

Budapesti Műszaki Szakképzési Centrum Verebélő László

Technikuma

506131203 Szoftverfejlesztő és -tesztelő szakma

Digitális napló

Készítette: Oláh Bence, Németh Attila

Konzulens tanár: Horváth Attila, Somogyi Erika

2024. április

NYILATKOZAT A VIZSGAREMEK EREDETISÉGÉRŐL

Alulírott Oláh Bence és Németh Attila a BMSZC Verebélly László Technikumának 506131203 Szoftverfejlesztő és -tesztelő képzésében részt vevő tanulók büntetőjogi felelősségünk tudatában nyilatkozunk és aláírásunkkal igazoljuk, hogy a digitális napló című vizsgaremek saját munkánk, és abban betartottuk az iskola által előírt, a vizsgaremek készítésére vonatkozó szabályokat.

Tudomásul vesszük, hogy a vizsgaremekben plágiumnak számít:

- szó szerinti idézet közlése idézőjel és hivatkozás nélkül,
- tartalmi idézet hivatkozás megjelölése nélkül,
- más publikált gondolatainak saját gondolatként való feltüntetése.

E nyilatkozat aláírásával tudomásul vesszük továbbá, hogy plágium esetén szakdolgozatunk visszautasításra kerül.

Budapest, 2024. április 17.

Oláh Bence

Tanuló aláírása

Németh Attila

Tanuló aláírása

Tartalomjegyzék

1. Bevezetés.....	3
1. Követelményspecifikáció	4
2. Fejlesztői dokumentáció.....	5
1. Alkalmazott fejlesztői eszközök.....	5
2. Alkalmazott technológiák.....	5
3. Futtatási környezet.....	5
4. Adatbázis fejlesztése	5
5. Főbb algoritmusok.....	9
6. Tesztelés	12
3. Felhasználói dokumentáció, használati kézikönyv	13
1. Belépés	13
2. Általános funkciók	14
3. Diák funkciók	17
4. Tanár funkciók	18
5. Intézményvezető/Titkár funkciók.....	20
6. Adminisztrátor funkciók	24
7. Asztali alkalmazás	24
4. Továbbifejlesztési lehetőségek	24
5. Összegzés.....	25
6. Ábrajegyzék	26

1. Bevezetés

A mi téma választásunk a digitális tanári/iskolai naplóra esett, egy nem gyakori tematika, amit több platformon lehet implementálni. A témát többek között azért is választottuk, mert egy olyan feladatot

kerestünk, ami számunkra kihívásokkal teli és komplex adatbázist igényel, de segítségként a Kréta¹ digitális ellenőrző több éves jelenléte, használata egy jó alapot adott. Kíváncsiak voltunk, hogyan tudjuk elkészíteni saját tudásunk szerint és a megoldások során, milyen új dolgokat tanulhatunk, amiket magunkkal vihetünk az életünk során. A digitális napló olyan problémát old meg, amely valós igényekhez kapcsolódik. A diákok és pedagógusok mindennapi életét teheti egyszerűbbé, és ezáltal hosszú távon is hasznos lehet és a digitalizáció korszakában szinte már megkövetelt, hogy nem papíron tároljuk ezeket az adatokat, mellékesen egy alkalmazással egyszerűbbé tehetjük az adatok kezelését.

1. Követelményspecifikáció

A webes felületet igyekeztünk barátságossá tenni, hogy a felhasználói élményt fokozzuk, még a színtémát is lehet változtatni. A webes felületre bármely jogosultságú felhasználó be tud lépni és tudja használni a saját jogosultsága szerint. A napló lehetővé teszi a diákok teljesítményének nyomon követését. A szülők és pedagógusok könnyen ellenőrizhetik a diákok jegyeit és a fejlődésüket.

A számítógépes applikáció leginkább adminisztratív funkciókat lát el, például: adatok feltöltése, exportálása.

A mobilalkalmazás egy webes megjelenítés, ami a weboldalt jeleníti meg és ugyanazokat a funkciókat látja el.

Több, más funkciót ellátó jogok léteznek:

- Diákok hozzáférnek a saját órarendjükhöz, adataikhoz, feljegyzéseikhez
- Tanárok/osztályfőnökök hozzáférnek saját órarendjükhöz, adataikhoz, tudnak hiányzást írni, jegyeket beírni
- Intézmény vezetők tudnak órákat/órarendeket feltölteni, tanulókat és tanárokat bevinni az adatbázisba (kvázi „regisztrálni”) különböző adatokkal, tantárgyakat rögzíteni és tanárokhöz rendelni
- Adminisztrátorok szinte ugyanazt látják, mint az intézményvezetők, azzal a különbséggel, hogy ők tudnak profilokat törölni és jelszavakat változtatni. Ez a helyi rendszergazdáknak szánt jogosultság.

¹ <https://www.e-kreta.hu/>

2. Fejlesztői dokumentáció

1. Alkalmazott fejlesztői eszközök

- Visual Studio
- Visual Studio Code
- MySQL Workbench
- XAMPP

2. Alkalmazott technológiák

- Programozási nyelvek: PHP, Javascript, C#
- Webes technológiák: HTML, CSS, Bootstrap(v. 5.3), Laravel(v. 10)
- Adatbázis motor: MariaDB

3. Futtatási környezet

A weblap bármilyen modern böngészőben elfut és reszponzív, tehát mobilon is megnyitható és ergonomikus. Az asztali alkalmazás windows 10 és a fölötti windows verziókban fut. A mobil alkalmazás androidos környezetben, 7.0 és azon fölüli verziókban futtatható.

4. Adatbázis fejlesztése

Az adatbázis 14 db táblából áll, melyek közül van egy kapcsoló tábla (**tanított tábla**).

Az adatbázis(enaplo) tartalmazza a diákok adatait(**diákok tábla**[**oktatási azonosító, név, születési dátum, születési hely, anyja neve, lakcím, osztály azonosító, felhasználói azonosító**]), az osztályokat és a hozzá tartozó osztályfőnököket(**osztályok tábla**[**osztály azonosító, osztály megnevezése, osztályfőnök azonosító**]), tanárokat(**tanárok tábla**[**tanár azonosító, név, felhasználói azonosító**]), az iskolában tanított tantárgyakat(**tantárgyak tábla**[**tantárgy azonosító, tantárgy megnevezése**]), melyik tanár milyen tantárgyakat tanít(**tanított tantárgyak egyed**[**tanított tantárgyak azonosító, tanár azonosító, tantárgy azonosító**]), továbbá órarend táblát(**órarend tábla**[**órarend azonosító, osztály azonosító, dátum, órák száma, tanított tantárgy azonosító, terem**]) is tartalmaz. A diákok jegyét(**értékelések tábla**[**értékelések azonosító, órarend azonosító, oktatási azonosító, értékelés leírás, értékelés, értékelés százalék, idő azonosító, típus azonosító**]) (**értékelés típusok tábla**[**típus azonosító, típus megnevezése**]) (**értékelés időpontok tábla**[**idő azonosító, idő**]) és hiányzásait(**hiányzások tábla**[**hiányzások azonosító, órarend azonosító, oktatási azonosító, egész**]).

órás-e, késés percben, igazolt-e]) is lehet naplózni, igazolni(igazolások tábla[igazolás azonosító, igazolás megnevezése]).

Az adatbázis tartalmaz egy felhasználók táblát(felhasználók tábla[felhasználó azonosító, felhasználói név, jelszó, email, telefonszám, jog azonosító, téma azonosító]) és a hozzájuk tartozó jogokat(jogok tábla[jog azonosító, jog megnevezése]), témákat(témák tábla[téma azonosító, téma megnevezése]) is.

Táblák, egyedek:

Diákok tábla (diakok)

Oktatási azonosító (diak_id)	egész szám, PK
Név (diak_nev)	szöveges, maximum 45 karakter
Születési dátum (diak_szuldatum)	dátum
Születési hely (diak_szulhely)	szöveges, maximum 45 karakter
Anyja neve (diak_anya)	szöveges, maximum 45 karakter
Lakcím (diak_lakcim)	szöveges, maximum 45 karakter
Osztály azonosító (oszt_id)	egész szám, FK, az Osztályok táblában PK
Felhasználói azonosító(fel_id)	egész szám, FK, a felhasználók táblában

Osztályok tábla (osztalyok)

Osztály azonosító (oszt_id)	egész szám, PK
Osztály megnevezése (oszt_nev)	szöveges, maximum 45 karakter
Osztályfőnök azonosító (oszt_fonok)	egész szám, FK, a Tanárok táblában PK

Tanárok tábla (tanarok)

Tanár azonosító (tanar_id)	egész szám, PK
Tanár neve (nev)	szöveges, maximum 45 karakter
Felhasználó azonosító (felhasznalo_id)	egész szám, FK, a Felhasználók táblában PK

Tantárgyak tábla (tantargyak)

Tantárgy azonosító (tantargy_id)	egész szám, PK
Tantárgy megnevezése (tant_nev)	szöveges, maximum 45 karakter

Tanított tantárgyak tábla (tanitott)

Tanított tantárgyak azonosító (tanit_id)	egész szám, PK
Tanár azonosító (tanar_id)	egész szám, FK, a Tanár táblában PK
Tantárgy azonosító (tant_id)	egész szám, FK, a Tantárgyak táblában PK

Órarend tábla (orak)

Órarend azonosító (ora_id)	egész szám, PK
Osztály azonosító (oszt_id)	egész szám, FK, az Osztályok táblában PK
Dátum (ora_datum)	dátum
Óra sorszáma (ora_szam)	egész szám
Tanított tantárgy azonosító (tanit_id)	egész szám, FK, Tanított tantárgyak táblában PK
Terem (ora_terem)	szöveges, maximum 45 karakter

Értékelések tábla (ertekelesek)

Értékelés azonosító (ert_id)	egész szám, PK
Órarend azonosító (ora_id)	egész szám, FK, az Órarend táblában PK
Oktatási azonosító (diak_id)	egész szám, FK, a Diákok táblában PK
Leírás (ert_leiras)	szöveges, maximum 45 karakter
Értékelés, mint jegy (ert_jegy)	egész szám, maximum 1 karakter
Idő azonosító (ido_id)	egész szám
Típus azonosító (tip_id)	egész szám, FK, Értékelés típusok táblában PK

Értékelés típusok tábla (erttipusok)

Típus azonosító (tip_id)	egész szám, PK
leírás (tip_nev)	szöveges, maximum 45 karakter

Értékelés időpontok tábla (ertidopontok)

Idő azonosító (ido_id)	egész szám, PK
leírás (ido_nev)	szöveges, maximum 45 karakter

Hiányzások tábla (hianyzasok)

Hiányzások azonosító (hia_id)	egész szám, PK
Óra azonosító (ora_id)	egész szám, FK, az Órarend táblában PK
Oktatási azonosító (diakok_oktazon)	egész szám, FK, a Diákok táblában PK
Késik-e (hia_keses)	egész szám, 0 vagy 1 lehet
Késlett perc (hia_perc)	egész szám, lehet NULL
Igazolás azonosító (iga_id)	egész szám, lehet NULL, FK, az Igazolások táblában PK

Igazolások tábla (igazolasok)

Igazolás azonosító (iga_id)	egész szám, PK
Igazolás megnevezése (iga_nev)	szöveges, maximum 45 karakter

Témák tábla (temak)

Téma azonosító (tema_id)	egész szám, PK
Megnevezése (tema_nev)	szöveges, maximum 45 karakter

Felhasználók tábla (felhasznalok)

Felhasználó azonosító (fel_id)	egész szám, PK
Név (fel_nev)	szöveges
Jelszó (fel_jelszo)	szöveges

Email (fel_email)

szöveges

Telefonszám (fel_telszam)

egész szám, lehet NULL

Jog azonosító (jog_id)

egész szám, FK, a Jogok táblában **PK**Téma azonosító (**tema_id**)

egész szám, FK a Témák táblában PK

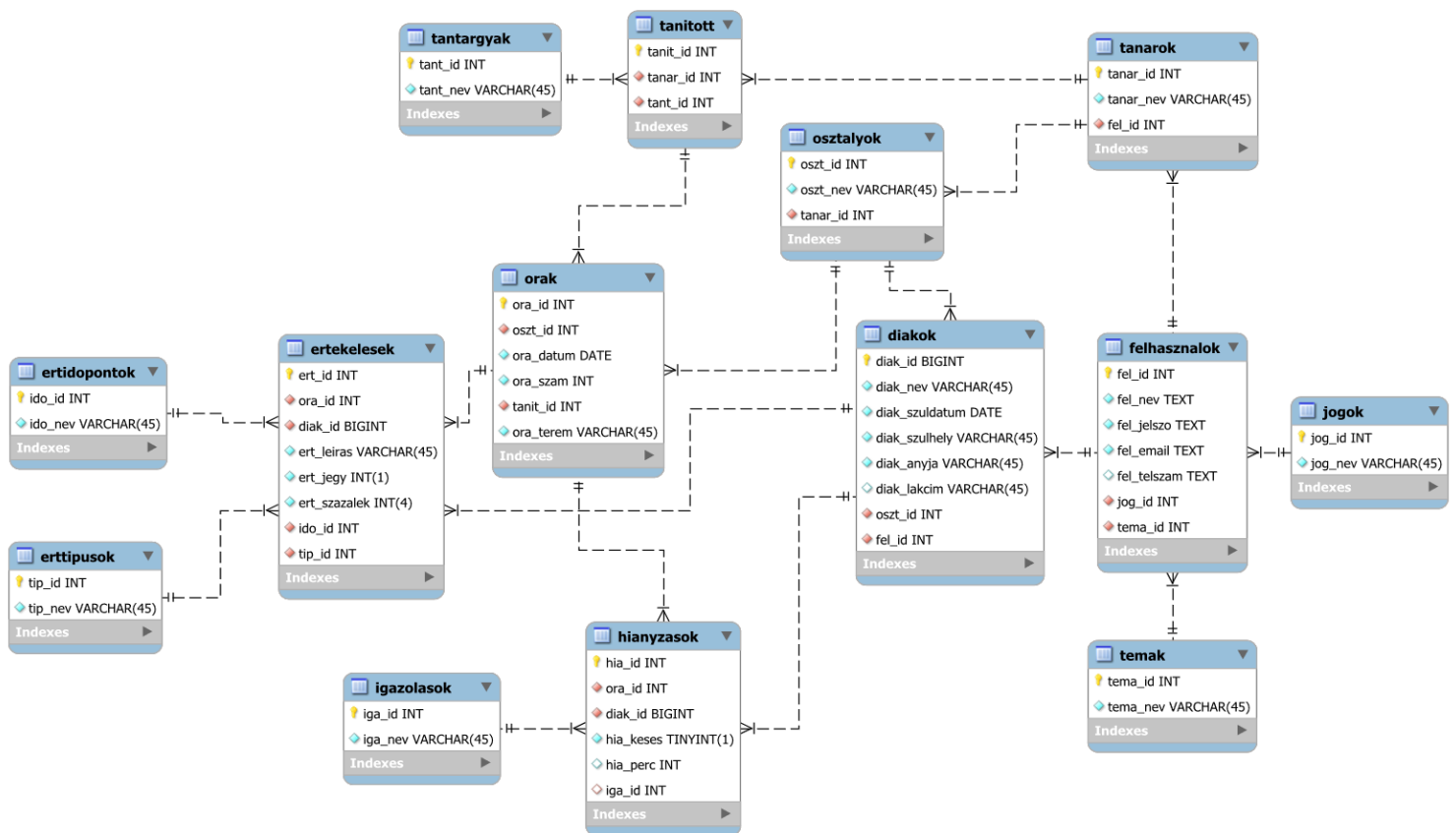
Jogok tábla (jogok)

Jog azonosító (**jog_id**)

egész szám, PK

Jog megnevezése (**jog_nev**)

szöveges, maximum 45 karakter



1. ábra EER modell az adatbázisról

5. Főbb algoritmusok

Itt a főoldalnál lefutó laravel funkciót látjuk. Megnézzük, hogy be van-e jelentkezve valaki. Ha diák az illető, akkor átadja a „fooldal” nézetnek az adatbázisból

- a bejelentkezett adatait
- a jogát

- témáját
- jövőbeli óráit
- jegyeit
- hiányzásait

Ha nem diákról beszélünk, akkor csak a

- bejelentkezett adatait
- jogát
- témáját
- elkövetkezendő óráit

adja át.

Ha nincs bejelentkezve senki, akkor visszadob a bejelentkezés oldalra, ez mindenhol igaz.

```
public function Foooldal(){
    if (Auth::check()){
        $user = User::find(Auth::id());
        $jog = $user->jog_id;
        if ($jog == 1){
            return view('foooldal',[
                'user' => $diak::where('fel_id', '=', User::find(Auth::id())->fel_id->get())->first(),
                'jog' => $jog,
                'tema' => tema::find(User::find(Auth::id())->tema_id->tema_nev,
                'jovo' => ora::selectraw('tantargyak.tant_nev, tanarok.tanar_nev, orak.ora_terem, orak.ora_datum, orak.ora_szam')
                ->join('tanitott','tanitott.tanit_id','orak.tanit_id')
                ->join('tantargyak','tantargyak.tant_id','tanitott.tant_id')
                ->join('tanarok','tanarok.tanar_id','tanitott.tanar_id')
                ->where('oszt_id','=', $diak::where('fel_id','=', User::find(Auth::id())->fel_id->first())->oszt_id)
                ->where('orak.ora_datum','>=',date('Y-m-d'))
                ->orderBy('ora_datum')->orderBy('ora_szam')->limit(5)->get(),
                'jegyek' => ertekeles::selectraw('orak.ora_datum, orak.ora_szam, ertekelesek.*, tantargyak.tant_nev, erttipusok.tip_nev, ertidopontok.ido_id')
                ->join('erttipusok','erttipusok.tip_id','ertekelesek.tip_id')->join('ertidopontok','ertidopontok.ido_id','ertekelesek.ido_id')
                ->join('orak','orak.ora_id','ertekelesek.ora_id')->join('tanitott','tanitott.tanit_id','orak.tanit_id')
                ->join('tantargyak','tantargyak.tant_id','tanitott.tant_id')->where('ertekelesek.diak_id','=', $diak::where('fel_id','=', Auth::id())->get())->first())->diak_id)
                ->groupby('ertekelesek.ert_id')->orderBy('ora_datum','desc')->orderBy('ora_szam','desc')->get(),
                'hianyzasok' => hianyzas::selectraw('orak.ora_datum, orak.ora_szam, hianyzasok.', tantargyak.tant_nev')
                ->join('orak','orak.ora_id','hianyzasok.ora_id')->join('tanitott','tanitott.tanit_id','orak.tanit_id')
                ->join('tantargyak','tantargyak.tant_id','tanitott.tant_id')->where('hianyzasok.diak_id','=', $diak::where('fel_id','=', Auth::id())->get())->first())->diak_id)
                ->groupby('hianyzasok.hia_id')->orderBy('ora_datum','desc')->orderBy('ora_szam','asc')->get()
            ]));
        } else{
            return view('foooldal',[
                'user' => tanar::where('fel_id', '=', User::find(Auth::id())->fel_id->get())->first(),
                'jog' => $jog,
                'tema' => tema::find(User::find(Auth::id())->tema_id->tema_nev,
                'jovo' => ora::selectraw('tantargyak.tant_nev, osztalyok.oszt_nev, orak.ora_terem, orak.ora_datum, orak.ora_szam')
                ->join('tanitott','tanitott.tanit_id','orak.tanit_id')->join('tantargyak','tantargyak.tant_id','tanitott.tant_id')
                ->join('osztalyok','osztalyok.oszt_id','orak.oszt_id')->join('tanarok','tanarok.tanar_id','tanitott.tanar_id')
                ->where('tanarok.tanar_id','=', tanar::where('fel_id','=', User::find(Auth::id())->fel_id->first())->tanar_id)
                ->where('orak.ora_datum','>=',date('Y-m-d'))->orderBy('ora_datum')->orderBy('ora_szam')->limit(5)->get(),
            ]));
        }
    } else {
        return redirect('/belepes');
    }
}
```

2. ábra Foooldal algoritmus

Ez az algoritmus fut le egy profil módosításakor, amennyiben valaki be van jelentkezve. Itt kerülnek mentésre az adatok az adatbázisba. Csak tanárfölötti joggal rendelkező személy írhat át adatot. Itt vannak leírva a lehetséges hibaüzenetek.

```

public function ProfilPost(Request $request, $id){
    if (Auth::check()){
        if (Auth::user()->jog_id > 2) {
            if (User::find($id)->jog_id == 1) {
                $request->validate([
                    'dnev' => 'required',
                    'dszulido' => 'required',
                    'dszulhely' => 'required',
                    'danyja' => 'required'
                ],[
                    'dnev.required' => 'Név megadása kötelező!',
                    'dszulido.required' => 'Születési dátum megadása kötelező!',
                    'dszulhely.required' => 'Születési hely megadása kötelező!',
                    'danyja.required' => 'Név megadása kötelező!',
                ]);
                $data = Diak::where('fel_id','=',$id)->get()->first();
                $data->diak_nev = $request->dnev;
                $data->diak_szuldatum = $request->dszulido;
                $data->diak_szulhely = $request->dszulhely;
                $data->diak_anya = $request->danyja;
                if (!is_null($request->dakcim)) {
                    $data->diak_lakcim = $request->dakcim;
                } else {
                    $data->diak_lakcim = null;
                }
                $data->save();
            }
            $request->validate([
                'fnev' => 'required',
                'femail' => 'required'
            ],[
                'fnev.required' => 'Felhasználónév megadása kötelező!',
                'femail.required' => 'E-mail cím megadása kötelező!'
            ]);
            $data = User::find($id);
            $data->fel_nev = $request->fnev;
            $data->fel_email = $request->femail;
            if (!is_null($request->ftel)) {
                $data->fel_telszam = $request->ftel;
            } else {
                $data->fel_telszam = null;
            }
            $data->save();
            return redirect('/profil/'.$id);
        } else {
            if(isset($request->dakcim)){
                $data = Diak::where('fel_id','=',$id)->get()->first();
                if (!is_null($request->dakcim)) {
                    $data->diak_lakcim = $request->dakcim;
                } else {
                    $data->diak_lakcim = null;
                }
                $data->save();
            }
            $request->validate([
                'femail' => 'required'
            ],[
                'femail.required' => 'E-mail cím megadása kötelező!'
            ]);
            $data = User::find(Auth::id());
            $data->fel_email = $request->femail;
            if (!is_null($request->ftel)) {
                $data->fel_telszam = $request->ftel;
            } else {
                $data->fel_telszam = null;
            }
            $data->save();
            return redirect('/profil');
        }
    } else {
        return redirect('/belep');
    }
}

```

3. ábra Profil módosítás algoritmus

```

public function Visszaallit($id){
    if (Auth::check()){
        if (Auth::user()->jog_id > 3){
            return view('profil',[
                'user' => Auth::user(),
                'mod' => User::find($id),
                'jog' => Auth::user()->jog_id,
                'tema' => tema::find(Auth::user()->tema_id)->tema_nev
            ]);
        } else {
            return redirect('/')->withErrors(['msg' => 'Nem engedélyezett művelet!']);
        }
    } else {
        return redirect('/belepes');
    }
}

public function VisszaallitPost(Request $request, $id){
    if (Auth::check()){
        if (Auth::user()->jog_id > 3){
            $data = User::find($id);
            $data->fel_jelszo = hash::make('RKT-'. $data->fel_nev.'-123');
            $data->save();
            return redirect('/profil/'. $id);
        } else {
            return redirect('/')->withErrors(['msg' => 'Nem engedélyezett művelet!']);
        }
    } else {
        return redirect('/belepes');
    }
}

```

4. ábra Visszaállítás algoritmus


Ez csak az admin jogosultságú fiókoknál futhat le. Ez a metódus meg kap egy felhasználói azonosítót, aszerint visszatudja állítani az illető jelszavát.

6. Tesztelés

A tesztelés folyamatosan történt, kód írása közben. Több számos hiba is előfordult, de a legtöbb az órarendnél volt. Az órákat eleinte nem tudtuk, hogyan kéne, illetve érdemes kiírni egy táblázatba. Több módszerrel is próbálkoztunk, ami legtöbbször kudarcba is fulladt és vicces kimenetelt eredményezett. Igyekeztünk sokszor tesztelni, de minél nagyobb lett a program, annál többször volt rá szükség és vissza kellett térni a működő komponensekhez is. Bár azokat a hibákat kijavítottuk, amikről tudtunk, de egy hiba javítása több hibát eredményezhet és sosincs hibamentes program.

3. Felhasználói dokumentáció, használati kézikönyv

1. Belépés



The image shows a login form with a purple header bar containing the title 'Belépés'. Below the header, there are two input fields: the first is labeled 'Felhasználónév:' and the second is labeled 'Jelszó:'. Below these fields is a purple button labeled 'Belépés'. At the bottom of the form, there is a blue link labeled 'ELFELEJTETT JELSZÓ'.

5. ábra Belépés oldal

Ez az oldal mindenkinél ugyan az. Akár a weboldal, vagy bármelyik két applikáció megnyitásakor ez fogad bennünket. Egy belépési felület, ahonnan csak az tud továbblépni, akinek van valamilyen jogosultságú fiókja regisztrálva. Két beviteli mezőt látunk:

- Felhasználónév: csak a diák jogosultságnál változó
 - diák: a diákigazolványszám
 - alapértelmezetten: az adott személy felvételekor megadott felhasználónév
- Jelszó: szintén a diákoknak más, felhasználó létrehozásnál generál egy jelszót, amit ajánlott meg változtatni a későbbi használat során biztonsági okokból
 - diák: „RKT-diákigazolványszám-123”
 - alapértelmezetten: „RKT-felhasználónév-123”

A belépés gombra kattintva, a helyes adatok megadása esetén a főoldalra jutunk, ellenkező esetben hiba üzenetet ír ki:

Elfelejtett jelszó esetén eligazít és leírja az ilyen esetben végrehajtandó lépéseket.

Belépés

Helytelen felhasználónév vagy jelszó!

Felhasználónév:

Jelszó:

Belépés

[ELFELEJTETT JELSZÓ](#)

6. ábra Első példa a sikertelen belépésre

Belépés

Felhasználónév:

Üresen hagyta a felhasználónév mezőt!

Jelszó:

Üresen hagyta a jelszó mezőt!

Belépés

[ELFELEJTETT JELSZÓ](#)

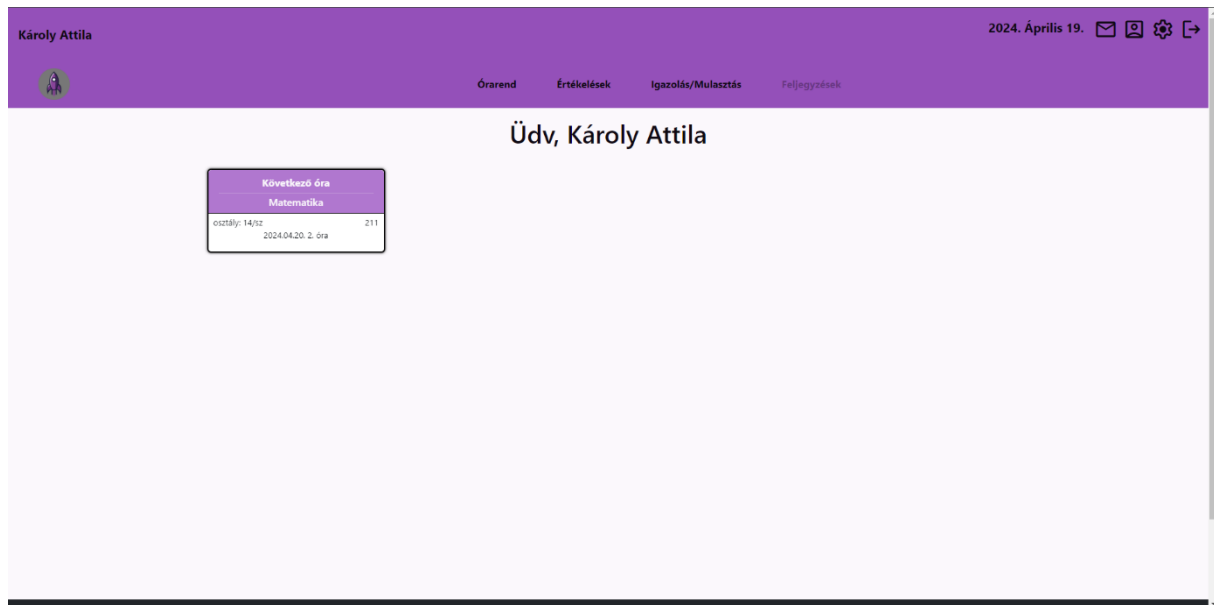
7. ábra Második példa a sikertelen belépésre

2. Általános funkciók

A belépés után, a főoldallal találjuk szembe magunkat, ahol a legutóbbi hiányzások és a jövőbeli órák jelennek meg, jogtól függően. Továbbá található egy navigációs fejléc, ahol láthatók a következők:

- felhasználó neve
- az aktuális dátum
- profil ikon, ahol a felhasználói adatokat lehet megtekinteni, akár módosítani
- beállítások ikon, ahol jelszót és témát lehet állítani

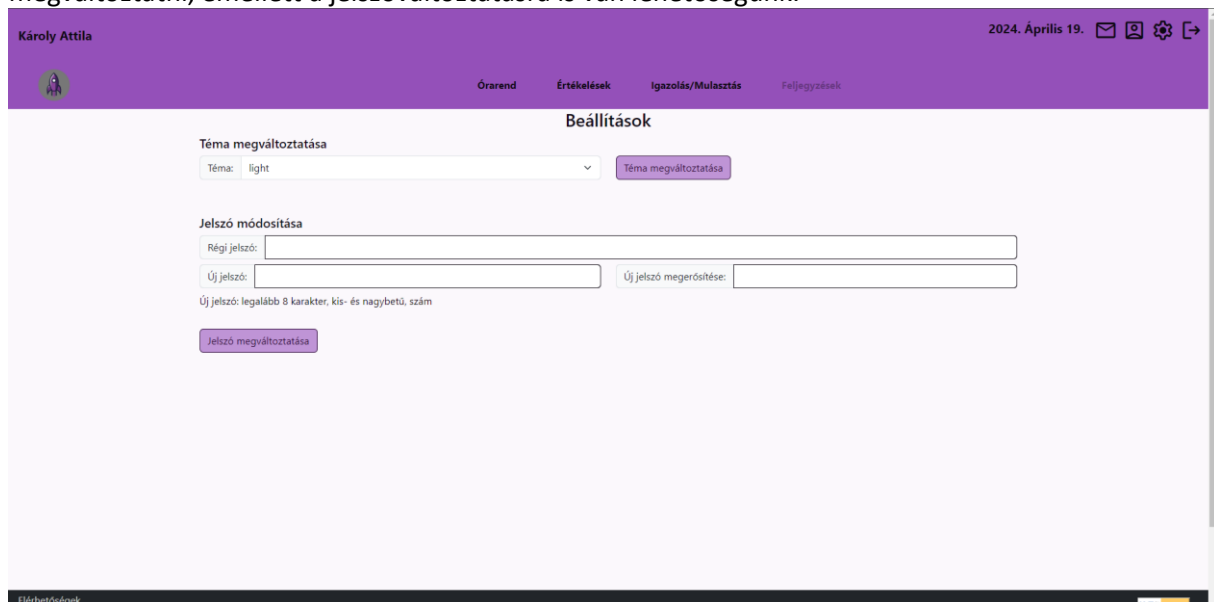
- kilépés ikon, ami kilépteti a felhasználót
- órarend
- értékelések
- hiányzások



36. ábra Főoldal

A felhasználónévre kattintva a főoldalra jutunk vissza.

Beállítás ikonra kattintva a beállítások oldalon találjuk magunkat, ahol a weboldal témáját tudjuk megváltoztatni, emellett a jelszóváltoztatásra is van lehetőségünk.



37. ábra Beállítások oldal

A jelszó megváltoztatásához a régi jelszót is meg kell adni, illetve vannak megkötések az új jelszóra, melyek az új jelszó mező alatt is fel vannak tüntetve, ezek a következők:

- minimum 8 karakter hosszúságúnak kell lennie

- számot kell tartalmaznia
- kis- és nagybetűket kell tartalmaznia

A z új jelszót a „jelszó megváltoztatása” gombbal lehet véglegesíteni, ha a megkötéseknek megfelelően jártunk el. Ellenkező esetben hibaüzenetet ír ki.

38. ábra Hibaüzenet a rosszul megadott új jelszó esetén

A téma beállítást egy legördülő lista segítségével tudjuk kiválasztani majd a „téma megváltoztatása” gombbal véglegesíteni.

39. ábra Téma módosítás

Adat	Régi	Módosítás
Felhasználónév:	karoly.attila	
E-mail cím:	karoly.attila@citromail.com	karoly.attila@citromail.com
Telefonszám:	nincs megadva	
Jog:	tanár	
Téma:	light	Módosítás

40. ábra Profil oldal

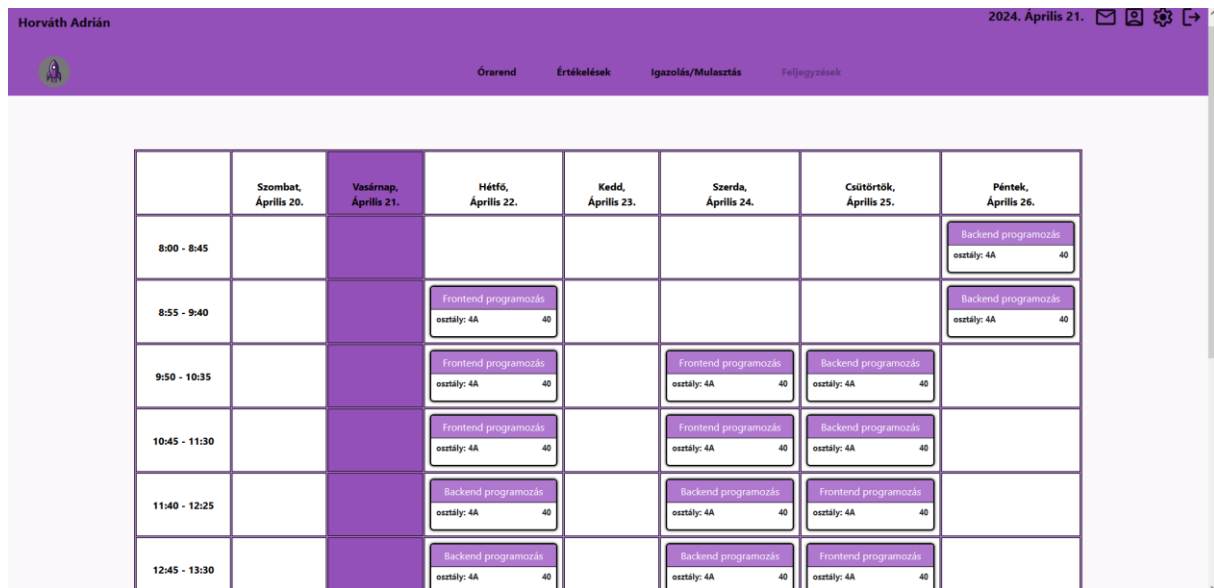
A profil ikonra kattintva a profil oldalra jutunk el.

Itt a személyes adatokat tudjuk szerkeszteni, melyek jogtól függően változnak. A „módosítás” gombra kattintva véglegesíthetjük a változtatott adatokat.

Kilépés ikonra kattintva az oldal kijelentkeztet és a belépés oldalon találjuk magunkat

3. Diák funkciók

A diák az alapfelhasználó, saját adatain kívül mást nem tud módosítani, csak megtekinteni a számára generált információkat. Az „Órarend” menü opcióra kattintva az órarend oldalra navigál bennünket a

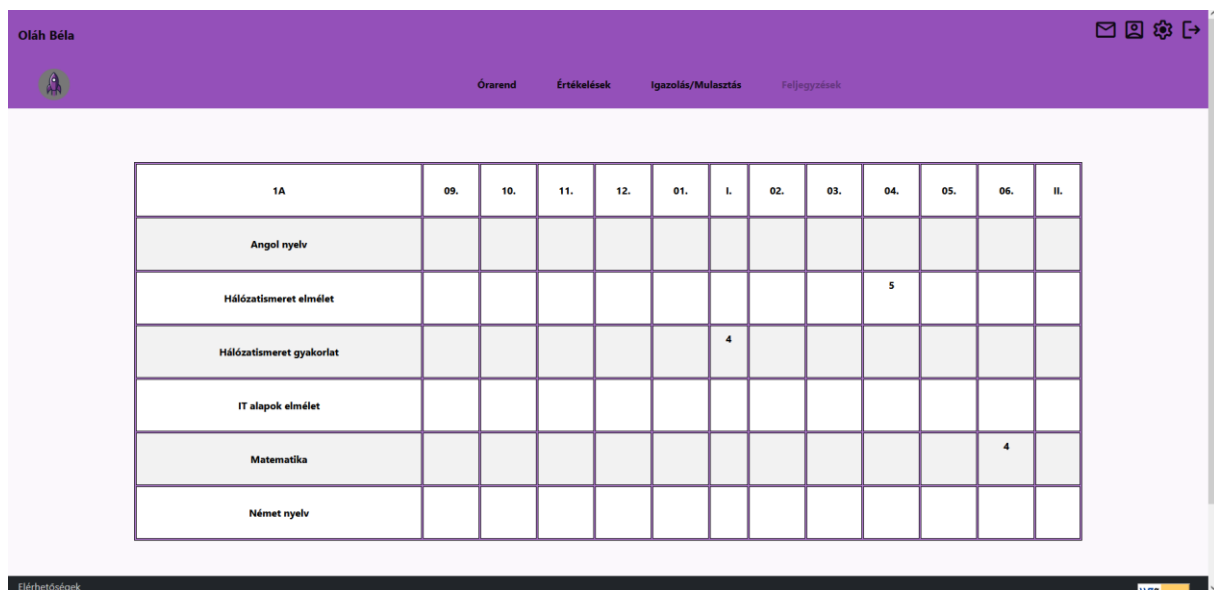


	Szombat, Április 20.	Vasárnap, Április 21.	Hétfő, Április 22.	Kedd, Április 23.	Szerda, Április 24.	Csütörtök, Április 25.	Péntek, Április 26.
8:00 - 8:45							Backend programozás osztály: 4A 40
8:55 - 9:40			Frontend programozás osztály: 4A 40				Backend programozás osztály: 4A 40
9:50 - 10:35			Frontend programozás osztály: 4A 40		Frontend programozás osztály: 4A 40	Backend programozás osztály: 4A 40	
10:45 - 11:30			Frontend programozás osztály: 4A 40		Frontend programozás osztály: 4A 40	Backend programozás osztály: 4A 40	
11:40 - 12:25			Backend programozás osztály: 4A 40		Backend programozás osztály: 4A 40	Frontend programozás osztály: 4A 40	
12:45 - 13:30			Backend programozás osztály: 4A 40		Backend programozás osztály: 4A 40	Frontend programozás osztály: 4A 40	

13. ábra Órarend

napló. Itt az adott diák órarendjét találhatjuk meg. Táblázatban szerepel az osztály, a napok, az órák kezdési és befejezési ideje. A táblázatban megjelenik nem csak a tantárgy, de az is, hogy mely tanár tartja azt az adott órát, illetve a termet, ahol a tanóra helyet foglal.

„Értékelések” menüpontra kattintva az értékelések oldalra jutunk. Itt egy táblázatot láthatunk, aminek oszlopai hónapokból, félévi és az év végi oszlopokból állnak. A táblázat sorait a tantárgyak képzik.



	09.	10.	11.	12.	01.	I.	02.	03.	04.	05.	06.	II.
Angol nyelv												
Hálózatismeret elmélet									5			
Hálózatismeret gyakorlat						4						
IT alapok elmélet												
Matematika											4	
Német nyelv												

14. ábra Értékelések

Valamely jegyre kattintva részletek jelennek meg a táblázat fölötti dobozban, többek között:

- tantárgy megnevezése

- maga a jegy
- a jegy súlya (százalékban)
- a jegy típusa (lehet évközi, évvégi, félévi)
- a jegy témája (órai munka, házi feladat, dolgozat, stb...)
- fűzhető leírás a jegyhez
- a jegy rögzítésének időpontja az adott óránál

1A	09.	10.	11.	12.	01.	I.	02.	03.	04.	05.	06.	II.
Angol nyelv												
Hálózatismeret elmélet									5			
Hálózatismeret gyakorlat						4						
IT alapok elmélet												

15. ábra Értékelések és egy jegy részletesebb leírása

Az „Igazolás/Mulasztás” menü opcióra kattintva az igazolások oldalra jutunk el.

Az oldalon egy táblázat fogad bennünket, aminek sorait az órák építik fel, míg az oszlopok a mulasztásról az információkat.

dátum	óra	hiányzás/késés	igazolva
2024.06.11. 7. óra	Koax Ábel - Matematika	késés: 32 perc	nincs
2024.04.29. 3. óra	Koax Ábel - Hálózatismeret gyakorlat	hiányzás	nincs
2024.04.22. 4. óra	Koax Ábel - Hálózatismeret gyakorlat	hiányzás	nincs
2024.04.22. 1. óra	Koax Ábel - Hálózatismeret elmélet	hiányzás	nincs

16. ábra Igazolások/mulasztások

4. Tanár funkciók

A tanár jog a diák jogtól egy fokkal komplexebb.

Az „órarend” menü opció alatti oldalon a tanár a saját órarendjét láthatja.

Az „értékelések” menü pontra kattintva, osztályok közül tudunk válogatni, ha az adott tanárnak van/volt/lesz órája valamely osztállyal, ha nincs ilyen akkor kiír egy üzenetet, hogy: nincs óra, ahova értékelést lehetne föl venni.

17. ábra Értékelések

Ha kiválasztottunk egy osztályt, akkor egy új űrlap jelenik meg előttünk.

Az első legördülő listából az órát tudjuk kiválasztani, ahová a jegyet szeretnénk megadni. A következő listából az értékelés típusát. Az utolsó listából az értékelés témáját adhatjuk meg. Az ezt követő mezőbe leírást kell fűznünk hozzá az osztályzathoz. Az utolsó beviteli mezőbe a jegy „erősségét” adhatjuk meg százalékosan. Ezt egy táblázat követi, amiben az osztályba tartozó diákok vannak felsorolva. A jegy oszlopban található jegyek közül lehet válogatni. Egyszerre több tanulónak lehet jegyet rögzíteni.

18. ábra Értékelések megadása

Az „igazolás/mulasztás” menü opcióra kattintva egy, az értékelések oldalhoz hasonló élmény fogad minket. Itt is egy osztályt kell először kiválasztani, ami csak akkor jelenik meg, ha volt már valaha órája az adott tanárnak egy osztállyal.

19. ábra Hiányzások

Az osztály kiválasztása után egy másik beviteli űrlap nyílik meg. Ebben a táblázatban tudjuk rögzíteni diákonként a jelenlétet. A „hiányzás” gombra kattintunk egy adott diák sorában, akkor hiányzásként fogja értelmezni a napló, de ha értéket is adunk a mellette lévő kitölthető mezőbe, akkor azt késésként fogja értelmezni, tehát a késéshez is kell a hiányzás gomb. Mulasztást csak Intézményvezető, vagy a fölötti joggal rendelkezők tudják igazolni.

4A	Név	Hiányzás	Késés
	Chat Elek	hiányzik	0
	Csák Ányos	hiányzik	0

20. ábra Hiányzások, részletesebb leírása

5. Intézményvezető/Titkár funkciók

Intézményvezető-/Titkárként az adatok felvételére nyílik lehetőségünk. A menüsorban megjelenik egy új menüpont, a „felvétel” opció. Az adatok felvitele oldalra térve több opció közül válogathatunk:

- osztály felvétele

Itt osztályokat vihetünk fel az adatbázisba, azzal a megkötéssel, hogy legyen egy szabad tanár, akit be tudunk állítani osztályfőnöknek.

21. ábra Osztály felvétele

- tanuló felvétele

Itt a tanulókat és adataikat rögzíthetjük

22. ábra Tanuló felvétele

- tanár felvétele

Itt a tanárokat rögzíthetjük

23. ábra Tanár felvétele

- tanár hozzárendelése tantárgyhoz

A tanárokhöz tudunk hozzárendelni/kapcsolni tantárgyakat ezen az űrlapon, csak akkor ha van tantárgy.

24. ábra Tantárgy tanárhoz való hozzárendelése

- órarend felvétele

Az órákat egyesével, óránként lehet fölvenni, illetve itt kell a terem is megadni, megkötése, hogy legyen tanár, akihez van tantárgy hozzákapcsolva.

241. ábra Órarend felvétele

- tantárgy felvétele

Itt tantárgyakat vehetünk fel

26. ábra Tantárgy felvétele

Az adatokat a mentés gombbal érvényesíthetjük.

Intézményvezetőként megnyílik a lehetőség, hogy fiókokra keressünk rá, ezt a fiók ikonnál érhetjük el. Akár felhasználónév, vagy email cím alapján is kereshetünk a személyre. Ha kiválasztjuk az „E-mail cím” opciót és a keresőmezőbe csak egy „@” jelet írunk és így kattintunk a keresés gombra, akkor mindenki ki lesz listázva. Az intézményvezető

27. ábra Fiók(ok) keresése gomb a fiók oldalon

diákokon kívül, más jogosultságú fiók(ok)nak nem tudja megváltoztatni az adatait. A „Módosítások végrehajtása gomb megnyomásával véglegesítjük a változtatásokat.”

28. ábra Fiók(ok) keresése

29. ábra Példa a profilkeresésre

6. Adminisztrátor funkciók

Az adminisztrátor kicsivel többet tud, mint az intézményvezető. Mint „admin” lehetőségünk nyílik a feljebb említett fiókok keresésénél az bármely fiók, bármely adatait módosítani. A másik nagy különbség, hogy bármely fióknak alaphelyzetbe tudjuk állítani a jelszavát egyetlen egy kattintással.

Fiók keresése

Bombo Netti profilja

Adat	Régi	Módosítás
Oktatási azonosító:	73012345678	
Tanuló neve:	Bombo Netti	<input type="text" value="Bombo Netti"/>
Születési idő:	2003-02-23	<input type="text" value="2003. 02. 23."/>
Születési hely:	Miskolc	<input type="text" value="Miskolc"/>
Anyja neve:	Romolhatatlan Rozália	<input type="text" value="Romolhatatlan Rozália"/>
Lakcím:		<input type="text"/>
Osztály:	2A	<input type="text" value="2A"/> ▼
Felhasználó id:	8	
Felhasználónév:	73012345678	<input type="text" value="73012345678"/>
E-mail cím:	bombo.netti@raketa.hu	<input type="text" value="bombo.netti@raketa.hu"/>
Telefonszám:	nincs megadva	<input type="text"/>
Jog:	diák	<input type="text" value="diák"/> ▼
Téma:	light	

Módosítások végrehajtása Jelszó visszaállítása

30. ábra Profil adatok módosítása Adminként

7. Asztali alkalmazás

Az asztali alkalmazásba csak intézményvezetőként, vagy adminisztrátorként lehet belépni. Az alkalmazásban csak adatokat lehet felvinni.

4. Továbbifejlesztési lehetőségek

A szoftver fejlesztése során több olyan ötlet merült fel, aminek az implementálásához nem volt elég idő, ezeket ebben a listában foglaltuk össze.

- **Központosítás:** Jelenleg a szoftver csak egy intézmény kiszolgálására alkalmas, tehát minden iskolának saját adatbázis szervert kell futtatnia, saját adatokkal. Bár ez nem gond az adminisztrációs oldalról, de nem mindegyik iskola rendelkezik megfelelő eszközökkel/kapacitással, hogy egy szervert tudjanak üzemeltetni minden nap minden órájában.
- **Feljegyzések használata:** A diákoknak jegyeken kívül lehetne feljegyzéseket is rögzíteni, akár beírás, feljegyzés, vagy egy szimpla tájékoztató formájában.

- **Üzenet küldése:** Üzenetek funkció nincs implementálva, ami megkönnyítené a kommunikációt az egyének között, a felhasználói jogoktól függetlenül és egy külön felületet adna az iskolai témák megbeszélésére.
- **Házi feladatok:** Házi feladatok felvétele órától, dátumtól, lejárat dátumtól, tanártól, tantárgytól függően.
- **Csoport bontás:** Az osztályokat lehetne különböző tantárgyaknál (ha szükséges) csoportokra bontani és mindegyik csoportot más tanár tanítana, kezelne a digitális naplóban, az órarendben is máshogy jelennének meg.
- **Külön gondviselő jog/fiók:** Külön gondviselő fiók könnyebbé tenné a tanuló felügyeletét, illetve üzenet küldésnél a gondviselő(k)nek szánt információkat nem a „diákon keresztül” kell megtudniuk.
- **Különálló osztályfőnök jog/fiók:** Az osztályfőnökök számára a saját osztályuk felügyelése, kezelése az adminisztrációs folyamatokat felgyorsítaná és leegyszerűsítene, mint például a mulasztások kezelése.
- **Rendszer üzenet:** Szerver karbantartás, vagy bármilyen rendszergazdától a felhasználók felé irányuló üzenetet a rendszer üzeneteknél lehetne csoportosítani.

5. Összegzés

A digitális napló készítése alatt sok tapasztalatot szereztünk PHP² és a Laravel³ területén, bár egyikünk dolgozott leginkább backend⁴-en. A vizsgaremek létrehozása megmutatta a csapat munka, kommunikáció és annak fontosságát, hisz nem mindenki rendelkezik ugyan azzal a tudással más-más téren és más tagok perspektívát tudnak hozni a munkához, ami a dobozon kívüli gondolkodáshoz⁵ vezet, ez által kreatívabbak tudunk lenni és más megoldásokkal oldhatjuk meg a problémákat.

Megjegyzendő, hogy az időbeosztásunk nem volt a legjobb, sok mindent maradt az utolsó hetekre, ami néha nem az elvárt, de mégis működő eredményhez vezetett.

² <https://www.php.net/>

³ <https://laravel.com/>

⁴ https://en.wikipedia.org/wiki/Frontend_and_backend#Backend_focused

⁵ https://en.wikipedia.org/wiki/Thinking_outside_the_box

6. Ábrajegyzék

1. ábra EER modell az adatbázisról.....	9
2. ábra Belépés oldal	13
3. ábra Második példa a sikertelen belépésre	14
4. ábra Első példa a sikertelen belépésre	14
5. ábra Főoldal	15
6. ábra Beállítások oldal	15
7. ábra Hibaüzenet a rosszul megadott új jelszó esetén	16
8. ábra Téma módosítás	16
9. ábra Profil oldal	16
10. ábra Órarend	17
11. ábra Értékelések	17
12. ábra Értékelések és egy jegy részletesebb leírása.....	18
13. ábra Igazolások/mulasztások	18
14. ábra Értékelések	19
15. ábra Értékelések megadása.....	19
16. ábra Hiányzások.....	19
17. ábra Hiányzások, részletesebb leírása.....	20
18. ábra Osztály felvétele	20
19. ábra Tanuló felvétele.....	21
20. ábra Tanár felvétele	21
21. ábra Tantárgy tanárhoz való hozzárendelése	21
22. ábra Órarend felvétele	22
23. ábra Tantárgy felvétele	22
24. ábra Fiók(ok) keresése gomb a fiók oldalon.....	23
25. ábra Fiók(ok) keresése.....	23
26. ábra Példa a profilkeresésre.....	23
27. ábra Profil adatok módosítása Adminként.....	24