Integer
constructor(name : String,code : String,day : Integer,startTime : Integer,endTime : Integer,location : String,type : Integer,groupNumber : Integer,teacher : String,id : Integer) overlapsWith(otherLesson : Lesson) : Boolean hasTimeData() : Boolean startTimeString() : String endTimeString() : String dayString() : String typeString() : String formatTimestamp(timestamp : Integer) : String equals(that : Lesson) : Boolean

21. ábra: A Lesson osztály felépítése

Popup
timetable : Timetable search : Search lesson : Lesson
<<create>> constructor(closeBtn,removeBtn,altsBtn) close() remove() displayAlternatives() populate(lesson : Lesson) newOperation()

22. ábra: A Popup osztály felépítése

Search
searchTimer : Integer results timetable : Timetable searchField resultsList
<<create>> constructor(resultsField,searchField,timetable : Timetable) selectClass(event) displaySearchPanel() fireSearch(term : String)

23. ábra: A Search osztály felépítése

Timetable
element id : Integer unsaved : Boolean popup : Popup days
constructor(element,popup : Popup) addLesson(lesson : Lesson) removeLesson(lesson : Lesson) findLesson(lesson : Lesson) : Integer unloadWarning(event) : String renderLessons(onlyThisDay : Boolean) handleLessonSelection(event) save(name : String)

24. ábra: A Timetable osztály felépítése

TimetableManagement
timetableTitle : String
<pre> <<create>> constructor(listOfTimetables,timetableTitle : String) timetableClicked(event) deleteTimetable(timetableName : String,timetableId : Integer) renameTimetable(timetableName : String,timetableId : Integer,timetableTitleInList,timetableTitle toggleTimetablePublicity(timetableId : Integer,setToPublic : Boolean,icon) </pre>

25. ábra: A *TimetableManagement* osztály felépítése

UserManagement
<pre> timetable : Timetable loginForm registerButton </pre>
<pre> <<create>> constructor(logoutButton,loginForm,registerButton,timetable : Timetable,changePasswordButton,deleteUserButton) changePassword() deleteUser() login(event) register() logout() disableLoginForm() enableLoginForm() </pre>

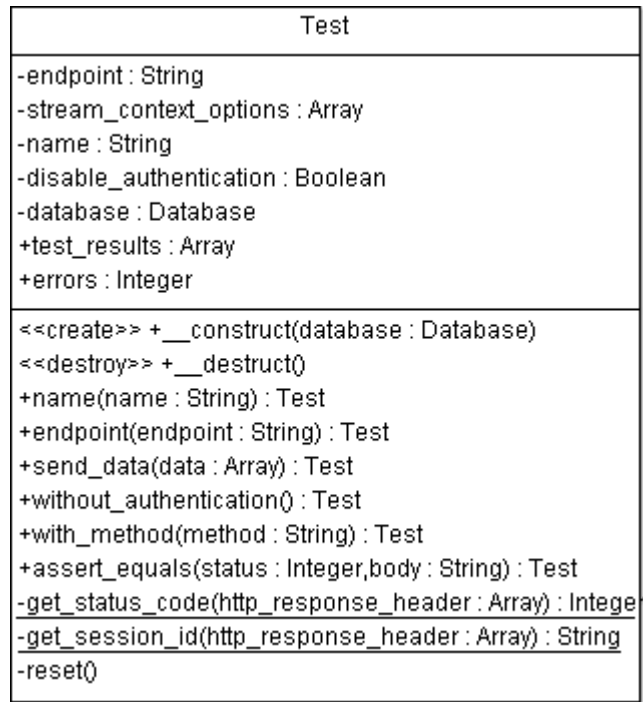
26. ábra: A *UserManagement* osztály felépítése

Tesztesetek

Egységtesztek

Az alkalmazás szerveroldali API-ja teljes mértékben lefedett egységtesztekkel. A tesztelésről egy saját fejlesztésű tesztelőosztály gondoskodik, mely a `test` könyvtárban található. Az egyes tesztesetek, illetve a tesztelés menete az ugyanebben a könyvtárban található `index.php` fájlba került. A tesztek futtatásához így elég megnyitnunk a `test` könyvtárat böngészőnkben, hisz az Apache webservert alapértelmezetten az `index` nevű fájl tölti be.

A tesztelőosztály lehetőséget biztosít a végpont és a metódus megválasztására, továbbá beállíthatjuk a kérés törzsében küldendő adatot, illetve kikapcsolhatjuk az autentikációs süti elküldését is. A lekérés befejeztével ellenőrizhetjük a válasz státusz kódját és tartalmát is. Az elkészített egységteszteket a lentebbi táblázat foglalja össze.



27. ábra: A tesztelő osztálydiagramja

Lekérés				Várt válasz	
Végpont	Metódus	Törzs	Autentikáció	Státusz	Törzs
/users	POST	-	nem	400	-
/users	POST	username: teszt	nem	400	-
/users	POST	password: teszt	nem	400	-
/users	POST	username: teszt password: teszt	nem	201	1
/users	POST	username: teszt password: teszt	nem	409	-

Lekérés				Várt válasz	
Végpont	Metódus	Törzs	Autentikáció	Státusz	Törzs
/users	POST	username: teszt2 password: teszt	nem	201	2
/sessions	POST	-	nem	400	-
/sessions	POST	username: teszt	nem	400	-
/sessions	POST	password: mas_jelszo	nem	400	-
/sessions	POST	username: teszt password: mas_jelszo	nem	401	-
/sessions	POST	username: teszt passdword: teszt	nem	200	-
/timetables	POST	-	nem	401	-
/timetables	POST	-	igen	400	-
/timetables	POST	name: Teszt órarend	igen	201	1
/classes	POST	-	nem	401	-
/classes	POST	-	igen	400	-
/classes	POST	id: 1	igen	201	1
/sessions/current	DELETE	-	nem	401	-
/sessions/current	DELETE	-	igen	200	-
/sessions	POST	username: teszt2 passdword: teszt	nem	200	-

Lekérés				Várt válasz	
Végpont	Metódus	Törzs	Autentikáció	Státusz	Törzs
/timetables	POST	name: Órarend	igen	201	2
/classes	POST	id: 2	igen	201	2
/timetables/2	PATCH	-	nem	401	-
/timetables/1	PATCH	-	igen	403	-
/timetables/2	PATCH	name: Új név	igen	200	-
/users/me	PATCH	-	nem	401	-
/users/me	PATCH	password: uj_jelszo	igen	200	-
/classes/2	DELETE	-	nem	401	-
/classes/1	DELETE	-	igen	403	-
/classes/2	DELETE	-	igen	200	-
/timetables/2	DELETE	-	nem	401	-
/timetables/1	DELETE	-	igen	403	-
/timetables/2	DELETE	-	igen	200	-
/users/me	DELETE	-	nem	401	-
/users/me	DELETE	-	igen	200	-

Manuális tesztek

Regisztráció

Amennyiben van elkezdett órarend

A felhasználó által látott órarend nem üres, azaz már adott hozzá tanórákat. Ebben az állapotban kísérli meg a regisztrációt. Ha a felhasználónév, vagy a jelszó mező üres a regisztráció beküldésekor, az oldal hibaüzenetet jelez ki. Amennyiben egy már létező felhasználónevet próbál regisztrálni, szintén hibaüzenetet kap. Ellenkező esetben létrejön a felhasználói fiók a megadott adatokkal. A rendszer ekkor rákérdez, hogy kívánjuk-e elmenteni a megkezdett órarendet. Pozitív válasz esetén megkérdezi az új órarend nevét. Ezután a rendszer automatikusan beléptet. Amennyiben elmentettük az elkezdett

órarendet, az előbb megadott néven jelenik meg. Ellenkező esetben egy új, üres órarend jön létre és jelenik meg.

Amennyiben nincs elkezdett órarend

Ugyanaz történik, mint az előző esetben, azzal a különbséggel, hogy nem történik órarend-mentés, az automatikus bejelentkeztetés után pedig mindenképpen egy új, üres órarend jelenik meg.

Bejelentkezés

Van elkezdett órarend

Az aktuálisan megnyitott órarend nem üres, és így próbálunk meg bejelentkezni. Ha üresen hagyjuk a felhasználónév, vagy a jelszó mezőt, a rendszer hibaüzenetet jelenít meg. Ha nem létező felhasználónevet, vagy érvénytelen jelszót adunk meg, szintén hibaüzenetben figyelmeztet az oldal. Ha helyes adatokat adunk meg, a rendszer megkérdezi, hogy a megkezdett órarendet el szeretnénk-e menteni. Amennyiben igen, az oldal megkérdezi tőlünk az új órarend nevét. Ezt megadva az órarend mentésre kerül az adatbázisba, az oldal pedig automatikusan frissül, hogy a felhasználói adataink jelenhessenek meg. Az újonnan létrehozott órarend kerül megnyitásra.

Nincs elkezdett órarend

Az előző esettel megegyező folyamat játszódik le, azonban a rendszer nem kérdez rá, hogy menteni szeretnénk-e az elkezdett órarendet. Bejelentkezés után a legutoljára létrehozott órarend jelenik meg.

Órák keresése

Tanórák kereséséhez az oldalsávon található keresőmezőbe kell gépelnünk, mely a bevétel befejezése után automatikusan elindítja a keresést. Az alább tárgyalt esetek merülhetnek fel.

Nem érhető el a TTK TO adatbázis

Ebben az esetben a találati lista helyén egy figyelmeztető üzenet jelenik meg, nem tudunk tanórákat keresni.

Nem létező tárgynév

Ekkor a találati lista helyén megjelenő üzenet figyelmezteti a felhasználót a keresés sikertelenségére.

Aktuális órarendben szereplő tanóra

Amennyiben a keresőkifejezésre vannak találatok, azok tantárgyak szerint csoportosítva jelennek meg a keresőmező alatti találati listában. Ha az adott tanórát már hozzáadtuk az órarendünkhöz, a találati listában kiemelt háttérrel jelenik meg. A jobb oldalon egy mínusz-piktogram jelzi, hogy a tanóra való kattintással törölhetjük azt az órarendünkből.

Aktuális órarendben nem szereplő tanóra

Ha még nem szerepel az adott tanóra aktuális órarendünkben, a neve mellett egy plusz-piktogram jelzi, hogy kattintással hozzáadhatjuk ahhoz.

Tanórák hozzáadása

Tanórát kizárólag a keresési találatok közül tudunk hozzáadni órarendünkhöz. Amennyiben egy, az aktuális órarendben nem szereplő tanóra kattintunk, az megjelenik az órarendünkben. Az oldal újratöltésével ellenőrizhetjük, hogy a szerveroldalon is elmentődött az adatbázisba.

Tanórák törlése

Megnyitott órarendünkből tanórát törölni kétféleképpen tudunk: a keresési találatok közül, vagy az órarendben a tanóra kattintva, majd a felugró ablakban a törlés lehetőséget választva.

Keresésből

Ha egy olyan tanóra kattintunk a keresési eredmények között, mely már szerepel a jelenlegi órarendünkben, akkor ezzel eltávolítjuk. Ekkor eltűnik az órarendből, oldalfrissítéssel pedig megbizonyosodhatunk róla, hogy az adatbázisból is törlésre került.

Órarendből

Amennyiben rákattintunk egy tanórára az órarendünkben, majd a felugró ablakban a törlés lehetőséget választjuk, el fog tűnni. Az előző esethez hasonlóan, újratöltéssel ellenőrizhetjük ennek sikerességét.

Órarend létrehozása

Bejelentkezés után az oldalsávon lévő “Órarendek” szekcióban található “+” gombbal hozhatunk létre új, üres órarendet. Ekkor azonnal átirányít minket a rendszer a frissen elkészített órarendre.

Órarend átnevezése

Órarendet az oldalsávon, órarendjeink listájában, a szerkesztés gombra kattintással nevezhetünk át. Ekkor egy felugró ablakba kell beírunk az órarend új nevét.

Ha az átnevezendő órarend van megnyitva

Ebben az esetben az órarend neve az oldalsávon és a fejlécben is az újonnan megadott névre változik. Újratöltéssel ellenőrizhetjük, hogy az adatbázisban is mentésre került a változtatás.

Ha másik órarend van megnyitva

Ekkor csak az oldalsávon változik meg az órarend neve, sikerességét szintén oldalfrissítéssel ellenőrizhetjük.

Órarend láthatóságának módosítása

Az órarendek listájában adott órarend láthatóságát a szemet ábrázoló piktogrammal kapcsolhatjuk ki vagy be. Változtatásunkat könnyen ellenőrizhetjük, ha megnyitjuk az adott órarendet egy új ablakban, inkognitó módban, bejelentkezés nélkül.

Órarend törlése

Órarendet az oldalsávon, a neve mellett található szemétkosarat ábrázoló piktogramra kattintással törölhetünk. A rendszer megerősítést kér az órarend tényleges törlése előtt.

Ha a törlendő órarend van megnyitva

Ha az éppen megnyitott órarendet töröljük, az oldal frissül, a rendszer pedig figyelmeztet, hogy a keresett órarend nem található. Ez jelzi, hogy az adott azonosítószámú órarend ténylegesen törlődött az adatbázisból.

Ha másik órarend van megnyitva

Ha egy másik órarendet törölünk, neve eltűnik az oldalsávból. Oldalfrissítéssel, illetve a törölt órarend URL-jének felkeresésével ellenőrizhetjük a törlés sikerességét.

Jelszó módosítása

Az oldalsávon a jelszó módosítása gombra kattintva egy felugró ablakban kéri tőlünk a rendszer az új jelszót.

Üres jelszó

Ha üresen hagyjuk a mezőt és nem vonjuk vissza jelszóváltoztatási szándékunkat, a rendszer hibaüzenetet ad.

Legalább 1 karakteres jelszó

Amennyiben megadunk egy új jelszót és megerősítjük szándékunkat, a rendszer egy felugró ablakban tájékoztat bennünket a jelszóváltoztatás sikerességéről.

Kijelentkezés

Az oldalsávon a kijelentkezés gombra kattintva a teljes oldal frissül, mellyel egyidőben megtörténik a felhasználói munkamenet lezárása. A frissítés után kijelentkezett állapotban láthatjuk az oldalt, ahol egy üres órarend jelenik meg.

Felhasználó törlése

Ha megerősítjük a fiókunk törlésére vonatkozó szándékunkat a felugró ablakban, a rendszer a háttérben törli felhasználói fiókunkat és a hozzá tartozó információkat az adatbázisból, majd kijelentkeztet minket. Ezt ellenőrizhetjük egy bejelentkezési kísérlettel, mely hibára fog futni.

ÖSSZEGZÉS

Véleményem szerint a felvázolt problémára sikerült elkészítenem egy felhasználóbarát, könnyen használható alkalmazást, mely segítséget nyújthat hallgatótársaimnak. A program elkészítése során sok új ismeretet szereztem a webes technológiákról, melyeket később munkám során is használni tudok majd.

Továbbfejlesztési lehetőségek

A programot több módon is lehet a jövőben fejleszteni, mind funkcionálisan, mind felhasználói élmény tekintetében.

A program jelenleg a 2018/2019-es tanév első félévének tanóráival működik. Ezt a forráskódban lehet megváltoztatni. Egy jövőbeli verzióban az alkalmazás kezelhetné a félévek múlását, illetve a felhasználóknak lehetőséget lehetne adni félév kiválasztására. Így egy órarend egy adott félévhez kötődne, de egy félévhez több órarend-változatot is készíthetnénk.

Jelenleg az alkalmazás használatához regisztrációra van szükség, ezt ki lehetne egészíteni, esetleg teljes mértékben lecserélni külső autentikációs szolgáltatások használatával. Ilyen például a bejelentkezés Google, Facebook, esetleg ELTE felhasználói fiókkal.

A felhasználói élmény tekintetében sokat jelentene a program mobileszközökre való optimalizálása, hisz így akár számítógéptől távol is könnyen használható lenne.

Egy másik lehetséges fejlesztési irány a tanórákhoz előnézeti funkció hozzáadása. Ez úgy működne, hogy amikor a felhasználó a találati listában egy tanóra fölé viszi a kurzort, a tanóra előnézeti képe megjelenne az aktív órarendben megfelelő kiemeléssel, például halványabb, áttetsző háttérrel. Így még jobban láthatóvá válnának az esetleges ütközések egy új tanóra hozzáadása előtt.

Hasonló módszerrel lehetne megoldani az alternatív időpontok keresését is, ahol az adott tárgyból az összes többi tanóra megjelenne más kiemeléssel az órarendben, majd közülük választhatnánk egy másikat.

További fejlesztési lehetőség a TTK TO adatbázisának lokális kiváltása. Ezzel a program tárigénye némileg megnőne, azonban gyorsabb és részletesebb keresést tenne lehetővé, valamint nem függene a távoli szerver pillanatnyi elérhetőségétől. Ezt legegyszerűbben egy ütemezett szkripttel lehetne megoldani, ami meghatározott időközönként, például havonta, lekérné a teljes TTK TO adatbázist, majd a megfelelő feldolgozás után elmentené a Chronos saját szerverén. Egyedi azonosító használatával a felhasználók órarendjei a konkrét tanóra-adatok helyett erre a központi táblára hivatkoznának, így esetleges adatváltozás esetén is mindig az aktuális információkat jelenítenék meg.

IRODALOMJEGYZÉK

[1] <https://hu.wikipedia.org/wiki/PHP>

Elérés dátuma: 2018.11.05.

[2] <https://hu.wikipedia.org/wiki/MySQL>

Elérés dátuma: 2018.11.05.

[3] <https://hu.wikipedia.org/wiki/JavaScript>

Elérés dátuma: 2018.11.05.

[4] <https://hu.wikipedia.org/wiki/HTML>

Elérés dátuma: 2018.11.05.

[5] https://hu.wikipedia.org/wiki/Cascading_Style_Sheets

Elérés dátuma: 2018.11.05.