**Záródolgozat**

Tanuló neve: **Schneider Dávid**

Osztálya: **2/14.Z**

Elkészítés éve: **2020/2021**

**ZÁRÓDOLGOZAT ADATLAP**

**A záródolgozat készítőjének neve:**

Schneider Dávid

**e-mail címe:**

schneider.david116@gmail.com

**A záródolgozat témája:**

Weboldalon keresztül egy könyvtárprogram megírása. Feladat specifikáció és dizájn meghatározása.

A két fejlesztési környezet bemutatása és összehasonlítása egymással. Miben különbözik a két

rendszer. Mik az előnyei, illetve hátrányai. Melyiket éri meg, ha egyedi, szinte hibamenteset szeretnénk. Melyik mellett döntenék én és miért.

**A záródolgozat címe:**

Keretrendszer a saját fejlesztéssel szemben

**Konzulens tanár:**

Zaletnyik Péter Tibor

Kelt: Budapest, 2020. december 11.

.................................................................. .................................................................

a záródolgozat készítőjének aláírása a konzulens tanár aláírása

EREDETISÉG NYILATKOZAT

Alulírott tanuló kijelentem, hogy a záródolgozat saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült záródolgozatomban található eredményeket az intézmény egy példányban archiválhatja.

Budapest, 20

tanuló aláírása

# ZÁRÓDOLGOZATI KONZULTÁCIÓ IGAZOLÓ LAP

Alulírott Zaletnyik Péter Tibor konzulens tanár aláírásommal igazolom

Schneider Dávid nevű tanuló konzultációkon való részvételét[[1]](#footnote-1).

|  |  |  |  |
| --- | --- | --- | --- |
| **Dátum** | **Téma** | **Tanuló aláírása** | **Tanár aláírása** |
| 2020.12.15 | Témarögzítés, határidők beállítása |  |  |
| 2021.01.22 | Adatbázis rögzítése |  |  |
| 2021.02.12 | Frontend + összehasonlítás |  |  |
| 2021.03.16 | Backend megvalósítása |  |  |
| 2021.04.07 | Dokumentációs konzultáció |  |  |
|  |  |  |  |
|  |  |  |  |

Tartalomjegyzék

[1. Bevezetés 6](#_Toc69666066)

[2. Feladatspecifikáció 7](#_Toc69666067)

[3. A szoftver megtervezése 8](#_Toc69666068)

[3.1 Fejlesztői környezet 8](#_Toc69666069)

[3.2 WordPress 9](#_Toc69666070)

[3.3 XAMPP 7.3.27 12](#_Toc69666071)

[3.4 Rendszer tervek 13](#_Toc69666072)

[3.5 Grafikus tervek 14](#_Toc69666073)

[3.6 Biztonsági kérdések 14](#_Toc69666074)

[4. Adatbázis létrehozása 15](#_Toc69666075)

[4.1 Telepítés 15](#_Toc69666076)

[6. Az Adatbázis 16](#_Toc69666077)

[6.1 Lépések 16](#_Toc69666078)

[7. Felhasználói felületek 18](#_Toc69666079)

[7.1 Adminisztrátor szintek 18](#_Toc69666080)

[7.2 Felhasználói szint 18](#_Toc69666081)

[7.3 Látogatói szint 19](#_Toc69666082)

[8. Kódolás és megvalósítás 19](#_Toc69666083)

[9. Összefoglalás 21](#_Toc69666084)

[9.1 Összehasonlítás 21](#_Toc69666085)

[9.2 Feladat összegzése 22](#_Toc69666086)

[10. Irodalomjegyzék 23](#_Toc69666087)

[11. Ábrajegyzék 23](#_Toc69666088)

# Bevezetés

A weboldal megírását inspirálta a jelenlegi vírushelyzet. Nagyon sok mindent be kellett záratni a járvány miatt. Ez rengeteg kiesés sok vállalatnak is. Nem mindenki tudja vagy teheti meg, hogy online módon tovább fusson vállalata. A könyvtárak is ilyen helyzetbe kerültek.

Ekkor jött az ötlet, hogy írok egy szoftvert a könyvtáraknak ezzel is segítve őket. Végig gondoltam mi is, illetve hogyan lehetne hasznos szoftver a könyvtárak számára. Miként működhetne érintésmentesen, biztonságosan.

Elsőként elkezdtem gondolkodni azon, hogyan valósítsam meg. Mik azok a dolgok, amikre szükségem van. Miként vigyem át a könyvtárat digitális formába. Támadt egy olyan ötletem, hogy akkor már azt is bemutatom mi a különbség két webfejlesztési környezet között. A kettő miben és hogyan tér el egymástól. Mennyivel gyorsabb az egyik, mint a másik és melyiknek milyen funkciója van. Beszereztem a megfelelő programokat hozzá majd neki is álltam. Megterveztem az adatbázisát. Utána a weboldal kinézetét és működését. Milyen kódokra lesz szükségem, hogyan és hova kell alkalmaznom őket. Koncepciókat gyártottam. Miután megtaláltam és megterveztem a megfelelő dolgokat hozzá nekiláttam a feladatnak. Megírtam az adatbázist, utána weboldalon keresztül ellenőriztem, hogy működik-e és elérem-e rendesen, aztán neki is láttam a kódolásnak. Ez egy hosszabb folyamat volt, nagyon sok mindenre figyelni kellett. Elsősorban a felhasználói felület volt a fontos, Pontosabban a bejelentkezés és regisztráció. Utána következett a többi. A könyvek tárolása, rendezése. A kölcsönzés megírása, hogyan működjön. Jelezze ki, ha már az adott könyv ki van kölcsönözve és nincs több a könyvtárban. Közben a kinézetét is írtam, hogy ne csak egy fekete-fehér egyberakott dolgokat lássak, hanem szép és átlátható legyen.

Mindezek elkészítésével már csak a tesztelések és az ellenőrzésének voltak hátra.

# Feladatspecifikáció

Maga a szoftver egy weboldal a könyváraknak, mely segíti, hogy online nyitva tudjanak lenni és tovább működhessenek. Ez kivitelezve két weboldalon kétfajta fejlesztői környezettel. Az volt a célja ennek, hogy a felhasználók továbbra is tudjanak kölcsönözni könyveket, ha olvasni akarnak. Ezt minél egyszerűbb felülettel igyekeztem megoldani. Így működhetnek a könyvtárak is tovább, nem kell teljesen bezárniuk.

Funkciói:

* Kezelőfelület
* Könyvek tárolása
* Kikölcsönzési rendszer
* Adatok gyűjtése, felhasználása

# A szoftver megtervezése

## Fejlesztői környezet

Az egyik fejlesztői környezet a WordPress. Sablonok, widgetek, bővítmények használatára tökéletesen alkalmas.

Ezáltal sok beépített funkcióval rendelkezik, amely meggyorsítja a fejlesztést.

Ennek az egyik hátránya, hogy az ember keze valamilyen szinten megvan kötve. Nem annyira szabad, bizonyos funkciók elérhetők csupán, például a sablonok.

A másik saját kezűleg megírt kódok és kinézet használása volt.

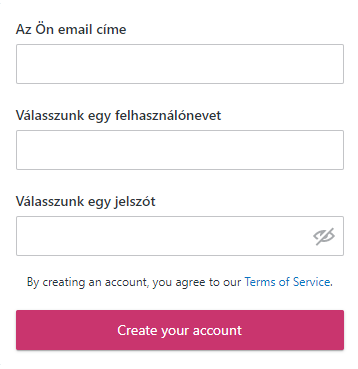
Ennél ugyan lassabb a fejlesztés, viszont szinte határtalan a kínálat. Olyanra szabjuk amilyenre csak szeretnénk.

Ez azonban több időt vesz igénybe, hisz magunk írjuk meg a kódokat, és állítjuk össze a kinézeteket.

## WordPress

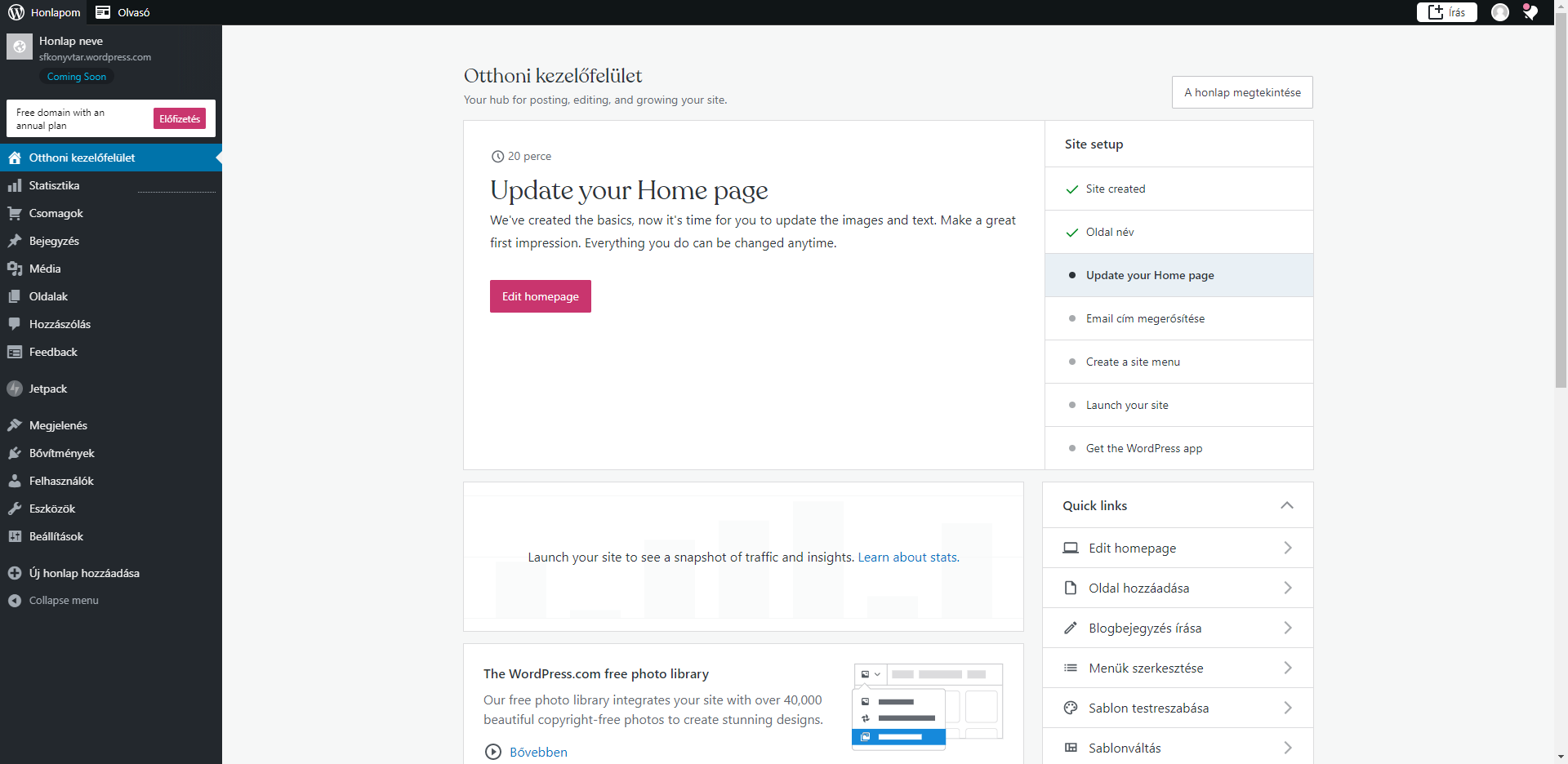
A WordPress egy nyílt forráskódú, PHP-alapú tartalomkezelő rendszer. Használhatja a MySQL adatbáziskezelőt adattárolásra. Különböző szolgáltatásokra képes, melyeket előállítva kapunk meg.

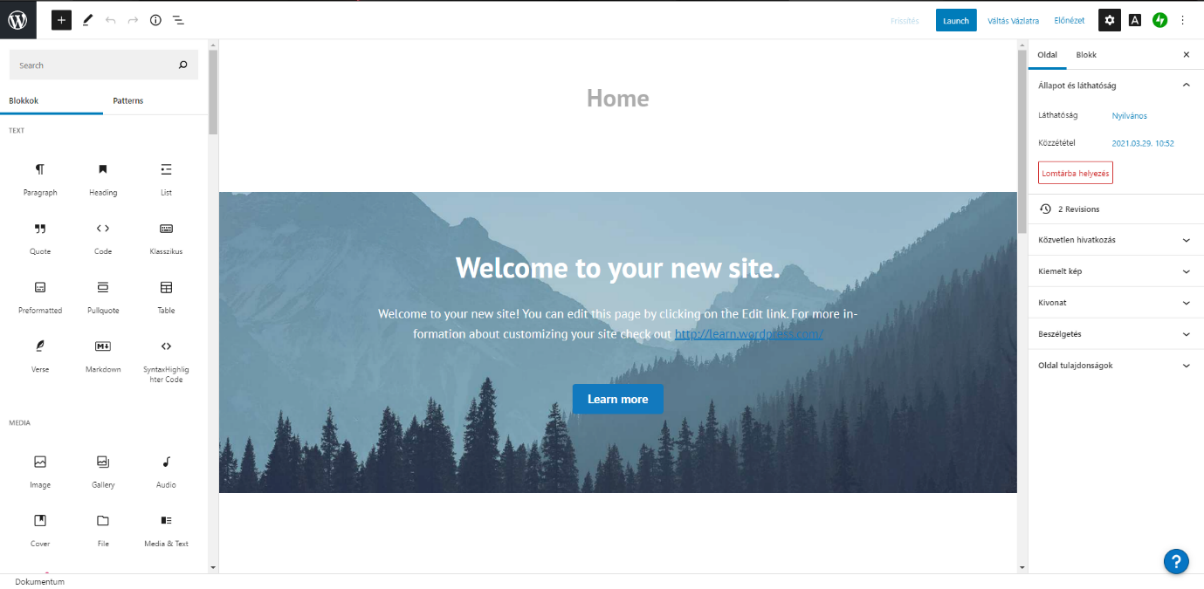
* Sablonok: Sablonokat telepíthetünk és válogathatunk oldalunkra. Ezeknek köszönhetően a kinézet úgy tud módosulni, hogy a fontosabb információk nem fognak változni az adott oldalon.
* Bővítmények: Ez az egyik legközkedveltebb tulajdonsága a WordPress-nek. Rengeteg bővítmény beépítésére van lehetőség melyeket a felhasználók könnyen továbbfejleszthetnek.
* Widgetek: Ezek olyan modulok melyek lehetőséget biztosítanak a drag and drop funkciókra.

Létrehozunk egy WordPress szolgáltatótól egy tárhelyet regisztrációval.

1. ábra

Ezután megkapjuk a webtárhelyünket hozzá, amin már el is kezdhetünk dolgozni.

2. ábra

Amennyiben rákattintunk a 2. ábrán szereplő Edithomepage gombra, már át is irányít minket a weboldalunkra, amit könnyedén szerkeszthetünk kedvünkre a megfelelő sablonokból.

3. ábra

Innentől kezdve kész is vagyunk a WordPress telepítésével. Most már lehet használni és dolgozni rajta.

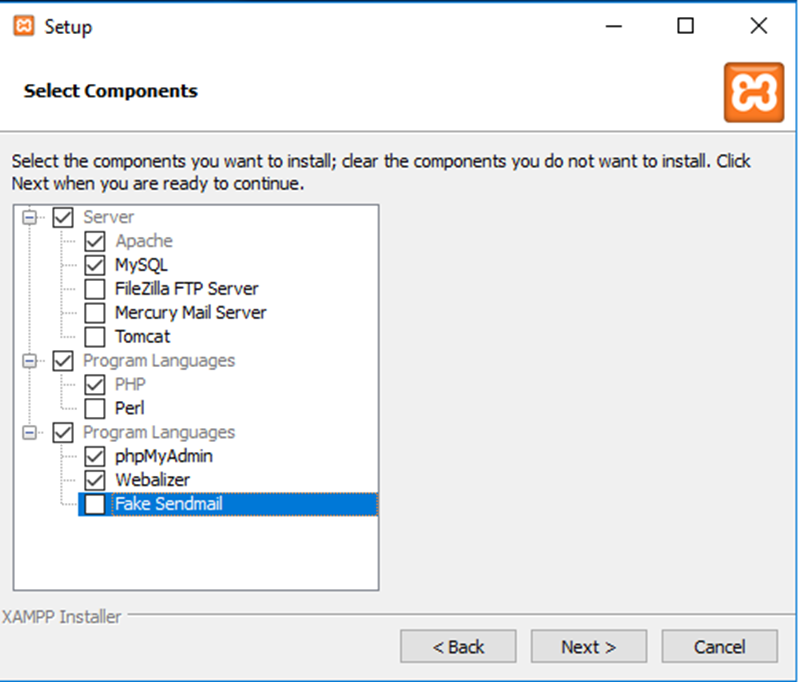
## XAMPP 7.3.27

Ez egy fontos külső eszköz egy weboldal fejlesztése és futtatása során, amennyiben lokálisan szeretnénk végleges domain nélkül. Használhatóságán felül számtalan előnyt hordoz még magában.

Egyik nagy előnye még hogy platformfüggetlen, így a lehető legtöbb operációs rendszeren hibajelenség nélkül üzemel. Pl.: Windows, Linux, Mac OS X stb.

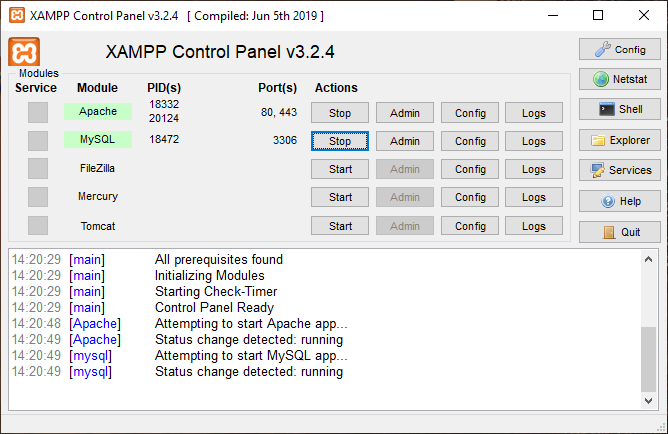
A szoftver telepítésének sikere, a lefutás környezetének megteremtésénél kezdődik, aminek minden szükséges komponense megtalálható a program gyökérkönyvtárában elhelyezett mappájában.

A telepítő futtatása után a Next gombra kattintva haladhatunk tovább.

Következő lépés, hogy az installáció alkalmával a megfelelő komponenseket válasszuk ki.

4. ábra

Ezután pár Next gomb és már telepíti is és készen is van a használatra az alkalmazásunk.

**A Finish-re kattintva pedig azonnal el is indítjuk az alkalmazást.

5. ábra

Miután elindult, elindítjuk az Apache és a MySQL funkciókat.

## Rendszer tervek

Mindkét esetben MySQL adatbázisba kerültek az adatok tárolása. A program felépítésében a PHP nyelv lehetőségei is részt vesznek. Így megvalósítható a login system is. A felülete HTML jelölőnyelven van, ami Bootstrap és CSS keretrendszereket használ. JavaScript is található benne a kényelmesebb felhasználói élményhez.

## Grafikus tervek

Figyelni kell a grafikus felület megalkotása közben arra is, hogy a felülete hogyan is mutasson. Illenie kell egy könyvtár színvilágához. A dizájnnak viszonylag egyszerűnek kell, hogy legyen. A zöld szín és annak különböző árnyalatai dominálnak és arra épül fel a weboldal. Könnyen és gyorsan átláthatónak kell lennie, hogy minél könnyebben tudja kezelni egy egyszerű felhasználó is.

## Biztonsági kérdések

Napjainkban elengedhetetlen az adatvédelem és a biztonság egy weboldal esetében. Nem különb annál, mintha a saját lakásunk biztonságáról lenne szó. A felgyorsuló világunkban a növekvő rendszereink is egyre gyorsabban fejlődnek, és a felhasználó is időről időre várja a szoftverek fejlesztését és frissítését. Az igényeiket pedig teljesíteni kell, hogy a megfelelő érdeklődési és elégedettségi szint fennmaradjon.

Ezzel egyidőben viszont a biztonsági részleg nem volt képes tartani a lépést, így a támadó könnyen hozzáférhetett az adatainkhoz, és számtalan módon kárt okozhat nekünk, ez ellen pedig védekezésre van szükségünk. A hirtelen megugrott fejlődés mellett meghatározó szerepet játszik a biztonság, és a szoftverek nyelvének és keretrendszerének attribútumai.

A legnehezebb helyzetekben a nyílt forráskódú alkalmazások esetében vagyunk, ugyanis ilyenkor bárki hozzáférhet a forráskódhoz, továbbá módosításokat is végrehajthat rajta, ha nincs megfelelő védelem az adott szoftveren.

Biztonsági szempontból különböző szempontokból lehet megközelíteni egy programot annak függvényében, hogy mi célból kell annak biztonságosnak lennie, az eltárolt adatok milyen típusú felhasználók elől legyenek elrejtve. Érdemes az egymással kommunikáló kódokat fizikailag külön szedni, ezzel is megnehezítve a támadó gyors támadását. Az adatbázis kapcsolatot használó weboldalak legtöbbje fenyegetve van a szövegdobozok miatt, mert ezeken az elemeken keresztül hozzáférhetnek különböző SQL utasításokhoz és akár módosíthatják azokat is, így elég nagy károkat téve bennük. Ilyen esetekhez érdemes védelmet biztosítani. (védelem beleírása)

Szükségünk van még az adatbázis védelme mellett a felhasználói fiókokat is levédenünk. Regisztrációkor a hasshelt jelszavak jönnek létre. Ez egy jól titkosított jelszóvédés.

# Adatbázis létrehozása

A MySQL-ben létrehozott adatbázisok kezelhetőok a phpmyadmin felületén keresztül, ami a localhost (127.0.0.1) kapcsolaton keresztül történik. Ez egy grafikus kezelőfelület, ami átlátható, könnyen kezelhető. Egyszerűen exportálható is.

## Telepítés

1. Megnyitjuk alkalmazást tartalmazó mappát. És megkeressük benne a konyvtar.sql fájlt.
2. Majd a böngészőnk címsorába gépeljük a következőt: 127.0.0.1/phpmyadmin/
3. Kiválasztjuk az importálás opciót
4. A megnyitott mappából kiválasztott \*.sql fájlt az importálás mezőbe dobjuk
5. Az adatbázis innentől kezdve használható

# Az Adatbázis

## Lépések

Az adatbázis felépítésének megvan a maga menete, és hogy az adatkapcsolatunk megfelelően működjön, kell hozzá pár lépés, amit be kell tartanunk.

Elsőként meg kell határoznunk, hogy mi is legyen az adatbázisunk célja, mire is szeretnénk az adatbázist használni. Milyen típusú adatokra lesz szükségünk, mekkora lesz az adathalmaz.

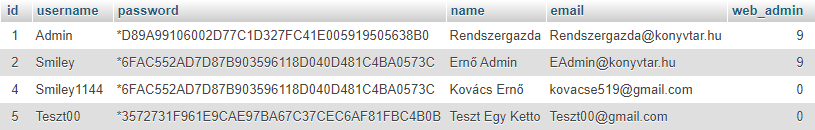
Másodszor az elkészítendő táblákat kell meghatároznunk. A program során az adatokkal, amikkel dolgoznunk kell érdemesebb valamilyen rendszerrel szétszelektálni őket és ellenőrizni azokat. Ez számunkra is nagy könnyítés a későbbiekben, amennyiben nagy adatbázisban kell dolgozni.

Harmadik pedig a táblák mezőinek megalkotása és a szerepeinek kiosztása. Típusa, adatai és méretei. Az adatai alapján azonosítjuk be, hogy hova is kerüljenek.

A mezők eléréséhez minden egyes táblának szükséges lesz létrehozni egy olyan mezőt, amiből egynek egyedi azonosítónak, azaz elsődleges kulcsnak kell lennie. Általában ID-nak szoktuk elnevezni és Auto Increment funkciót kap, ami annyit jelent, hogy automatikusan számol 1-től felfelé, így adva ID-ket minden sorhoz.

A táblák kapcsolatai elkerülhetetlen lépés. Szintén meg kell terveznünk azokat. Ezekhez a kapcsolatokhoz létre kell hoznunk olyan mezőket, amik a két táblában ugyan azok lesznek, így létrehozva velük kapcsolatot.

Adatok tárolására jelenleg három, egymással kapcsolatban álló táblából jön létre. Két táblának egyedi azonosítója van, és van egy kapcsolati tábla, ami gyakorlatilag összeköti a kettőt. Az első tábla a felhasználókat kezeli. A neveket, jelszavakat, e-maileket.

*6*. ábra

Ez a tábla az account néven található meg. A weboldal session-ben éri el ezeket és vannak tárolva leginkább. A jelszavaknál már a hashelt jelszó látható.

A következő tábla magát a könyveket tárolja. Mi a címe, kategóriája, helye és hogy elérhető-e még a kikölcsönzésre.

7. ábra

A harmadik táblánk pedig a kapcsolati tábla. Ez maga a kölcsönzés, ami összeköti a két táblát. Itt megtalálhatóak a kölcsönzések. Mikor, mennyi időre mely könyvek vanna kikölcsönözve, és ki a kölcsönző.

8. ábra

Később ez még bővíthető is. Például külön táblákba helyezzük el kategória szerint a könyveket, ami gyorsíthatja a keresést és az adatok egyszerűbb tárolását.

# Felhasználói felületek

## 7.1 Adminisztrátor szintek

Az oldalon megtalálható az Admin felület is. Ez kilenc szintre bontható. Itt megszabhatjuk, hogy az egyes szintekkel rendelkező adminok, avagy alkalmazottak milyen jogosultságokat kapjanak a weboldalhoz. Például aki a könyveket akarja hozzáadni az adatbázishoz annak elég csak 3-as jogosultság. Elleneben, aki minden kezelni akar, azaz a rendszergazda, neki van a legmagasabb evidensen. A rendszergazda tudja állítani az Admin szinteket, kinek milyen jogosultságai legyenek.

A web adminnal rendelkező felhasználóknak belépéskor a fiókpanel alatt megjelenik egy Adminisztráció nevű felület. Erre kattintva érik el azokat a dolgokat, amik a szintjeik képest tudnak használni. Természetes a sima felhasználói funkciók ugyan úgy működnek nekik is.

## 7.2 Felhasználói szint

A felhasználóknak egy azonos szintjük van. Mivel nekik nincs Admin szintjük (0) ezért nem is férnek azokhoz a funkciókhoz hozzá, amihez egy Admin. A Fiók panelben sem jelenik meg az Adminisztráció fül.

Egy felhasználó a Fiók adatait áttekintheti. Módosíthatja is azokat. Megnézheti a már saját kikölcsönzött könyveit. Megjelenik neki egy funkció, ami csak akkor elérhető, ha már az adott felhasználó bejelentkezett. Így csak az tud könyvek kikölcsönözni, aki már regisztrált sikeresen és van fiókja. Csak úgy, mint a könyvtárakban, csak itt nem kártyás megoldás van, hanem regisztrációs.

## 7.3 Látogatói szint

A nem regisztrált felhasználók csak a weboldalt tudják megtekinteni és a könyvtárról olvasni. Bár hozzáférnek még a könyvek megtekintéséhez is. Ez azért van így, mert ha megtetszik nekik egy könyv és van is raktáron, akkor csak regisztrálniuk kell és már ki is tudják kölcsönözni, mint felhasználók. Így nyomon követhető, hogy ki és hogyan kölcsönzött ki könyveket.

# Kódolás és megvalósítás

Mielőtt kitérnék a kódolásokra szeretném bemutatni a felépítését a weboldalnak. Elsősorba vannak az oldalak. Minden oldalhoz gyakorlatilag tartozik egy úgynevezett include(inc) PHP egy külön mappában. A sima oldalak tartalma az egyszerű HTML kódok. Az inc PHP fájlokban pedig a PHP kódolás megvalósítása. Ezen kívül található még egy classes mappa, amiben szintén include képpen működik, de az gyakorlatilag a fő szerkezeti kódolása. Ezek össze vannak kötve egymással. Sokkal átláthatóbb, illetve ha bármi probléma akadna, tudja a kódoló, hol is keresse elsősorban.

Vegyük sorra a fontosabbakat, ami a könyvtár oldalon van. Kezdjük magával a könyvek tárolásával. Úgy oldottam meg hogy először, ha rámegyünk a könyveink oldalra, akkor kiadja az adott könyveket, amik tárolva vannak. A könyv címét, kategóriáját és hogy elérhető-e. Ez egybe van építve egy kereséssel. Amennyiben a keresőbe beírjuk a kívánt címet akkor kiadja az adott címből az összes könyvet összes adatával.

A könyveket az adatbázisunkban tároljuk. Előbb azt el kell érni és lehívni.

$query = "SELECT bname, category, available FROM books WHERE bname NOT LIKE '[%]%' GROUP BY bname ORDER BY bname ASC";

Amennyiben a keresés van használatban akkor pedig ez az SQL parancs fut le.

$query = "SELECT id, bname, category, place, available FROM books WHERE bname NOT LIKE '[%]%' AND bname LIKE :search ORDER BY bname ASC, available DESC, place ASC";

Ez egy if else ágban van megvalósítva.

A következő fontosabb kódrészlet a saját kikölcsönzött könyvekhez tartozik.

a Könyveim oldalra kattintva megjelennek azok a könyvek, amiket épp kikölcsönöztünk és nálunk vannak. Itt is PDO-t alkalmaztam egy funkción belül.

function books\_list()

{

global $database;

$stmt = $database->runQueryBooks("SELECT books.id, books.bname, loan.lstart, loan.lend

FROM books

INNER JOIN loan ON books.id = loan.bookid

WHERE loan.userid=:id ORDER BY bname ASC");

$stmt->bindParam(':id', $\_SESSION['uid']);

$stmt->execute();

$result=$stmt->fetchAll();

return $result;

}

Ezután az oldalra felhasználtam ezeknek az adatoknak a tárolását és megjelenítettem őket.

Az weboldal készítésében nagyon sok helyen PDO szisztematikát alkalmazom. Sokkal hatékonyabb és számomra kedvezőbb.

# Összefoglalás

## Összehasonlítás

A feladatom egy könyvtárprogram megírása volt. Ezt a feladatot weboldalra írtam meg. Ezen belül viszont két fejlesztői környeztet választottam összehasonlítás képpen. Az egyik a saját fejlesztés volt, míg a másik egy keretrendszeres fejlesztés. Az észrevételeim megoszlóak.

Beszéljünk kicsit a saját fejlesztésről. Amikor ezt a fejlesztői környezetet válasszuk rendelkezzünk a feladat megoldására elég idővel. Ugyanis ez több időt igénybe vesz, mint az előre elkészített sablonok. Itt mindent nekünk kell megírni a nulláról. A kódokat, a dizájnt, szkripteket. Viszont ennek cserébe olyanra szabhatjuk a weboldalt, amire csak szeretnénk. Nagyon sok lehetőségünk van a kódolásban és a dizájn kialakításában is. Saját magunk kialakíthatjuk a rendszerét is. Egyedi weblapokat is létrehozhatunk így.

A WordPress fejlesztői környezete valóban gyorsabb. A sok widgetek és sablonoknak köszönhetően hamar összelehet állítani egy weboldalt, ami megfelel bizonyos céloknak. Ellenben viszont nem annyira szabad, mint egy saját fejlesztésű. A meglevő dolgokból válogathatunk, szerkeszthetünk.

Összességében én a saját fejlesztést jobban preferálom. Szabadabb és átláthatóbb rendszert tudtam kialakítani, amely élvezetessé és gyorsította is a munkámat.

## Feladat összegzése

Első lepésnek megterveztem az adatbázist és a programot. Hogyan is fog felépülni. Az adatbázis tervezése után telepítettem a XAMPP nevű programot. Létrehoztam az adatbázist majd feltöltöttem adatokkal, amikkel dolgoztam. Következő lépés a weboldal megírása volt. Először megírtam a HTML részét, ami a váza volt az egésznek, majd a stíluslapokat, illetve a szkripteket hozzá. Ezután a PHP kódolása jött. Végezetül tesztelések, hibajavítások. A jelenlegi weboldal még továbbfejleszthető. Gondolok itt olyan funkciókra, hogy belehetne építeni egy futárszolgálatot, ami kiszállítja a kikölcsönzött könyveket és a lejárati időpontban vissza is viszi azokat a könyveket. Ezt a weboldalon is megjelenhető opció lenne.

# Irodalomjegyzék

<https://hu.wikipedia.org/wiki/MySQL>

<https://hu.wikipedia.org/wiki/WordPress>

[https://www.w3schools.com](https://www.w3schools.com/)

# Ábrajegyzék

[1. ábra WordPress regisztráció 10](#_Toc69665860)

[2. ábra WordPress kezdőfelület 11](#_Toc69665861)

[3. ábra WordPress weboldal szerkesztése 11](#_Toc69665862)

[4. ábra XAMPP telepítése 12](#_Toc69665863)

[5. ábra XAMPP felülete 13](#_Toc69665864)

[*6*. ábra Adatbázis: felhasználói tábla 17](#_Toc69665865)

[7. ábra Adatbázis: könyvek tábla 17](#_Toc69665866)

[8. ábra Adatbázis: kölcsönzés tábla 17](#_Toc69665867)

1. (A záródolgozat leadásig kötelező a három konzultációs alkalom, ennek hiányában a záródolgozat nem értékelhető!)

   [↑](#footnote-ref-1)