Veszprémi Szakképzési Centrum Ipari Technikum

**Szakképzés neve: Szoftverfejlesztő**

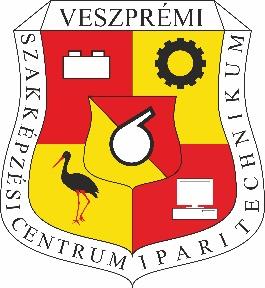
**OKJ száma: 54 213 05**

SZAKDOLGOZAT

SzámTech

Nádler Balázs Sömjén Máté

témavezető 13.C



Veszprém, 2022/2023

Tartalomjegyzék

**Bevezető 3**

Program kategóriája, célja 3

**Felhasználói dokumentáció 3**

A program általános specifikációja 3

Rendszerkövetelmények 4

Ajánlott követelmények 4

Minimum követelmények 4

A program telepítése 5

A program használatának részletes leírása 6

**Fejlesztői dokumentáció 10**

Témaválasztás indoklása 10

Alkalmazott fejlesztői eszközök 11

Adatmodell leírása 12

Részletes feladatspecifikáció, algoritmusok 14

Forráskód 17

Tesztelési dokumentáció 20

Továbbfejlesztési lehetőségek 23

Irodalomjegyzék, forrásmegjelölés 23

**Összegzés 24**

## Bevezető

## Program kategóriája, célja

## A program egy online felületű eredményrögzítő-követő alkalmazást foglal magában.

## Kategória: oktatás, haladásfigyelő rendszer valós időben

## Cél: A program célja, hogy segítse a tanulókat, szülőket és legfőképpen az őket tanító tanárokat az egyetemi kreditekhez szükséges eredmény elérésének rögzítésében, illetve nyomon követésében. Az alkalmazás nagy előnye, hogy az aktuális állást is megjeleníti, így se a tanulóknak, se oktatóiknak nem kell számolgatniuk az átlagokat. Mindez nemcsak időt, energiát takarít meg, hanem értékes papírt is, sőt a tanulók motivációját is növelheti azáltal, hogy látják számszerűen, hogy állnak jelenleg. Fontos azonban megjegyezni, hogy az alkalmazás csak a rögzítő tanár részére kínál belépési lehetőséget, így a diákok majd az ő közreműködésével tudnak tájékozódni az eredményeik aktuális állásáról.

## Felhasználói dokumentáció

## A program általános specifikációja

A készített program egy online felületű rendszer, ahol a Veszprémi Szakképzési Centrum Ipari Technikum Szoftverfejlesztő- tesztelő szak és a Pannon Egyetem Műszaki Informatikai Kar által kötött megállapodás alapján a középiskolai tanulmányok során megszerzett értékelések rögzíthetők, illetve az alapján ellenőrizhető a megállapodásban rögzített kreditelőnyhöz szükséges szint elérése, valamint a haladásnak folyamatos nyomon követése is lehetséges.

A kezdőképernyő egy bejelentkező felület, ahol egy admin tud csak belépni. Ezt követően az oldal átirányít a tevékenységválasztóra. Itt lehet új diákokat adminisztrálni, illetve értékeléseket rögzíteni, módosítani és nyomon követni a jelenlegi állapotot, valamint a tanév végén listázni a végzett diákok eredményeit, majd ezután törölni őket az adatbázisból. A diákok rögzítésére nagy részben a tanév elején van szükség, de év közben se kizárható, hogy jön egy új diák (a kezdő dátum ekkor is az adott tanév indulásának időpontja). A rögzítéshez a név, illetve kezdési dátum (tanév.09.01. formában) megadása szükséges.

Az érdemjegyek rögzítése az osztály és a kívánt témakör kiválasztása után lehetséges. A megjelenő lista tartalmazza a szűrési feltételeknek megfelelő diákokat. Itt egy diák nevére rányomva lehet megtekinteni a kredit megszerzéséhez szükséges részeredményt, illetve értékelést felvinni. Az értékelések felvitelénél százalékos eredményt szükséges megadni. Szintén itt lehetséges az eddigi értékelés módosítása is egy külön gombra kattintva, majd, ha ez megtörtént, akkor a friss aktuális állás jelenik meg. Ezeket a műveleteket minden tanulónál egyesével lehet végrehajtani, így is elkülöníthetőek az egyének, jobban kizárható a tévedés esélye.

## Rendszerkövetelmények

A szoftver web alapú, interneten keresztül érhető el, így használatához Internet kapcsolatra szükség van. A rendszer az itt felsorolt pontoktól eltérő, kisebb erőforrású eszközökön is használható, viszont a minimum elvásárok erősen mérvadóak, mindemellett a megfelelő gyorsaság, alkalmazhatóság érdekében az ajánlott kritériumoknak javasolt eleget tenni.

Ajánlott követelmények

* Hardver követelmények – személyi számítógép
  + Processzor: Intel Core i3-3xxx vagy hasonló, illetve nagyobb teljesítményű CPU
  + 4 GB vagy annál több RAM
* Szoftver követelmények, javallatok
  + Windows 10/11 operációs rendszer
  + XAMPP (Apache + MySQL)
  + Google Chrome vagy Mozilla Firefox vagy Microsoft Edge böngészőprogram naprakész verziója
  + 1440 x 900 felbontás vagy ennél nagyobb
  + Kijelző fekvő tájolása

Minimum követelmények

* Hardver követelmények – személyi számítógép
  + Processzor: Intel Core 2 Duo E7xxx vagy hasonló teljesítményű CPU
  + 2 GB RAM
* Szoftver követelmények
  + Windows 7 operációs rendszer
  + XAMPP (Apache + MySQL)
  + Google Chrome vagy Mozilla Firefox vagy Microsoft Edge böngészőprogram

## A program telepítése

1. A webalkalmazás használatához szükség van a XAMPP webszerver-szoftvercsomag telepítésére. Ezt a következő utasítások követésével lehet elvégezni:
2. XAMPP telepítőcsomag letöltése a következő linken keresztül (Windows 10:<https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/8.0.28/xampp-windows-x64-8.0.28-0-VS16-installer.exe>)
3. Telepítő elindítása
4. Alapvető telepítéshez kövesse a telepítő által automatikusan felajánlott lépéseket
5. A szoftver telepítése kész
6. Ezután az alkalmazást tartalmazó mappát (továbbiakban: Szakdolgozat) szükséges bemásolni a megfelelő helyre: OS(C:) > xampp > htdocs. Fontos megjegyezni, hogy a mappa már ne tömörített állapotban legyen.
7. A következő lépés a XAMPP Control Panel megnyitása, majd azon belül az Apache és MySQL modul elindítása (START)
8. Az előző pontban megnyitott panelen a MySQL modul sorában nyomjon rá az „Admin” feliratú gombra, ez megnyitja a phpMyAdmin felületét a böngészőjében
9. A phpMyAdmin felületen a felső sávban nyomjon az „Importálás” funkcióra
10. Itt importálja be a Szakdolgozat > db > nyilvantartas.sql fájlt, majd alul nyomjon az „Importálás” feliratú gombra. Ezáltal az adatbázis is betöltésre került.
11. Így az alkalmazás már elérhető lesz a böngészőből. A rögzítő admin felhasználónak egyszerűen csak egy URL-t kell beírni a böngésző címsorába, hogy beléphessen a rendszerbe. A leírt lépéseket követve a megfelelő URL: localhost/Szakdolgozat. A belépés egyetlen felhasználóval és annak jelszavával lehetséges, ezzel is növelve a biztonságot (A jelenlegi felhasználó – jelszó páros: admin – admin, amit majd tényleges használat előtt célszerű módosítani). Belépést követően az alkalmazás könnyen használható a grafikus felületnek köszönhetően. Viszont fontos a stabil internetkapcsolat a használatához. Az adminisztrációt követően elengedhetetlen, hogy a felhasználó a Kijelentkezés gombra rányomva kilépjen az alkalmazásból.

## A program használatának részletes leírása

## Felhasználói felület

A készített projekt egy webalkalmazás, amely grafikus elemekből épül fel. Alapvetően négy részre különíthető el, az első egy bejelentkező felület, ahol a belépési azonosító és jelszó páros megadásával lehet továbblépni. Az alkalmazáshoz nincs szükség minden személynek hozzáférésre, elég, ha listát csak egy felhasználó (admin) vezeti. Helytelen bejelentkezési adat/adatok megadása esetén az űrlap tájékoztat a hibáról, esetlegesen annak okáról (pl. hosszúsági kritériumok). A jelszó elfelejtése esetén kiválasztható egy opció, ahol kérvényezni lehet egy új beállítását. (Jelenleg az alkalmazás tájékoztatja a felhasználót, hogy kihez fordulhat új jelszó igénylési szándékával).

A belépést követően egy választási oldalhoz érünk el, ahol háromféle lehetőség van: diákok adminisztrációja, az értékelések rögzítése és azok nyomon követése, illetve új év kezdése, az adott tanévet elvégzett tanulók eredményeinek listázása. A választott lehetőségre való kattintáskor jelenik meg a kért tevékenység.

A diákok adminisztrációja felületen a jelenlegi érvényes név és a kezdési időpont megadása szükséges. A felvitel gomb megnyomását követően az adatok rögzítésre kerülnek, illetve megjelennek a „Legutóbb felvitt adatok” nevezetű részben. Elrontott vagy téves adatbevitel esetén lehetőség van javítani, illetve törölni a rekordot a felületen azonnal. Amennyiben nem volt előzőleg felvitt adat, a javítás, illetve törlés nem lehetséges és erről tájékoztat is minket az oldal.

A második opciónál a kívánt osztály és technikumi tárgykör kiválasztása után megjelenik a szűrési feltételeknek megfelelő diákok listája, ezután itt van lehetőség a diákok elért eredményének rögzítésére tárgykörök mentén, amit százalékos eredményként kell megadni, ezután az adatok eltárolásra kerülnek. Szintén ezen az oldalon van lehetőség az eddigi minősítések módosítására, továbbá az előrehaladás százalékos megtekintésére is mind technikumi tárgykör, technikumi tárgy és egyetemi tárgy vonatkozásában. A felsorolt eredmények listája alapján is esetlegesen látható, hogy ki nem fogja tudni elérni a minimumszintet, de erről a program ad visszajelzést is. Megfelelő állás esetén zöld színnel, veszélyzónában lévőknél narancssárga színnel, a nem megfelelően teljesítőknél pedig piros színnel látható az eddig elért százalékos eredmény.

A harmadik opciónál pedig lehetőség van a tanévet elvégzett diákok eredményeinek megtekintésére, ami egy listaként jelenik meg a képernyőn. Illetve szintén itt van lehetőség új év kezdésére, ami annyit jelent, hogy az adatbázisban lévő tanulókat a következő évfolyamba helyezi, az imént végzetteket pedig törli.

## Pontos elérése illusztrálva:

A képen szöveg látható

Automatikusan generált leírás

*1. ábra - Belépés oldal*



*2. ábra - Belépés oldal (nem megfelelő kitöltés)*

A képen szöveg látható

Automatikusan generált leírás

*3. ábra - Belépés oldal (sikeres belépés)*

A képen diagram látható

Automatikusan generált leírás

*4. ábra - választási oldal*



*5. ábra - adatrögzítés oldal*

A képen szöveg látható

Automatikusan generált leírás

*6. ábra - értékelések adminisztrálása oldal*

*A képen Webhely látható

Automatikusan generált leírás*

*A képen szöveg látható

Automatikusan generált leírás7. ábra - értékelések adminisztrálása oldal (diák egyedi nézete)*

*8. ábra - értékelések adminisztrálása oldal (diák eredményének javítása)*

*A képen diagram, szöveg látható

Automatikusan generált leírásA képen szöveg látható

Automatikusan generált leírás10. ábra – Új év kezdése (végzett tanulók listázása – nincs végzős osztály)*

*9. ábra – Új év kezdése (végzett tanulók listázása)*

## Fejlesztői dokumentáció

## Témaválasztás indoklása

A Veszprémi Szakképzési Centrum Ipari Technikum és a Pannon Egyetem Műszaki Informatikai Kar által kötött okleveles technikusképzésről szóló megállapodás, mely magában foglalja a középiskolából (VSZC Ipari Technikum) az egyetemre kerülő diákok eredményeiből történő kreditelőnyt a Pannon egyetem Műszaki Informatikai Kar képzésein. Ennek követéséhez, a megadott kritériumok teljesítésének ellenőrzéséhez nagy szükség van egy online felületű rendszerre, ahova rögzíteni lehet a megszerzett érdemjegyeket, akár azokat évközben módosítani, ha javítás történt a témakörből, majd azt a háttérben futó kódok átkonvertálják az egyetemen megszerezhető kreditekhez szükséges szint aktuális százalékos állására, amit meg is lehet tekinteni a felületen, így nyomon követhető a haladás, illetve látható az is, hogyha már nem érhető el a kreditek megszerzéséhez szükséges minimum szint.

## Alkalmazott fejlesztői eszközök

## Programozási nyelv:

* + - 1. HTML5 (Hypertext Markup Language): A HTML szabványos jelölőnyelv weboldalak létrehozásához, mely egy weboldal szerkezetét, elemeit írja le. A HTML elemek megmondják a böngészőnek, hogyan jelenítse meg a tartalmat. A HTML5 a legfrissebb változata.
      2. CSS3 (Cascading Style Sheets): A weblap stílusának, kinézetének formázására. A CSS leírja, hogyan kell a HTML elemeket megjelenítenie a böngészőnek. A projektben külső stíluslapokból vezérli főként a kinézetet.
      3. Bootstrap v5.0: A weboldal reszponzivitásáért, stílusának megadására alkalmazott front-end keretrendszer. A Bootstrap segítségével gyorsan és egyszerűen lehet az oldalt reszponzívvá tenni.
      4. JavaScript ES6 (ECMAScript 2020): Az oldal futási időben történő módosítását, az események kezelését, validálást a kliens oldalon, valós idejű kommunikáció biztosítását a szerverrel, továbbá az adatok böngészőben való tárolását végzi.
      5. PHP (8.1.13): Nyílt forráskódú szerveroldali szkriptnyelv a dinamikus weblap készítéséhez.
      6. SQL (Structured Query Language): Szabványos nyelv az adatbázisokhoz való hozzáféréshez és azok kezeléséhez.

Fejlesztői környezet:

1. Visual Studio Code 1.73.1: Forráskód-szerkesztő alkalmazás, mely lehetővé teszi HTML, CSS, JavaScript, PHP stb. programozási nyelveken kód létrehozását, szerkesztését.
2. Google Chrome: webböngésző, melyben a beépített fejlesztői eszközök számos hasznos funkcióval rendelkeznek a hatékony fejlesztés és tesztelés érdekében (pl. Console – hibakeresés, log, Network – oldal betöltési ideje, hálózati kérések-válaszok, erőforrások megtekintése)

Adatbázis-kezelő rendszer az adatok nyilvántartására:

1. MYSQL: Apache webszerver
2. XAMPP: könnyen telepíthető Apache disztribúció MariaDB, PHP és Perl komponenseket tartalmazva. A legnépszerűbb PHP fejlesztői környezet
3. phpMyAdmin: MYSQL webes adminisztrációjának kezelésére szolgál. A felhasználói felületén keresztül a gyakran használt műveletek grafikusan is végrehajthatóak, de lehetőséget biztosít bármilyen SQL utasítás közvetlen végrehajtására is.

Szövegszerkesztő program: Microsoft Office Word

## Adatmodell leírása

## Az adatbázis több táblából épül fel, ezek rendre:

Egyetemi\_targy: Pannon Egyetem Műszaki Informatikai Kar egyetemi tárgyait tartalmazza, amelyekhez megszerezhető a kredit. Mezők: egy\_targy\_azon – az egyetemi tárgy azonosítója, ami egész szám típus, automatikusan növekvő (primary key); egy\_targynev – az egyetemi tárgy teljes neve, ami szöveges típus (maximum 255 karakter);

Technikumi\_targy: Veszprémi Szakképzési Centrum Ipari Technikum informatika szakon lévő tárgyak. Mezők: tech\_targy\_azon: a technikumi tárgy azonosítója, ami egész szám típus, automatikusan növekvő (primary key); tech\_targynev – a technikumi tárgy teljes neve, ami szöveges típus (maximum 255 karakter);

Technikumi\_targykorok: Veszprémi Szakképzési Centrum Ipari Technikum informatikai szakon tanított tárgyakon belüli tárgykörök adminisztrálása. Mezők: tech\_targy\_azon - a technikumi tárgy azonosítója, ami egész szám típus, automatikusan növekvő, a Technikumi\_targy táblából (foreign key); egy\_targy\_azon - az egyetemi tárgy azonosítója, ami egész szám típus, az Egyetemi\_targy táblából (foreign key); tech\_targykor\_azon – az adott tárgyon belüli tárgykör azonosítója, ami egész szám típus, automatikusan növekvő (primary key); tech\_targykornev – az adott technikumi tárgyon belüli tárgykör teljes neve, ami szöveges típus (maximum 255 karakter);

technikumi\_targykor\_ertekeles: A diákok tárgykörök mentén megszerzett értékeléseit tartalmazza. Mezők: ttk\_ertekeles\_azon – a technikumi tárgykörhöz, meghatározott diákhoz tartozó értékelés azonosítója, ami egész szám típus, automatikusan növekvő (primary key); tech\_targykor\_azon – adott tárgykör azonosítója, ami egész szám típus, automatikusan növekvő, a Technikumi\_targykorok táblából (foreign key); diak\_azon - diák egyedi azonosítója, ami egész szám típus, automatikusan növekvő, az diakok táblából (foreign key) ;ttk\_ertekeles – a diákhoz és a technikumi tárgykörhöz tartozó valós szám típusú értékelés (százalékos eredmény);

technikumi\_targy\_ertekeles: A diákok technikumi tárgyaiból megszerzett értékeléseit tartalmazza. Mezők: tt\_ertekeles\_azon – a technikumi tárgyhoz, meghatározott diákhoz tartozó értékelés azonosítója, ami egész szám típus, automatikusan növekvő (primary key); tech\_targy\_azon – adott technikumi tárgy azonosítója, ami egész szám típus, automatikusan növekvő, a Technikumi\_targy táblából (foreign key); diak\_azon - diák egyedi azonosítója, ami egész szám típus, automatikusan növekvő, a diakok táblából (foreign key) ;tt\_ertekeles – a diákhoz és az adott tárgyhoz a tárgykörökből számított valós szám típusú értékelés (százalékos eredmény);

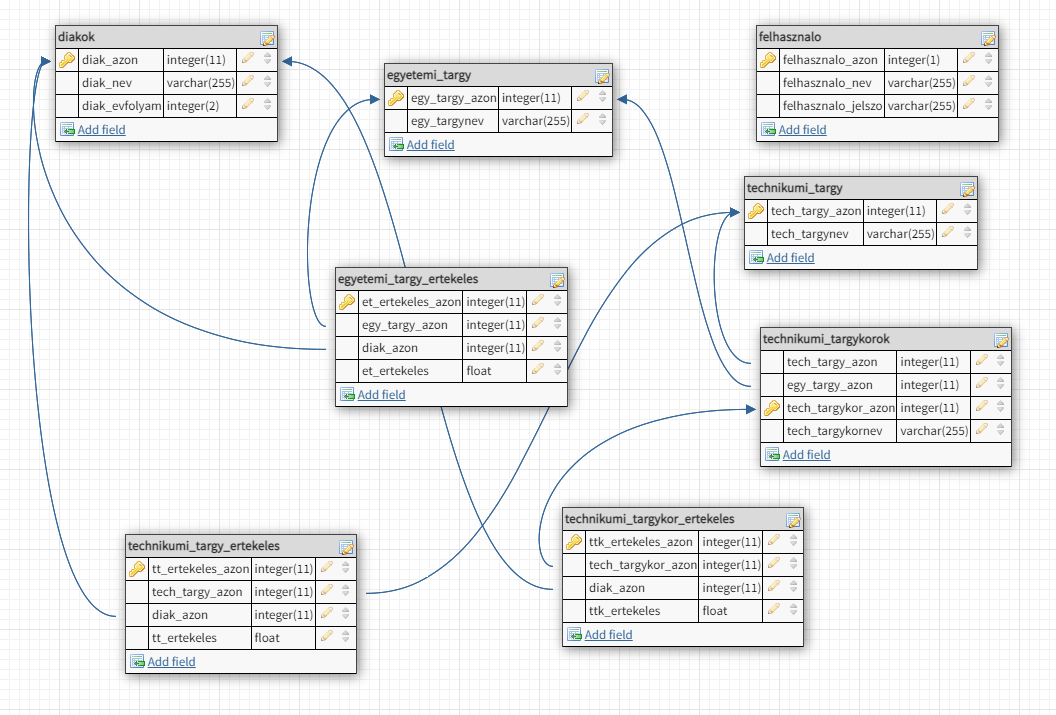
egyetemi\_targy\_ertekeles: A diákok technikumi tárgyaiból megszerzett értékeléseket az egyetemi tárgy eredményévé konvertálva tartalmazza. Mezők: et\_ertekeles\_azon – az egyetemi tárgyhoz, meghatározott diákhoz tartozó értékelés azonosítója, ami egész szám típus, automatikusan növekvő (primary key); egy\_targy\_azon – adott egyetemi tárgy azonosítója, ami egész szám típus, automatikusan növekvő, az egyetemi\_targy táblából (foreign key); diak\_azon - diák egyedi azonosítója, ami egész szám típus, automatikusan növekvő, az diakok táblából (foreign key) ;et\_ertekeles – a diákhoz és az adott egyetemi tárgyhoz a technikumi tárgyakból (azon belül a technikumi tárgykörökből) számított valós szám típusú értékelés (százalékos eredmény);

Diakok: A diákok adminisztrálására szükséges tábla. Mezők:

diak\_azon: a diák egyedi azonosítója, ami egész szám típus, automatikusan növekvő (primary key); diak\_nev – a diák teljes neve, ami szöveges típus (maximum 255 karakter); diak\_evfolyam: a diák évfolyama, ami egész szám típus, maximális hossza 2 karakter (9-től 13-ig);

felhasznalo: A bejelentkezési adatokat tárolja a későbbi azonosításhoz. Mezők:

felhasznalo\_azon – a felhasználó azonosítója, ami egész szám típus, automatikusan növekvő (primary key); felhasznalo\_nev - felhasználó teljes neve, ami szöveges típus (maximum 255 karakter); felhasznalo\_jelszo - felhasználó jelszava titkosítva, ami szöveges típus (maximum 255 karakter);

Az adatbázis grafikus modellje a következő képen látható:

*11. ábra - adatmodell*

## Részletes feladatspecifikáció, algoritmusok

Az alkalmazás működéséhez az egyik elengedhetetlen feladat a bejelentkezés megfelelő biztosítása. A kliens oldali műveleteket a következő (V. pontban) lehet megtekinteni. Itt a PHP-hez érkezett kérés kiszolgálása lesz részletezve. A kérés az index.php fájlhoz érkezik, ami a route.class.php-hez küldi az adatokat további műveletek végrehajtására. Itt megvizsgálásra kerül a kapott URL vége (ugyanis ez azonosítja az A képen szöveg látható

Automatikusan generált leíráselvégzendő műveletet), a következő ábra ezt tartalmazza:

*12. ábra – route.class.php (kódrészlet)*

A képen szöveg látható

Automatikusan generált leírásItt példányosításra kerül az osztály egyik objektuma, majd ezen hívjuk meg az egyik metódusát, ami bejelentkezes($fnev, $jelszo) névre hallgat. Ez a függvény látható a következő ábrán:

*13. ábra – nyilvantartas.class.php (kódrészlet)*

A bejelentkezes függvény paraméterként várja a beírt felhasználónév, illetve jelszó párost meghívásakor. Ezt követően ellenőrzi, hogy nem e üres egyik se, majd a jelszót hasheli a biztonságos kezelés, továbbítás érdekében (SHA256). Majd a megfelelő SQL lekérdezéssel a függvény lekéri az adatbázisban eltárolt bejelentkezési adatokat. Azt vizsgálja, hogy megtalálhatóak e a beírt adatok, ha nem, akkor a válasz az lesz, hogy a belépés sikertelen, ha igen akkor a válasz az lesz, hogy sikeres a belépés, illetve elindítja a session-t és beállítja a session tömb egyik változójának a felhasználónevet, másiknak pedig egy loggedin nevezetű segédváltozót (true/false) értékkel, aminek lényege a későbbiekben lesz kifejtve. Végül a függvény visszaadja a választ, amit majd az index.js-ben lévő Bejelentkezes() függvény kezel le.

Az előbb említett loggedin változó minden aloldalon felhasználásra kerül, annak érdekében, hogy ellenőrizzük azt, hogy a felhasználó bevan-e jelentkezve. Ez a következőképpen történik meg: Először is kérést indítunk:

*14. ábra – valaszto.js (kódrészlet)*

A képen szöveg látható

Automatikusan generált leírásAz isLoggedIn() nevezetű függvény kérést indít, és ha válaszként nem üres sztringet kap, akkor a visszakapott URL-re irányít át, ami az alapértelmezett index.html. Ez azért hasznos, mert ha nincs bejelentkezve a felhasználó, akkor a konkrét URL tudatában se fogja tudni elérni az aloldalakat. A következő ábrán az ezt megvalósító PHP kód látható:

*15. ábra – nyilvantartas.class.php (kódrészlet)*

A loggedIn() nevezetű függvény megvizsgálja, hogy a $\_SESSION tömbben van e egyáltalán ilyen néven, hogy loggedin változó és, hogy az értéke mi. Ha a loggedin értéke false, akkor már az előbb említett index.html-re mutató URL-t adja vissza válaszként, ha pedig true, akkor egy üres sztringet.

## Forráskód (részletek)

## A képen szöveg látható Automatikusan generált leírásA program több fajta kiterjesztésű fájlt is tartalmaz. A .html kiterjesztésűek a weboldalak szerkezetéért felelősek, különösebb kifejtésük nem szükséges. A szerver-kliens kapcsolatát JavaScript és PHP kódok segítségével valósítja meg, melyekből származnak a következő példák:

*16. ábra – index.js (kódrészlet)*

## Az index.js állományból származó Bejelentkezes() függvény gondoskodik a bejelentkező felületen a felhasználó (admin) ellenőrzéséről. Miután ellenőrizte, hogy minden szükséges mező kitöltésre került-e, a függvény aszinkron kéréssel kérést küld az index.php fájlnak, amely elvégzi az adatbázisban eltárolt felhasználó és jelszó páros összehasonlítását az elküldött bejelentkezési adatokkal. Amennyiben megegyezik a lekért páros a beírt adatokkal, akkor tájékoztat a sikeres belépésről, illetve továbbít a választási oldalra, vagy figyelmeztet a hiba okáról, ami lehet egyrészt azért, mert nem megfelelőek a beírt azonosítási adatok (sikertelen bejelentkezés), másrészt azért, mert valamelyik mező kitöltetlen maradt. (A bejelentkezési kérést elküldő PHP függvény az előbbi IV. pontban került kifejtésre).

*17. ábra – index.php (kódrészlet)*

## Az index.php fájlból származó adatokLekerese() függvény kezeli az index.js kérését. Csatlakozik az adatbázishoz, majd a megfelelő SQL lekérdezést (a kiválasztott ábra nem szemlélteti) futtatja, amennyiben kap adatot, akkor azt JSON formátumba alakítja és visszaadja, amennyiben nem, akkor vagy üres tömböt, vagy az adatbázis hiba üzenetét adja vissza.

## Az adatrogzites.js állomány gondoskodik a diákok rögzítéséről is. Ennek az aszinkron kérésnek egy részlete látható a következő ábrán:

*18. ábra – adatrogzites.js (kódrészlet)*

A diakRogzites() függvény először ellenőrzi, hogy a név és a kezdési dátum mezők kilettek e töltve, aztán megpróbálja végrehajtani a kérést, ahol elküldi a tanuló nevét és annak évfolyamát (amit az evfolyamSzamol() függvénnyel számol ki a kezdési dátum alapján) az index.php-nek, ami elvégzi a szükséges adatbázis műveleteket is. Illetve itt meghívásra kerül egy másik aszinkron kérés is, az idLekerese() nevezetű függvény. Ebben a felvitt rekord azonosítóját kérjük le, ami azért szükséges, mert azonnali javítás vagy törlés esetén ezzel tudjuk elvégezni a kívánt műveletet az adatbázis táblájában. A lekért azonosítót egy hidden típusú űrlapmezőben tároljuk a későbbi esetleges felhasználáshoz, de ez az érték természetesen minden új diák rögzítésénél felülíródik a legutóbbi azonosítóval.

Az index.php fájlon belül nagyon egyszerűen megtörténik a diák felvitele, mely a következő ábrán látható:

*19. ábra – index.php (kódrészlet)*

Az adatok rögzítését az adatokValtozasa() függvény végzi, mely futtatja a korábban meghatározott, a rögzítéshez szükséges SQL műveletet és annak alapján, hogy történt e változás az adatbázis táblájában visszaad egy üzenetet, mely lehet az, hogy ‘Siker’ vagy ‘Művelet végrehajtása sikertelen’ hibaüzenet.

A következő kódrészlet is fontos feladatot lát el az alkalmazásban. A tanulokLekerese() aszinkron függvénynek lényegében feladata egyszerű: a kiválasztott osztály tanulóit lekéri az adatbázis diakok táblájából. Először is megszerzi a lekéréshez szükséges adatot (kiválasztott osztály), majd megpróbálja a kérést végrehajtani egy try-catch szerkezettel. Amennyiben a lekérdezés eredménye nem üres, tehát kap vissza adatot, akkor meghívja az accordionItemHozzaad() függvényt annyiszor, amennyi rekord jött vissza a válaszban. Természetesen ezt a függvényt megfelelően paraméterezi is diák nevével, és annak azonosítójával. Az accordionItemHozzaad() függvény szerepe az, hogy létrehozza azokat a lenyíló listákat, melyeken szerepel az adott tanuló neve, majd azt lenyitva a listázott értékelések is. Mindez természetesen az index.php segítségével is lesz végrehajtva, ahol az adatokLekerese() függvény van használatba véve a megfelelően megadott sql lekérdezéssel.

A képen szöveg látható

Automatikusan generált leírás

*20. ábra – adminisztracio.js (kódrészlet)*

## Tesztelési dokumentáció

Teszteset: Érvényes felhasználónév és jelszó megadása a bejelentkezésnél, majd bejelentkezés

* Elvégzendő tesztlépések: A felhasználónév és jelszó beviteli mezők kitöltése a megfelelő belépési azonosítókkal, majd a Bejelentkezés gombra kattintás
* Kimenetek:
  + Ha valamelyik beviteli mező üresen maradt és úgy kerül megnyomásra a Bejelentkezés gomb, akkor egy tájékoztató üzenet jelenik meg „Valamelyik mező nem lett kitöltve” szöveggel, ahol az OK gombra nyomás után lehet újra próbálkozni
  + Ha mindkét mező kilett töltve, de a Bejelentkezés gombra nem lesz nyomva, akkor nem történik meg az adatok elküldése, mindenképpen szükséges a gombra kattintani
  + Várt: Ha helyesek a megadott adatok és megtörténik a gomb megnyomása, akkor egy tájékoztató üzenet jelenik meg „Sikeres belépés!” szöveggel, ahol az OK gombra nyomás után átirányít az alkalmazás a tevékenység választási oldalra.
  + Ha a megadott adatok helytelenek és megtörténik a gomb megnyomása, akkor egy tájékoztató üzenet jelenik meg „Belépés sikertelen!” szöveggel, ahol az OK gombra nyomás után lehet újra próbálkozni
* Jelenlegi eredmény: Sikeres belépés
* Teszt eredménye: sikeres

Teszteset: A diákok adminisztrációja aloldalon diák felvitele sikeres eredménnyel

* Elvégzendő tesztlépések: A teljes név és a kezdési dátum beviteli mezők kitöltése a megfelelő értékekkel, majd a Felvitel gombra kattintás
* Kimenetek:
  + Ha valamelyik beviteli mező üresen maradt és úgy kerül megnyomásra a Felvitel gomb, akkor egy tájékoztató üzenet jelenik meg „Mindkét mező kitöltése kötelező” szöveggel, ahol az OK gombra nyomás után lehet újra próbálkozni
  + Ha mindkét mező kilett töltve, de a Felvitel gombra nem lesz nyomva, akkor nem történik meg az adatok elküldése, mindenképpen szükséges a gombra kattintani
  + Várt: Ha az adatok meglettek adva, illetve megfelelő formátumban vannak (a dátum mező nem is fogad el nem helyes formátumot) és megtörténik a Felvitel gombra nyomás, akkor az adatok rögzítésre kerülnek a diákok között és megjelennek a Legutóbb felvitt adatok rész beviteli mezőiben
* Jelenlegi eredmény: megtörtént a felvitel, az adatok megjelentek a Legutóbb felvitt adatok részben
* Teszt eredménye: sikeres

Teszteset: A diákok adminisztrációja aloldalon a Legutóbb felvitt adatok részben az adatok javítása

* Elvégzendő tesztlépések: Ha nincs legutóbb felvitt adat, akkor az előző tesztnek megfelelően adatok felvitele, majd a Javítás gombra nyomással a beviteli mezők szerkeszthetővé tétele, az adatok korrigálása, majd a Javítás gomb újbóli megnyomása
* Kimenetek:
  + Ha nincs legutóbb felvitt adat és úgy kerül megnyomásra a Javítás gomb, akkor nem történik meg a beviteli mezők feloldása és egy tájékoztató üzenet jelenik meg „Nincs legutóbb felvitt adat!” szöveggel, ahol az OK gombra nyomással lehet elrejteni ezt az üzenetet, majd új adatot felvinni a megfelelő részben
  + Ha valamelyik beviteli mező üresen maradt és úgy kerül megnyomásra a Javítás gomb, akkor egy tájékoztató üzenet jelenik meg „Nincs legutóbb felvitt adat!” szöveggel, ahol az OK gombra nyomás után lehet újra próbálkozni
  + Ha mindkét mező kilett töltve, de a Javítás gombra nem lesz nyomva, akkor nem történik meg az adatok felülírása, mindenképpen szükséges a gombra kattintani
  + Várt: Ha az adatok meglettek adva, illetve megfelelő formátumban vannak (a dátum mező nem is fogad el nem helyes formátumot) és megtörténik a Javítás gombra nyomás, akkor az adatok felülírásra kerülnek a diákok között és egy tájékoztató üzenet jelenik meg „A javítás megtörtént!” szöveggel, ahol az OK gombra nyomással lehet elrejteni ezt az üzenetet. A gombra nyomás után a Legutóbb felvitt adatok részben a beviteli mezők újra csak olvashatóak lesznek.
* Jelenlegi eredmény: megtörténik a javítás, illetve újra csak olvashatóak lesznek a Legutóbb felvitt adatok beviteli mezői és egy tájékoztató ablak jelenik meg „A javítás megtörtént!” felirattal
* Teszt eredménye: sikeres

Teszteset: A diákok adminisztrációja aloldalon értékelés felvitele tárgykör mentén résznél a kívánt tanuló kiválasztása után az Értékelések gomb megnyomásával megjelenő értékelés módosítása résznél az értékelés helyes átírása esetén az alertMegjelenit(„Az értékelés módosítása megtörtént! (id:id)”) függvény lesz meghívva.

* Elvégzendő tesztlépések: A diákok adminisztrációja oldalon osztály, tárgykör kiválasztása, majd tanulók listázása, ahol a kívánt tanuló kiválasztása. Majd az Értékelések gombra nyomás, ahol megjelennek az eddigi értékelései a szűrési feltételeknek megfelelően, ezután az értékelés módosítása, ahol az input mező változására történik meg a kívánt művelet és jelenik meg az üzenet
* Kimenetek:
  + Megtörtént a megfelelő helyre navigálás, de az Értékelések gombra nyomva kiderül, hogy nincs még értékelése az adott diáknak az adott tárgykörből, így az alertMegjelenit() függvény paramétere „” lesz
  + Ha a beírt érték 0 vagy a mező üres, akkor az alertMegjelenit() függvény paramétere „A mező kitöltése kötelező egy 0%-nál nagyobb eredménnyel!” lesz
  + Ha a beírt érték 100-nál nagyobb, akkor az alertMegjelenit() függvény paramétere „100%-nál nagyobb eredmény megadása nem lehetséges” lesz
  + Várt: Ha az intervallumnak megfelelő érték kerül beírásra, akkor az alertMegjelenit() függvény paramétere „Az értékelés módosítása megtörtént! (id:id)” lesz
* Jelenlegi eredmény: alertMegjelenit(„Az értékelés módosítása megtörtént! (id:id)”)
* Teszt eredménye: sikeres

## Továbbfejlesztési lehetőségek

A készített webalkalmazás számára sokféle továbbfejlesztési lehetőség létezik, melyek közül itt kerül felsorolásra néhány ötlet:

1. Integráció más oktatási rendszerrel (például: E-Kréta): Az alkalmazás lehetővé teheti a tanárok számára, hogy a tanulók egyetlen helyen tárolják az összes adatukat, beleértve a jegyeket, a feladatokat, a jelenléti ívet és más fontos információkat, illetve itt legyen elérhető az egyetemi kreditek megszerzésének követéséhez szükséges rendszer is egyben. Akár mobilalkalmazás formájában is megvalósítható.
2. Mobil alkalmazás a diákoknak is (előnyei a különböző pontok mentén megemlítve).
3. Tanulási folyamat motiválására továbbfejlesztés: Az alkalmazás lehetővé teheti a tanulók számára, hogy elérjék a tanulási céljaikat, és követni tudják a haladásukat a tanulási folyamatban (akár tárgykörök, tárgyak mentén).
4. Automatikus értesítések és emlékeztetők: Az alkalmazás lehetővé teheti a tanárok számára, hogy értesítéseket küldjenek a tanulóiknak a jelenlegi állásról, eredményekről. A mobilalkalmazás segítségével következő számonkérések rögzítése is megoldható lenne, így értesítéseket is küldhetne az applikáció a diákoknak.
5. Elemzések és statisztikák: Az alkalmazás lehetővé teheti mind az oktatók, mind az oktatási intézmény számára, hogy elemezzék a tanulóik teljesítményét, és könnyen megértsék, hol van szükség fejlesztésre. Illetve lehetőség nyílna az oktatók teljesítményére is következtetéseket levonni a kimutatások alapján. Ezek a grafikonok, kimutatások a diákoknak szintén segíthetnének megérteni, átgondolni a jövőbeli terveiket, céljaikat, és az azok eléréséhez szükséges energiabefektetés hozzáadását.
6. Elfelejtett jelszó igénylése esetén olyan lehetőség, hogy e-mail címre új beléptetési azonosítót kérjen a felhasználó

Az itt felsorolt pontok csak néhány példák a sok továbbfejlesztési lehetőség közül, a valós lehetőségek függenek a további fejlesztők kreativitásától és az iskola, illetve az oktatási igényektől.

## Irodalomjegyzék, forrásmegjelölés

<https://tudasbazis.ekreta.hu/pages/viewpage.action?pageId=2426030>(E-kréta tudásbázis – megtekintve: 2022. december 7.)

<https://www.w3schools.com/html/html_intro.asp>(Bevezetés a HTML-be - megtekintve: 2023. február 10.)

<https://www.w3schools.com/css/css_intro.asp> (CSS Bevezetés - megtekintve: 2023. február 10.)

<https://getbootstrap.com/docs/5.0/about/overview/> (About Bootstrap v5.0 – megtekintve: 2023. február 10.)

<https://www.w3schools.com/php/php_intro.asp> (PHP Bevezetés - megtekintve: 2023. február 10.)

<https://www.w3schools.com/sql/sql_intro.asp> (SQL Bevezetés - megtekintve: 2023. február 10.)

<https://code.visualstudio.com/docs> (Documentation for Visual Studio Code – megtekintve 2023. február 10.)

<https://www.phpmyadmin.net/> (phpMyAdmin – megtekintve: 2023. február 10.)

<https://www.apachefriends.org/hu/index.html> (XAMPP Installers and Downloads – megtekintve: 2023. február 10.)

<https://icons.getbootstrap.com> (Bootstrap Icons – megtekintve: 2023. április 14.)

<https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/8.0.28/xampp-windows-x64-8.0.28-0-VS16-installer.exe> (XAMPP download – megtekintve: 2023. április 14.)

## Összegzés

Az elkészült alkalmazást megfelelően adminisztrálva a mindennapi naplózás megkönnyíthető. Segítségével a diákok számára előnyös lehetőséget kínáló megállapodás adminisztrálása egyszerűbb, ezzel is támogatva a gördülékenységet és az oktatók munkáját.