
CSALÁDINYÁR

WEBES ALKALMAZÁS SZÁLLÁSADÓKNAK

VIZSGAREMEK DOKUMENTÁCIÓ

2023. JANUÁR 2.

Készítette: Fulajtár Mónika

Tartalom

1	BEVEZETÉS	3
1.1	TÉMAVÁLASZTÁS	3
1.2	A VIZSGAMUNKA CÉLJA	3
2	FEJLESZTŐI DOKUMENTÁCIÓ	4
2.1	A PROGRAM FUTÁSÁHOZ SZÜKSÉGES RENDSZERKÖVETELMÉNYEK, PROGRAMOK	4
2.1.1	<i>Visual Studio Code</i>	4
2.1.2	<i>XAMPP</i>	4
2.1.3	<i>MYSQL WORKBENCH</i>	4
2.1.4	<i>PHP</i>	5
2.1.5	<i>Javascript</i>	5
2.1.6	<i>JSON</i>	5
2.1.7	<i>OOP</i>	5
2.1.8	<i>MVC</i>	6
2.1.9	<i>Symfony</i>	7
2.1.10	<i>Composer</i>	7
2.1.11	<i>Monolog</i>	9
2.1.12	<i>Twig</i>	10
2.1.13	<i>REST</i>	10
2.2	A FRONTEND OLDAL KIALAKÍTÁSÁHOZ HASZNÁLT PROGRAMOK	10
2.2.1	<i>HTML</i>	10
2.2.2	<i>CSS</i>	11
2.2.3	<i>BOOTSTRAP</i>	11
2.2.4	<i>GIT, GITHUB</i>	11
2.3	ADATBÁZIS	11
2.3.1	<i>Adatbázis tervezése</i>	11
2.3.2	<i>Az adatbázis létrehozása</i>	12
2.3.3	<i>Az adatbázis részletezése</i>	13
2.4	AZ MVC FELÉPÍTÉSE	14
2.4.1	<i>A program mappaszerkezete</i>	14
	<i>A vizsgamunka a vizsgamunka elnevezésű mappa létrehozásával kezdődött.</i>	14
2.4.2	<i>A Controller</i>	15
2.4.3	<i>A Model</i>	15
2.4.4	<i>A View</i>	16
3	FELHASZNÁLÓI DOKUMENTÁCIÓ	17
3.1	AZ ADMIN FELÜLET	17
3.1.1	<i>Vendégházak kezelés menüpont / houses</i>	19
3.1.2	<i>Felhasználók kezelés menüpont</i>	20
3.1.3	<i>A manager oldal /manager</i>	20
3.1.4	<i>Tulajdonos oldal / owner</i>	21
3.1.5	<i>A jogosultságok kezelése / title</i>	21
3.1.6	<i>Sliderek kezelése / slider</i>	21
3.1.7	<i>Jelszó kezelése</i>	22
3.2	A LÁTOGATÓK OLDAL	23
3.2.1	<i>A látogatói oldal kialakítása</i>	23
3.2.2	<i>Kezdőoldal / http://localhost:8000/</i>	23
4	TESZTELÉS	25
4.1.1	<i>Az oldal tesztelése</i>	25
4.1.2	<i>Jelszó tesztelése</i>	25

4.1.3	<i>Responsivitas ellenőrzése</i>	26
5	A PROGRAM TOVÁBBI FEJLESZTÉSÉNEK TERVE	26
6	MELLÉKLETEK	26
6.1	AZ ADATBÁZIS MODELL DIAGRAMJA.....	26
6.2	AZ _INSTALL SCRIPT FILE.	26

1 BEVEZETÉS

1.1 Témaválasztás

Családi vállalkozásunk vendégházak kiadásával foglalkozik már több éve. A vállalkozásunkban jelenleg két családi nyaraló működik az egyik Szarvason a másik Békésszentandrás. A térségben közel 1000 vendégház, nyaraló bérelhető eltérő felszereltséggel, különböző szabadidős tevékenységgel. Az ide látogatók többféle ház közül választhatnak, és különféle szolgáltatásokat keresnek. Sok vendég visszajár évek óta, ismeri a környéket. A térségben a turizmus számtalan magánembernek is megélhetést biztosít, így fontosnak tartom a turizmus fejlesztését és az ide látogatók számának a növelését. A vizsgamunkában elkészített program alapja lehet egy olyan alkalmazásnak, melyet az ide érkező vendégek, illetve a nyaraló tulajdonosok is egyaránt hasznosnak találhatnak.

1.2 A vizsgamunka célja

A vizsgamunkám elsődleges célja az volt, hogy a jelenleg WORDPRESS keretrendszerben létrehozott, egyszerű alapokon működő www.csaladinyar.hu weboldal egy sokkal interaktívabb, felhasználóbarát, és aktuális információkat tartalmazó weboldal legyen, melyre a későbbiekben akár több nyaraló is felkerülhetne. A nyaralótulajdonosok mellett szeretném a helyi kézművesekkel, vállalkozókkal is felvenni a kapcsolatot. Ennek célja, hogy az ide látogatók a weboldal segítségével, már előzetesen is tudjanak tájékozódni a számukra releváns programokról, az üzletek nyitvatartási idejéről, a helyi kézműves vállalkozásokról, azok termékeiről, vagy az éttermek étlapjairól. A program kizárólag e térségben található vendégházakra, a környéken megrendezésre kerülő programokra lenne létrehozva, de tartalmazna egyéb hasznos információkat, pl.: a horgászatra vonatkozó híreket, bérelhető elektromos eszközöket, hajóutakat, éttermi asztalfoglalásokat.

2 FEJLESZTŐI DOKUMENTÁCIÓ

2.1 A program futásához szükséges rendszerkövetelmények, programok

2.1.1 Visual Studio Code¹

A vizsgaremekben használt kód megírásához a Visual Studio Code programot használtam. A Visual Studio Code egy ingyenes használható kódszerkesztő, melyet a Microsoft fejleszt. A program rövidített nevén a VSC Windows, Linux és OS X operációs rendszereken is használható, Igen sok telepíthető bővítménnyel rendelkezik, melyek nagymértékben segítik a fejlesztők munkáját, felhasználói felülete személyre szabható.

2.1.2 XAMPP

A XAMPP egy szabad és nyílt forrású webserver-programcsomag, mely ingyenesen használható. A XAMPP egy betűszó ill. rövidítés, betűi a következő kifejezésekből származnak:

- X-cross-platform, azaz platformfüggetlenség
- A - Apache webszerver
- M - MariaDB adatbázis kezelő (korábban MySql)
- PHP - szerveroldali szkript nyelv
- Perl- általános célú szkript nyelv. A XAMPP segítségével a számítógép egy lokális szerverként használható, viszont hálózati gépként is hozzáférhetővé válik, valamilyen klienskezelő programmal. A program elkészítésekor a MYSQL adatbázis kapcsolódásához használtam.

2.1.3 MYSQL WORKBENCH

Egy relációs adatbázis kezelő program, melynek segítségével tárolhatjuk a programunkhoz szükséges adatokat, különböző adattáblák segítségével. A MySQL Workbench adatmodellezést, SQL-fejlesztést és átfogó adminisztrációs eszközöket kínál a szerverkonfigurációhoz, a felhasználók adminisztrációjához, a biztonsági mentéshez és még sok másához.

¹ https://hu.wikipedia.org/wiki/Visual_Studio_Code

2.1.4 PHP

A *Personal Home Page Tools* ² rövidített nevén a PHP egy szerveroldali szkript nyelv, mely HTML oldalba ágyazható. A PHP születésekor egy makrókészlet volt, mely személyes honlapok karbantartását szolgálta, mára azonban egy önálló programozási nyelvvé fejlődte ki magát. Elsősorban webfejlesztésnél használják, mely képes nagyméretű webes adatbázisalapú alkalmazások működtetésére is.

2.1.5 Javascript

A Javascript³ egy kliensoldalú programozói nyelv, melyet weboldalakon elterjedten használnak. A háttérben futva, a böngésző HTML lapjának szövegébe épül be. Azért fejlesztettek ki, hogy a különféle weboldalak komplexebb, vagyis sokkal gazdagabb és élvezetesebb felhasználói élményt nyújthassanak a weben böngészők számára. Elmondható róla, hogy a legtöbbet használt programozási nyelv.

2.1.6 JSON

A JSON (JavaScript Object Notation) egy könnyű adatcsere-formátum, amely könnyen olvasható és írható az emberek számára, a gépek pedig könnyen értelmezhetők és generálhatók. A JavaScript programozási nyelv egy részhalmazán alapul, és adatstruktúrák és objektumok szabványos, ember által olvasható formában történő megjelenítésére szolgál. Gyakran használják adatok interneten keresztüli továbbítására vagy adatok strukturált formátumban történő tárolására. Ez az adatcsere általános formátuma, különösen a webes alkalmazásokban, mivel könnyen olvasható és írható, és minimális többletköltséget igényel.

2.1.7 OOP

Az OOP vagyis az Objektum Orientál programozási paradigma, egységbe foglalják az adatokat és a hozzájuk tartozó műveleteket. Az OOP-t gyakran használják újra felhasználható és moduláris kód létrehozására, ami megkönnyítheti az összetett szoftverrendszerek fejlesztését és karbantartását. Az objektumorientált programozás minden objektumhoz kapcsolódik, és az objektumok egyetlen entitásba csoportosítva tartalmazzák az adatokat. Az OOP osztályokat használ az objektumok definiálására. Az osztályok tagokat

² https://hu.wikipedia.org/wiki/PHP#cite_note-History_of_PHP-5

³ https://hu.wikipedia.org/wiki/JavaScript#Szerkeszt%C3%A9s,_el%C5%91%C3%A1ll%C3%ADt%C3%A1s

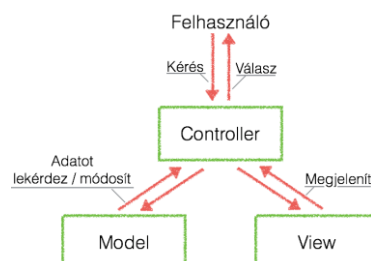
tartalmaznak: tulajdonságokat, melyek jellemzik az osztályt és metódusokat, amik valamilyen működést hajtanak végre. Példával:

pl.: egy autó esetében az osztály lesz az Auto, a tulajdonságok pedig az osztályon belül vannak.

```
class Autó{  
public $gyarto;  
public $tipus;  
public $ár;  
public $kategoria  
}
```

2.1.8 MVC

Az MVC⁴ (Modell - View - Controller) vagy a Modell-Nézet-Vezérlő egy programtervezési minta, melyet szoftvertervezéshez használnak. Három alapvető komponensre bontja a grafikus alkalmazásokat: a *Model* az alkalmazás adataiért felel, tartalmazza az adatok tárolásához és módosításához szükséges logikai kódokat. A *View* megjeleníti az alkalmazás adatait a felhasználó számára. A *Vezérlő* vagyis a *Controller* figyeli a felhasználó által küldött kéréseket, és átadja azt a modell megfelelő osztálymetódusainak a feldolgozásra, így biztosítva a kapcsolatot a felhasználói felület (nézet) és az alkalmazásfeldolgozási logika (modell) között. Működésének lényege és egyben előnye, hogy az adatok és a felhasználói felület elkülöníthető egymástól, így egyszerre többen is dolgozhatnak ugyanazon a fejlesztésen anélkül, hogy zavarnák egymás munkáját. Működését az alábbi ábra szemlélteti:



⁴ <https://hu.wikipedia.org/wiki/Modell-n%C3%A9zet-vez%C3%A9rl%C5%91>

2.1.9 Symfony⁵

A Symfony egy PHP alapú webalkalmazás keretrendszer és újra felhasználható komponensek/könyvtárak készlete. 2005 októberében jelent meg egy ingyenes szoftverként. Ez egy nyílt forrású keretrendszer, amelyhez így mindenki hozzáadhatja a saját modulját. Célja a webalkalmazások létrehozásának és karbantartásának a felgyorsítása, valamint az ismétlődő feladatok helyettesítése. A Symfony PHP keretrendszer alapértelmezett sablonmotorként Twig-et használ.

2.1.10 Composer⁶

Egy függőségeket kezelő eszköz, melyet PHP nyelvben szoktunk használni. A Composer különböző külső könyvtárakat használ, többféle függőség telepíthető a segítségével, legújabb verziójának futtatásához PHP 7.2.5 szükséges. A Composer segítségével telepíthetők különböző sablonkezelők, templatek, ilyen pl. a Twig, melyet ebben a vizsgaremekben is használtam. Legújabb verziója a 2.4.4. verziómmal rendelkezik.

Ahhoz, hogy a vizsgamunkámban használni tudjam, először le kellett töltenem, majd futtatnom a *Composer-Setup.exe* fájlt.

A futtatást követően a CMD parancssorban az 1. ábra. -n szereplő *composer init* parancsot kellett kiadnom. A telepítéskor figyeltem arra, hogy abba a mappába telepítsem fel, amelyben az adott projekt is megtalálható, azaz a parancs kiadása előtt ki kellett navigálni a megfelelő mappába.

⁵ <https://symfony.com/>

⁶ <https://getcomposer.org/>


```
Parancssor
Microsoft Windows [Version 10.0.19045.2364]
(c) Microsoft Corporation. Minden jog fenntartva.

C:\Users\monika>cd/

C:\>cd xampp

C:\xampp>cd htdocs

C:\xampp\htdocs>cd vizsgamunka

C:\xampp\htdocs\vizsgamunka>composer init
```

2. ábra

A telepítés során megadható a projekt megnevezése, hozzáadható esetleg egy leírás (pl.: milyen rendszert építünk fel). Lehetőséget ad a szerző megadására is. A telepítés végén a composer automatikusan létrehozza a szükséges könyvtárakat, bennük a megadott adatokkal. A telepítéskor beállításra került *autoload* mappa létrehozása. Ennek az az előnye, hogy az osztályokat automatikusan betölti majd, így kihagyható a file elérési útjának a megadása. Ahhoz, hogy az elérési utak betöltődjenek, egy hivatkozást kell hozzáadni majd az index.php file-hoz, melyet a 2. ábra mutat.

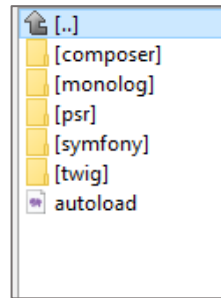
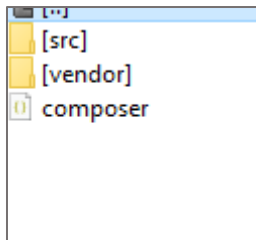
```
index.php > ...
1  <?php
2
3  require __DIR__ . '/vendor/autoload.php';
4
```

1. ábra

Ezzel a projektünket Composer projektté alakítottuk, és létrehoztuk az alapbeállításokat. A telepítés után a Composer által automatikusan létrehozott SRC mappát átneveztem App mappának, ezután pedig az elérési utat a composer.json file-ban is megadtam 2. ábra .

```
composer.json > {} require > monolog/monolog
1  {
2      "name": "monika/vizsgamunka",
3      "description": "ruander vizsgaprojekt",
4      "autoload": {
5          "psr-4": {
6              "App\\": "App/"
7          }
8      },
9      "authors": [
10         {
11             "name": "Monika Fulajtar"
12         }
13     ]
14 }
```

2. ábra



A telepítés végén a Composer automatikusan létrehoz egy JSON file-t, és létrehozza a mappákat 3. ábra.

3. ábra

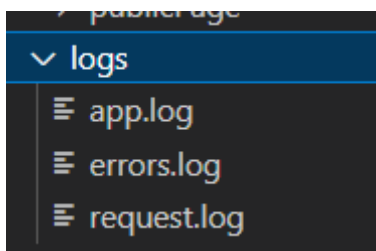
2.1.11 Monolog

A Monolog egy függőség, egy naplózó könyvtár a PHP számára. Egyszerű és rugalmas módot biztosít a naplóüzenetek írására, ami hasznos lehet az alkalmazásokban előforduló események hibakereséséhez, figyeléséhez és nyomon követéséhez. A Monolog segíthet a kód hibáinak és problémáinak nyomon követésében azáltal, hogy naplózza az alkalmazásban zajló eseményeket. A naplózást -log file-okban rögzíti (5. ábra). A 'monolog' telepítéséhez az alábbi kódot használtam és a parancssorban futtattam le.

```
composer require monolog/monolog
```

4. ábra

A lefutása után a már meglévő 'vendor' mappában létrejött egy új mappa a 'monolog'.



5. ábra

2.1.12 Twig⁷

Modern PHP sablonmotor egy template kezelő, amely lerövidíti és megkönnyíti a PHP használatát. A Twig tömör szintaxissal dolgozik, ami a sablonokat olvashatóbbá teszi. Jellemzője, hogy nagyon rugalmas, nagy projekteknél is biztonságosan használható. A Twig egyik célja, hogy a lehető leggyorsabb legyen. A lehető legjobb sebesség elérése érdekében a Twig a sablonokat egyszerű optimalizált PHP kódra fordítja. A normál PHP kódhoz képest a rezsit a minimumra csökkent. Feltelepítését a Composer segítségével, a windows parancssorban végeztem el az alábbi kóddal:

```
composer require "twig/twig:^3.0"
```

2.1.13 REST⁸

Egy http protokollra fejlesztett kommunikációs architektúra, mely kliens és szerver közti kommunikáció megvalósítását szolgálja. A kérések URI-k használatával történnek, melyek erőforrásokat azonosítanak, és az URI-k egységes interfészt biztosítanak a kliens számára. Minden kérésre azonos formátumban reagál a szerver, ez általában JSON, de használható még HTML és XML formátum is. Egy REST típusú architektúra kliensekből és szerverekből áll. A kliensek kéréseket indítanak a szerverek felé; a szerverek kéréseket dolgoznak fel és a megfelelő választ küldik vissza.

2.2 A frontend oldal kialakításához használt programok

2.2.1 HTML

A HTML⁹ rövidítés a Hypertext Markup Language, mely egy leíró nyelv. A weboldalak megjelenítésére fejlesztettek ki, legújabb verziója a HTML 5.2 2017-ben indult el. Eredetileg egy olyan kód volt, ami egyszerű, statikus szövegek írására alkalmas. A jelenlegi verzió már képes önmagában, külső szoftverek nélkül megbirkózni animációkkal, videókkal, audio fájlokkal, de webappok is létrehozhatók a segítségével, valamint együttműködik a JavaScripttel. Minden böngésző képes a megjelenítésükre.

⁷ <https://twig.symfony.com/>

⁸ <https://hu.wikipedia.org/wiki/REST>

⁹ <https://hu.wikipedia.org/wiki/HTML>

2.2.2 CSS¹⁰

A CSS a *cascading style sheet* kifejezés rövidítése, ami lefordítva egymásba ágyazott stíluslapokat jelent. A CSS felel a weblap kinézetéért, vagyis amit a HTML-ben egyszerű szöveggént megszerkesztettünk, annak itt van lehetőségünk adni egy személyre szabott egyedi arculatot. A CSS-t úgy tervezték, hogy lehetővé tegye a tartalom és a megjelenítés elkülönítését, beleértve az elrendezést, a színeket és a betűtípusokat. Ez a szétválasztás javíthatja a tartalom hozzáférhetőségét; nagyobb rugalmasságot adhat neki. Egy külön .css file-ban van lehetőség az arculati elemek megírására, ezzel csökkentve a szerkezet bonyolultságát.

2.2.3 BOOTSTRAP¹¹

A Bootstrap egy nyílt forráskódú keretrendszer (framework), mely HTML, CSS, JavaScript technológiákat használ. Alapvetően arra jó, hogy nagyon könnyedén, és minimális energia befektetéssel tudjon valaki jól kinéző, bármilyen képernyőméreten szépen megjelenő weboldalakat készíteni. Legújabb verziója a 5.3 verzió.

2.2.4 GIT, GITHUB¹²

A vizsgamunka beadásához szükséges programot a <https://gitforwindows.org/> oldalról töltöttem le. Úgynevezett szoftverforrást, verziókövetést kezelő rendszer, melyet használhatunk biztonsági mentésre, illetve egyszerre több felhasználó is tudja egy időben használni ugyanazokat a mappákat.

2.3 ADATBÁZIS

2.3.1 Adatbázis tervezése

Az adatbázis tervezését egy EXCEL file-ban kezdtem el (6. ábra) **Hiba! A hivatkozási forrás nem található..** Külön táblázatokban rögzítettem, hogy milyen adatokra lenne szükségem elsősorban az alkalmazásban. A meglévő adatbázis nem végleges, szükséges további adatfeltöltés.

¹⁰ <https://en.wikipedia.org/wiki/CSS>

¹¹ <https://getbootstrap.com/>

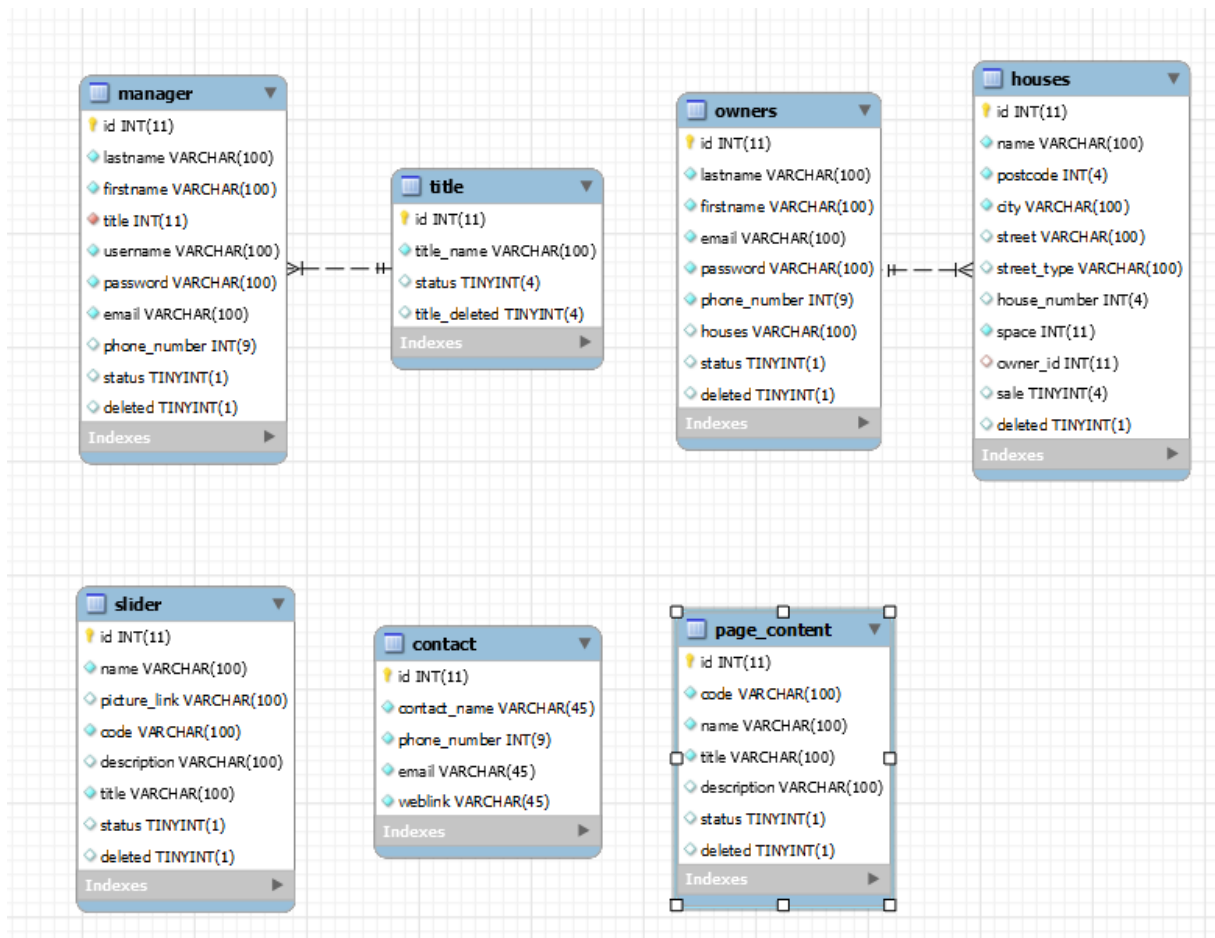
¹² <https://github.com/>

CSALÁDINYÁR alkalmazás adatbázis TERV									
manager		Title		owner		houses			
id	INT(11); AI; PK; NN;	id	INT(11); AI; PK	id	INT(11); AI; PK; NN;	id	INT(11); AI; PK		
lastname	VARCHAR(100);NN	title_name	VARCHAR(100)	lastname	VARCHAR(100);NN	name	VARCHAR(100)		
firstname	VARCHAR(100)	status	TINYINT(4); DEFAULT:1	firstname	VARCHAR(100);NN	postcode	INT(4);		
title	INT(11); FK;NN	title_deleted	TINYINT(4) DEFAULT:0	email	VARCHAR(100);NN	city	VARCHAR(100)		
username	VARCHAR(100);NN			phone_number	INT(9); NN	street	VARCHAR(100)		
password	VARCHAR(100);NN			password	VARCHAR(100)	street_type	VARCHAR(100)		
email	VARCHAR(100); UQ;NN			houses	VARCHAR(100)	house_number	INT(4);		
phone_number	INT(9)			status	TINYINT(4); DEFAULT:1	space	INT(11);		
status	TINYINT(4); DEFAULT:1			deleted	TINYINT(4) DEFAULT:0	owner_id	INT(11); FK		
deleted	TINYINT(4) DEFAULT:0					sale	TINYINT(4)		
						deleted	TINYINT(1)		
slider		contact		page content					
id	INT(11); AI; PK; NN;	id	INT(11); AI; PK; NN;	id	INT(11); AI; PK; NN;				
name	VARCHAR(100)	contact_name	VARCHAR(100)	code	VARCHAR(100); NN				
picture_link	VARCHAR(100)	phone_number	INT(9); NN	name	VARCHAR(100)				
code	VARCHAR(100); NN	email	VARCHAR(100);NN	title	VARCHAR(100);NN				
description	VARCHAR(100)	weblink	VARCHAR(100)	description	VARCHAR(100)				
title	VARCHAR(100);NN			status	TINYINT(1); DEFAULT:1				
status	TINYINT(1); DEFAULT:1			deleted	TINYINT(1) DEFAULT:0				
deleted	TINYINT(1) DEFAULT:0								

6. ábra

2.3.2 Az adatbázis létrehozása

Az adatbázis létrehozását a MYSQL Workbench programban hoztam létre. A lokális szerverhez való kapcsolódás után elkészítettem a 'family' sémát. Az

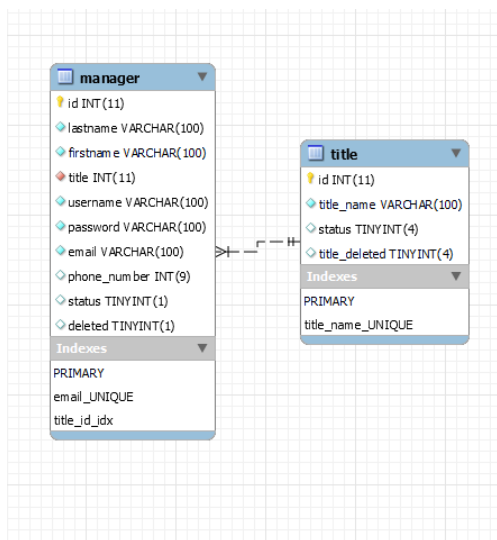


7. ábra

adatbázis és annak tartalmának létrehozásához szükséges parancsokat a vizsgamunka mellékleteként beadott *family.sql* file tartalmazza.

2.3.3 Az adatbázis részletezése

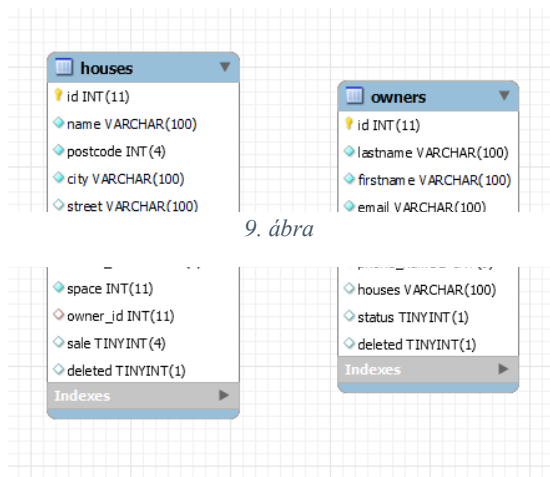
Az adatbázisban hét tábla található (8. ábra): *manager*, *title*, *owner*, *houses*, *slider*, *contact*, *page_content*. A *manager*, *title*, *owner* és *houses* táblákban szereplő adatokat jelenleg csak az admin felületen használom. A *contact*, *slider*, és *page_content* táblázatok pedig a felhasználói oldalon található adatokat tartalmazzák.



8. ábra

A *manager* és *title* táblákat összekötöttem egy Foreign Key segítségével: létrejött a 'title_id', amely a 'manager' tábla 'title' oszlopának adatait köti össze a 'title' tábla 'id' oszlopában lévő értékkel. Ennek a célja, hogy a megjelenítéskor a managerek neve mellett a titulusuk jelenleg meg szöveges formában **Hiba! A hivatkozási forrás nem található.**9. ábra).

Szintén Foreign Key összekötés történt az 'owners' és 'houses' tábláknál is 10. ábra : a 'houses' tábla 'owner_id' oszlopa összekötésre került az 'owner' tábla 'id' oszlopával. A Foreign key-ek összekötések NO ACTION paraméterrel lett létrehozva mindkét esetben, azaz nem módosíthatóak és nem törölhetőek, amíg a másik hozzákapcsolt tábla is valamilyen adatot tartalmaz.



9. ábra

2.4 Az MVC felépítése

2.4.1 A program mappaszerkezete

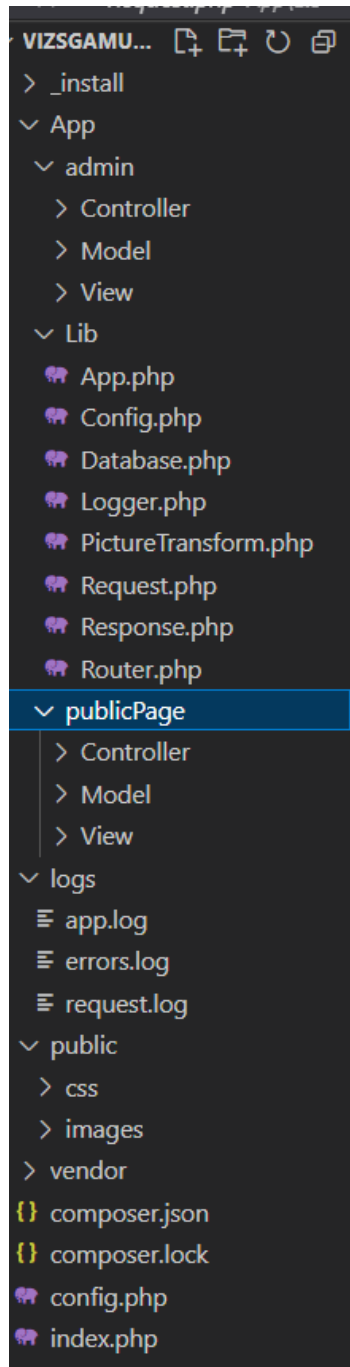
A vizsgamunka a vizsgamunka elnevezésű mappa létrehozásával

kezdődött.

Az előzőekben leírtak szerint a Composer telepítésével létrejöttek már almappák, amelyekbe automatikusan generált file-ok kerültek. Ezt követően a mappaszerkezetet az alábbiak szerint alakítottam ki 11. ábra:

Az `_install` mappa tartalmazza a programhoz tartozó *family.sql* file-t.

10. ábra



Az **App** mappát szétválasztottam az admin, a Lib és a publikus oldalakra, létrehozva ezeknek egy-egy új mappát (admin, Lib, publicPage).

Mind az admin, mind a publikus oldal tartalmaz

- Controller
- Modell és
- View mappát. Ez a mappa felosztás segít abban, hogy a megfelelő kódokat a megfelelő mappába tegyem.

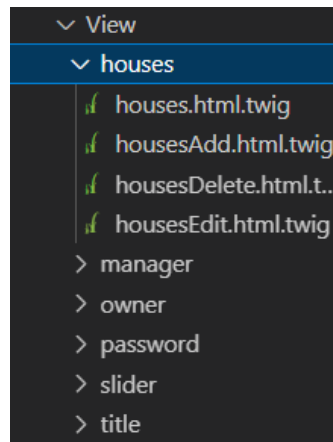
A **Lib** mappa tartalmazza az általánosabb, mindkét felületre vonatkozó konfigurációs beállításokat, az adatbázishoz kapcsolódást, a sliderek feltöltéséhez szükséges metódusokat, feltételeket, stb.

Minden mappában látható, hogy a létrehozott file-ok nagy kezdőbetűvel vannak írva. Ennek az a magyarázata, hogy az objektumorientált programozásban így jelzik az osztályokat.

A **logs** mappa a 'monolog' vagyis a naplózó rendszer által elmentett adatokat tartalmazza. A **public** mappa a képeket és a css stíluslapokat tartalmazza. A vendor

mappa a Composer által automatikusan létrehozott mappa.

A View mappát is felosztottam, így könnyebben átlátom a megjelenítési réteget, vagyis a végpontokhoz szükséges twigeket:



A *config.php* és az *index.php* file-okat már én hoztam létre. Az *index.php* az oldalak betöltéséért felel, endpoint-okat tartalmaz, vagyis itt a Controllerben létrehozott metódusokat hívjuk meg. Az OOP-ra jellemző osztályok létrehozása jól látszik a file-ok szerkezetén, mindenhol osztályokat hoztam létre.

2.4.2 A Controller

A Controller feladata, hogy a Model rétegen belül lévő, a megfelelő file-hoz kapcsolódó kéréseket továbbítsa a View vagyis a megjelenítési réteg felé. A Controlleren semmilyen adatbázis művelet nem hajtható végre. Itt az esetleges feltételek szabhatunk meg. Innen az adatok a View megjelenítés oldalra kerülnek át.

2.4.3 A Model

A Model vagy adatbázisrétegen belül létrehozott file-ok tartalmazznak minden olyan műveletet, amely az adatbázissal kapcsolatos. Itt kapcsolódunk a táblákhoz és megadhatjuk, hogy milyen adatokra van szükségünk a táblákból. Az itt lekért adatok kerülnek át a Controllerbe, majd onnan pedig a View rétegbe.

A lenti képernyőképen az *all()* metódust hoztam létre a Model rétegen belül, melyben először kapcsolódtam az adatbázishoz, majd a szükséges táblából kieszedtem az adatokat. Ebben a példában szükséges volt két tábla összekapcsolása az INNER JOIN segítségével, mivel a View, vagyis a megjelenítés rétegen a

managerek megnevezését, vagy titulását nem egy számmal, hanem szövegesen szeretném megjeleníteni.

```
class ManagerDao implements Icrud {  
    public static function all()  
    {  
        $dbObj = new Database();  
        $conn = $dbObj->getConnection();  
        $sql= "SELECT m.id, m.lastname, m.firstname, m.title, m.username, m.password, m.email, m.phone_number, m.status, t.title_name FROM manager  
        as m INNER JOIN title as t ON m.title = t.id WHERE deleted =0 ORDER BY m.id ;";  
        $statement = $conn->prepare($sql);  
        $statement->setFetchMode(PDO::FETCH_OBJ);  
        $statement->execute();  
        return $statement->fetchAll();  
    }  
}
```

Ezután az itt létrehozott all() metódust meghívtam a Controllerben, ami a list metódussal pedig elküldi a View rétegnek az adatokat, a twig segítségével a megfelelő endpointra irányítjuk az adatokat.

```
public static function list()  
{  
    $data = ManagerDao::all();  
    $twig = (new ManagerController())->setTwigEnvironment();  
    echo $twig->render('manager/manager.html.twig', ['managers' => $data]);  
}
```

Az index.php -ban vannak a végpontok, amelyekre a metódust alkalmazhatjuk, és megjeleníthetjük a Model rétegben lekért adatokat.

```
Router::get('/manager', function(Request $req, Response $res){  
    return (new ManagerController())->list();  
});
```

2.4.4 A View

A View réteg felel az adatok megjelenítésért, vagyis a felhasználó az itt kialakított felület látja. Az általam választott Twignek legnagyobb szerepe ebben a rétegben van. A rétegek kialakításánál a .html végződés kiegészítésre kerül a .twig végződéssel. A View nézet kialakításában ha adatokat kértünk be az adatbázisból, azt a twig segítségével tettük meg, a függőség szabályai szerint. Itt látszik, hogy a twig használata hogyan egyszerűsítheti le a php programozást.

3 FELHASZNÁLÓI DOKUMENTÁCIÓ

3.1 Az admin felület

A program egyelőre lokális felületen fut, nincsen élesbe állítva.

A futtatáshoz a PHP beépített szerverét használtam, melyet a parancssorból tudok elindítani az alábbi parancs kiadásával: `php -S localhost:8000`. Fontos, hogy az indítás előtt a megfelelő mappába kinavigáljunk!

```
Microsoft Windows [Version 10.0.19045.2364]
(c) Microsoft Corporation. Minden jog fenntartva.

C:\Users\monika>cd/

C:\>cd xampp

C:\xampp>cd htdocs

C:\xampp\htdocs>cd valami

C:\xampp\htdocs\valami>cd..

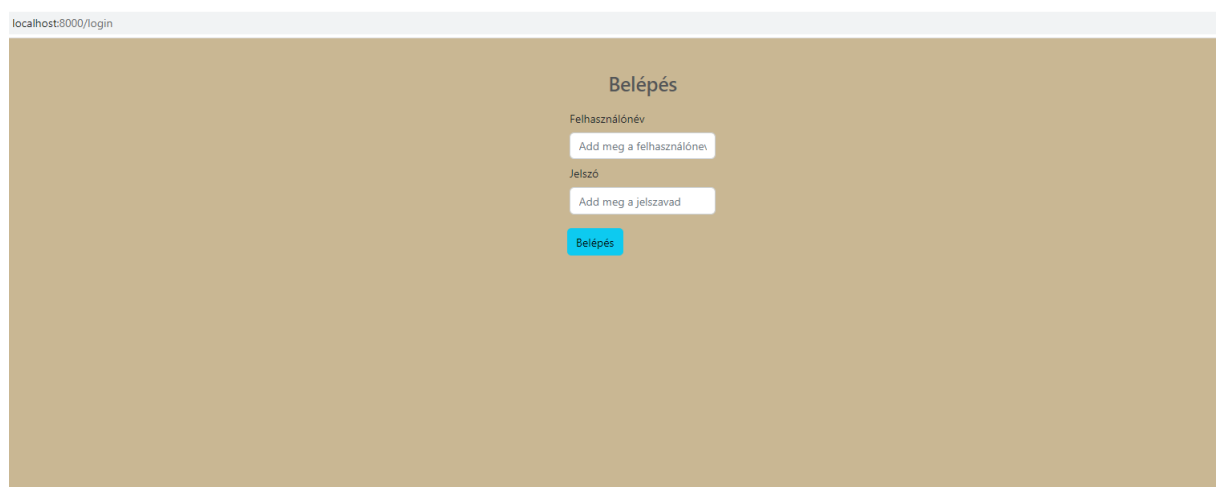
C:\xampp\htdocs>cd vizsgamunka

C:\xampp\htdocs\vizsgamunka>php -S localhost:8000
[Tue Jan  3 12:41:54 2023] PHP 8.1.12 Development Server (http://localhost:8000) started
```

Az alkalmazás így az admin felület is jelenleg a <http://localhost:8000/admin> oldalon érhető el.

Az admin oldalon kétféle belépésre van lehetőség. A managerek és tulajdonosok külön felületre lépnének. (Jelenleg a login és a jogosultságok kialakításához további fejlesztés szükséges).

Kiválasztva valamelyiket a <http://localhost:8000/login> oldalra lépünk át:



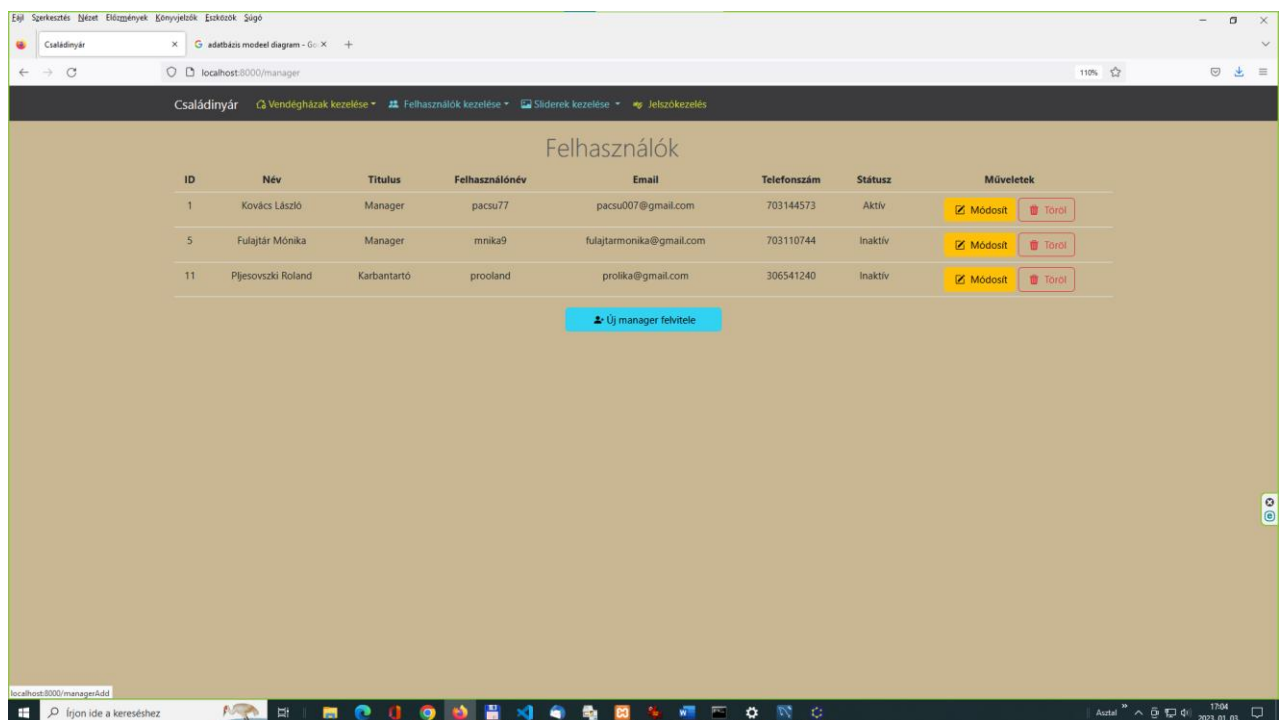
Ha a felhasználó valamelyik adatot nem tölti ki, akkor az alkalmazás hibaüzenetet dob, melyet a 12. ábra mutat. A felhasználónév és jelszó beírásával az admin felületre lépünk.

A jelszó hiányzik

A felhasználónév hiányzik

11. ábra

A belépés után a managerek listáját látjuk, és az alábbi kép jelenik meg:



ID	Név	Titulus	Felhasználónév	Email	Telefonszám	Státusz	Műveletek
1	Kovács László	Manager	pacsu77	pacsu007@gmail.com	703144573	Aktív	Módosít Töröl
5	Fulajtár Mónika	Manager	mnika9	fulajtarmonika@gmail.com	703110744	Inaktív	Módosít Töröl
11	Píjesovszki Roland	Karbantartó	prooland	prolika@gmail.com	306541240	Inaktív	Módosít Töröl

[Új manager felvétele](#)

A menüpontok alatt lehetőségünk van a manager, title, owner és houses táblák adatainak a módosítására, esetleges új adat felvitelére, vagy valamelyik adat törlésére. Egy menüpont felel a jelszó módosításáért.

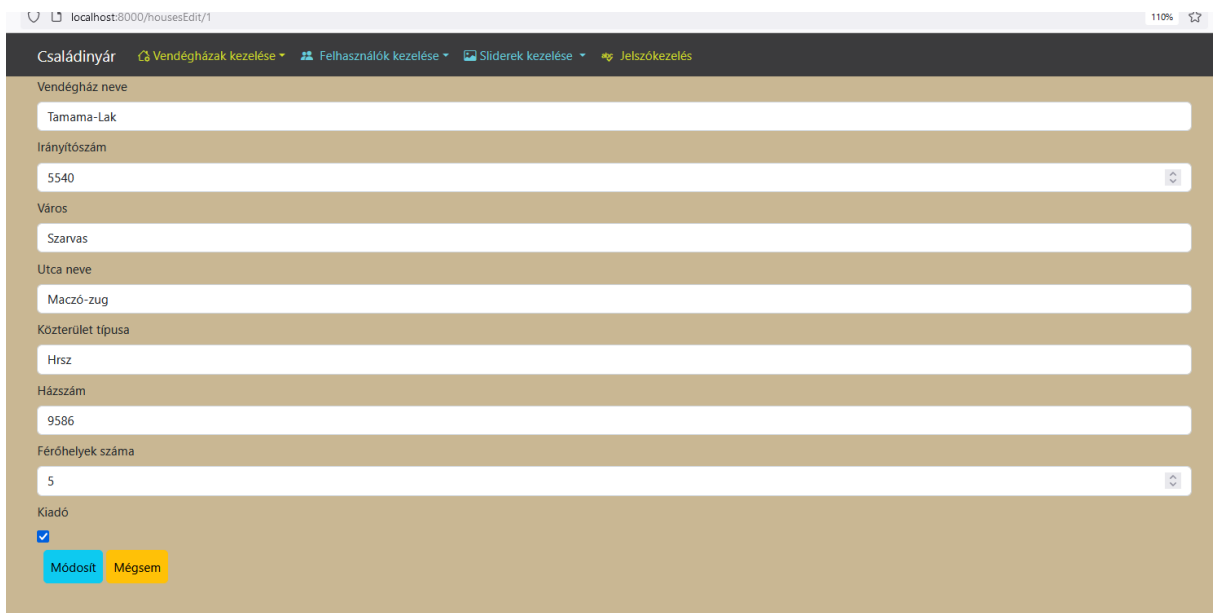
3.1.1 Vendégházak kezelés menüpont / *houses*



ID	Vendégház neve	Címe	Férőhelyek száma	Tulajdonos	Kiadó	Műveletek
1	Tamama-Lak	5540 Szarvas Maczó-zug Hrsz 9586	5	Pribelszki György	Igen	Módosít Töröl
3	Napsugár Apartman	5540 Szarvas Arborétumi üdülősor 54	10		Igen	Módosít Töröl
4	Gödörök Gyöngye A	5561 Békésszentandrás Siratói üdülősor 674	10	Kovács László	Igen	Módosít Töröl
5	Toscana Vendégház	5540 Szarvas Arborétumi üdülősor 115	12		Igen	Módosít Töröl
6	Levendula	5561 Békésszentandrás Kenderligeti üdülősor 121	8	Pribelszki György	Igen	Módosít Töröl
7	Pipacsos	5561 Békésszentandrás Siratói üdülősor 41	6		Igen	Módosít Töröl
10	Pusztá Gyöngye	5540 Szarvas Tanya II. külkerület 112	12	Pribelszki György	Igen	Módosít Töröl

[Új vendégház felvitele](#)

A Vendégház módosításnál a Módosít gombra kattintva betölti a módosítani kívánt ház adatait: */housesEdit/ id*



Vendégház neve
Tamama-Lak

Irányítószám
5540

Város
Szarvas

Utca neve
Maczó-zug

Közterület típusa
Hrsz

Házsám
9586

Férőhelyek száma
5

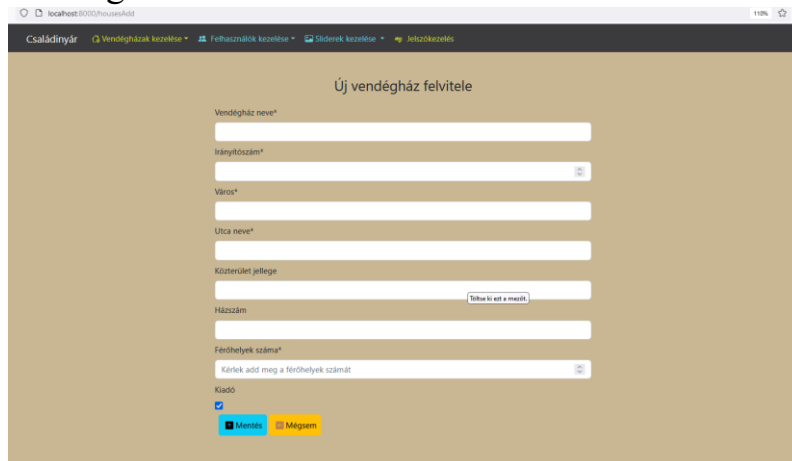
Kiadó
☒

[Módosít](#) [Mégsem](#)

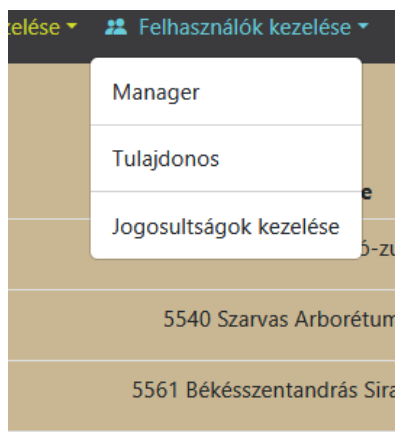
A Módosít gomb lemodosítja a kért adatot.

A Mégsem gombra kattintva visszavigál a Vendégházak listájára. A Vendégházak listázó oldalán az Új vendégház hozzáadásával az alábbi képernyőn van lehetőségünk elmenteni az új adatot, vagy visszlépni a listázott vendégházak oldalára.

Vendégház felvitele: /housesAdd



3.1.2 Felhasználók kezelés menüpont

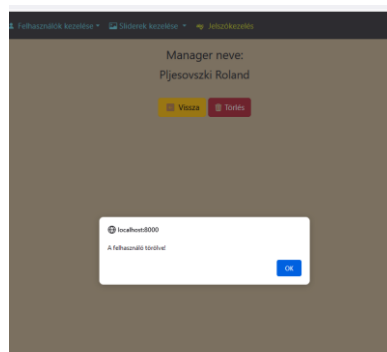


A felhasználó kezelése menüpont alatt található a tulajdonosok, a manager-ek, a tulajdonosok és a titulusokhoz tartozó felület. Mindhárom menüpontnál lehetőség van a módosításra, törlésre, vagy új adat létrehozására.

3.1.3 A manager oldal /manager

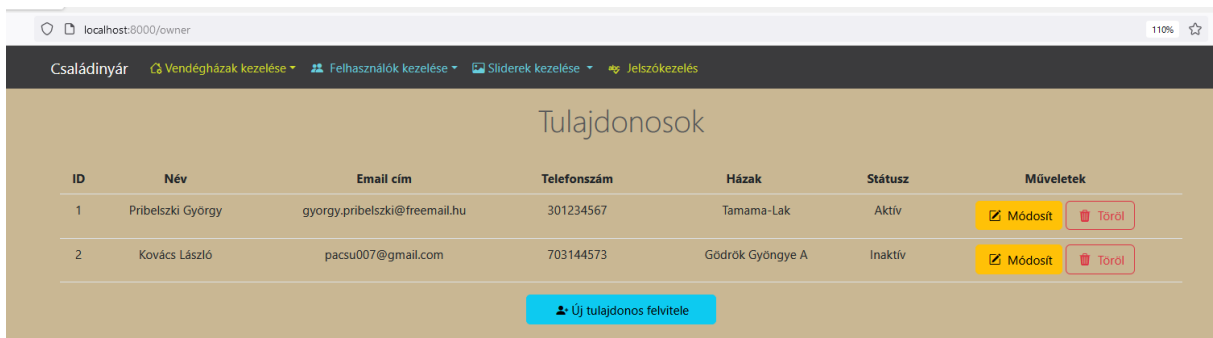
A manager oldalon a vendégház oldalhoz hasonlóan lehetőség van az adatok felvitelére, módosítására és törlésére.

Manager törlésénél betölti az adott manager nevét, a törlés gombra kattintva egy felugró ablak jelenik meg, mely közli a felhasználó törölve lett. Ezt a felugró ablakot a Javascript egyik függvényével oldottam meg. Ha a törlés után visszamegyünk a lista oldalára, a kitörölt felhasználó már nem jelenik meg. A listában ez a felhasználó már nem szerepel. A vissza gombnál visszavigál a managerek listájára.



3.1.4 Tulajdonos oldal / owner

A Tulajdonos oldal ugyanúgy működik, mint a manager oldal, a lista oldalon van lehetőség a módosításra törlésre, vagy új tulajdonos felvitelére is.



ID	Név	Email cím	Telefonszám	Házak	Státusz	Műveletek
1	Pribelszki György	gyorgy.pribelszki@freemail.hu	301234567	Tamama-Lak	Aktív	Módosít Töröl
2	Kovács László	pacsu007@gmail.com	703144573	Godrók Gyöngye A	Inaktív	Módosít Töröl

[+ Új tulajdonos felvitele](#)

Új tulajdonos felvitele / ownerAdd:



Új tulajdonos felvitele

Vezetéknév*

Keresztnév*

Email cím*

Kérlek add meg az email címed

Telefonszám*

Formátum: 123456789

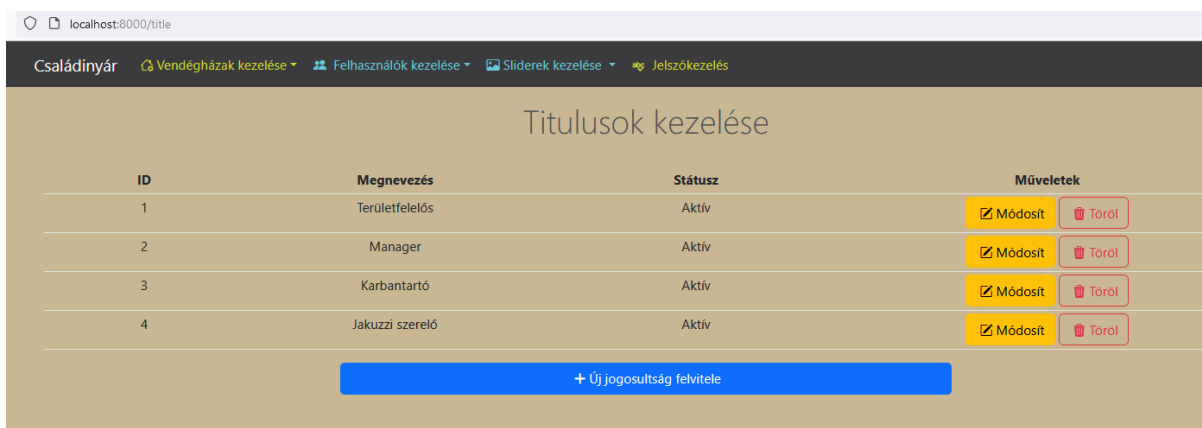
Státusz

☒

[Mentés](#) [Mégsem](#)

3.1.5 A jogosultságok kezelése / title

Itt is lehetőség van minden adat módosítására, vagy törlésére, vagy egy új titulus felvitelére



ID	Megnevezés	Státusz	Műveletek
1	Területfelelős	Aktív	Módosít Töröl
2	Manager	Aktív	Módosít Töröl
3	Karbantartó	Aktív	Módosít Töröl
4	Jakuzzi szerelő	Aktív	Módosít Töröl

[+ Új jogosultság felvitele](#)

3.1.6 Sliderek kezelése / slider

A sliderek kezelésénél is lehetőség van új slider felvitelére, annak módosítására, vagy akár a törlésére.

Képváltók					
ID	Név	Cím	Kód	Állapot	Műveletek
8	Slider01	Slider01	slider	Aktív	<input checked="" type="checkbox"/> Módosít <input type="checkbox"/> Töröl
10	nemtudom	Belső_tér	fooldal	Aktív	<input checked="" type="checkbox"/> Módosít <input type="checkbox"/> Töröl
11	ujramegint	Napozógyak	bszta	Aktív	<input checked="" type="checkbox"/> Módosít <input type="checkbox"/> Töröl
12	ujraujra	Csigalépcső	bszta	Aktív	<input checked="" type="checkbox"/> Módosít <input type="checkbox"/> Töröl
13	Minimagyarszág	miniMagyarország	minimagyarország	Aktív	<input checked="" type="checkbox"/> Módosít <input type="checkbox"/> Töröl
14	Nádaházam	Nádaházam	bszta	Aktív	<input checked="" type="checkbox"/> Módosít <input type="checkbox"/> Töröl
15	Arborétum	Arborétum	arboretum	Aktív	<input checked="" type="checkbox"/> Módosít <input type="checkbox"/> Töröl

Sliderek hozzáadása / sliderAdd

/sliderAdd

Vendégházak kezelése ▾ Felhasználók kezelése ▾ Sliderek kezelése ▾ Jelszókezelés

Slider feltöltés

Név*

Adja meg a fájlt:

Tallózás... Nincs kijelölve fájl.

Kód*

Leírás

Cím*

☒ Aktív

3.1.7 Jelszó kezelése

A jelszókezelésnél csak módosításra van lehetőség, törölni nem lehet!

Családnyár						
Vendégházak kezelése						
Felhasználók kezelése						
Sliderek kezelése						
Jelszókezelés						
Jelszókezelés						
ID	Név	Beosztás	Felhasználónév	Email	Állapot	Műveletek
1	Kovács László	Manager	pacsu77	pacsu007@gmail.com	Aktív	<input checked="" type="checkbox"/> Módosít
5	Fulajtár Mónika	Manager	mnika9	fulajtarmonika@gmail.com	Inaktív	<input checked="" type="checkbox"/> Módosít

3.2 A látogatók oldal

3.2.1 A látogatói oldal kialakítása

A látogatói oldalt a CSS és BOOSTRAP programok segítségével alakítottam ki.

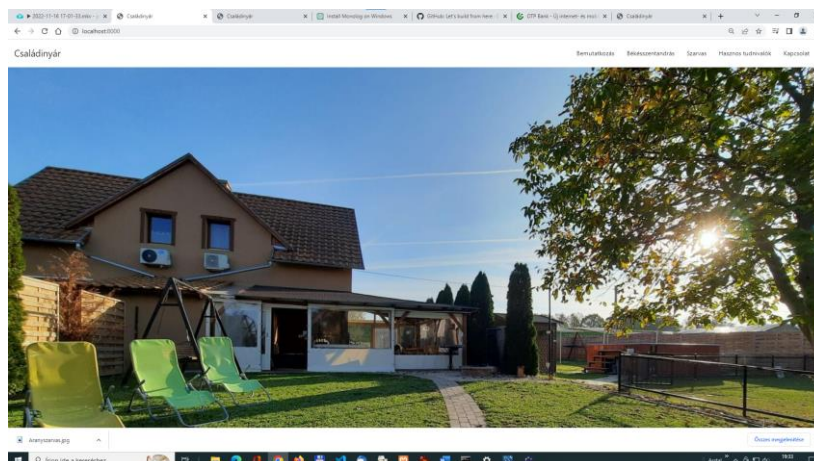
Igyekeztem figyelni a responsivitásra, az átláthatóságra, és az oldal könnyű kezelhetőségére. A látogatók a menüben választhatnak a két helyiség közül, vagy akár rögtön az akciós napokra is navigálhatnak.

A látogatók számára a Vendégházak közötti választás, a programok kínálata, a hasznos információk lennének láthatóak. A későbbiekben megvalósulna a hírlevélre való feliratkozás, az esetleges online foglalási rendszer

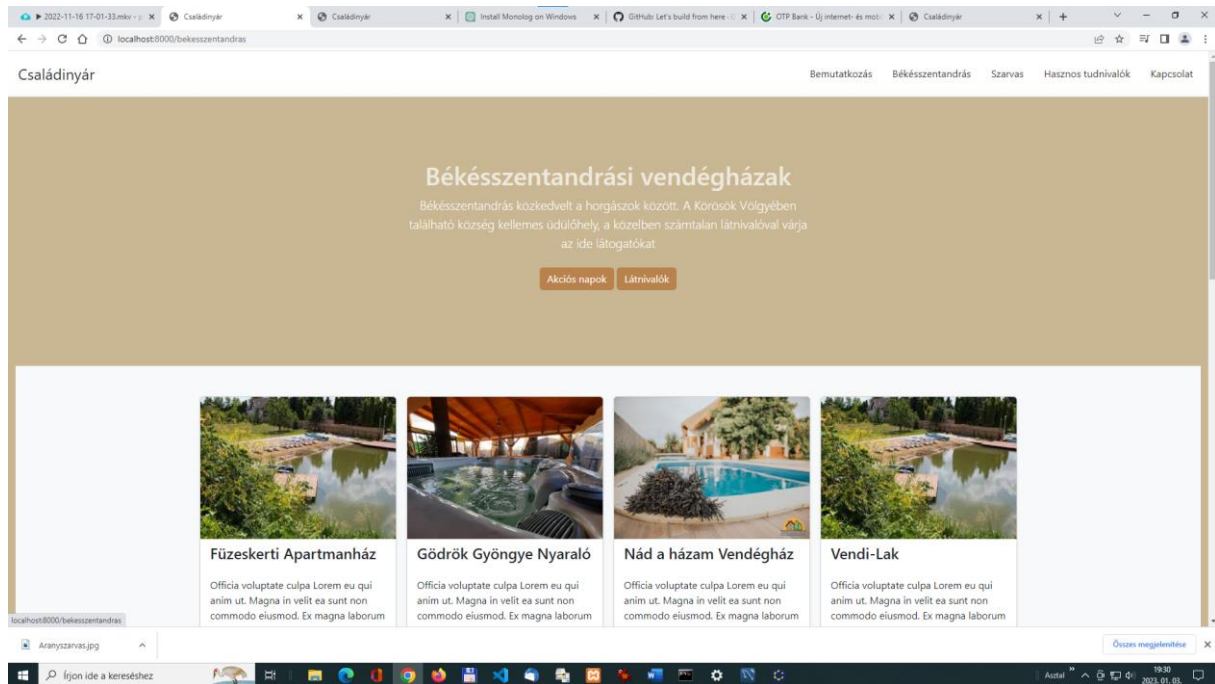
Az oldal adatainak feltöltését több személy is végezhetné. A nyaraló tulajdonosai előzetes regisztráció után feltölthetnének képeket, egyéb információkat a saját vendégházaihoz. A karbantartók és a házzal kapcsolatos tennivalókért felelősök megjegyzéseket fűzhetnének az adott vendégházhoz. A managerek pedig minden információt szerkeszthetnének.

3.2.2 Kezdőoldal / <http://localhost:8000/>

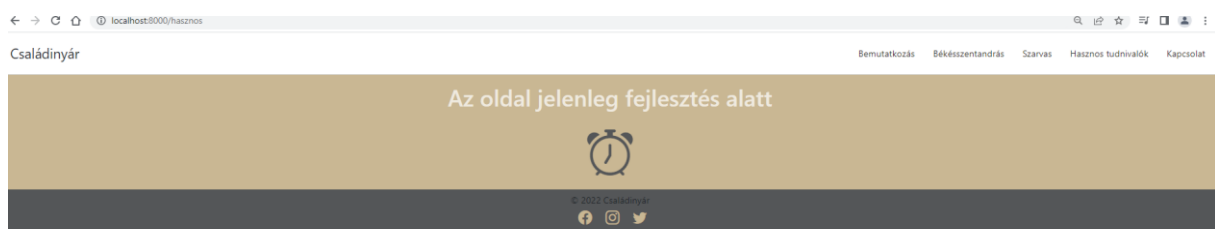
A kezdőoldal sliderekkel indul, és egy rövid bemutatkozó szöveg található az alján. A sliderek folyamatosan változnának, ez akciós napoktól vagy esetleg eseményektől tenném függővé.



A helyiségek oldalain a Vendégházakról egy kép, illetve egy rövid leírás található, a gombra kattintva jelenleg a vendégház saját honlapjára navigálhatunk. Az általános szöveg alatt található 'Látnivalók' gomb ugyanezen az oldalon tartja a látogatót, míg az 'Akciónapok' átnavigál egy másik oldalra.



A hasznos tudnivalók fejlesztés alatt van, ennek a képernyőjén az alábbi látható:



A kapcsolat fülön az adatok a contact táblából érkeznek, ide tervezem a kapcsolatfelvételi űrlap megvalósítását.

4 TESZTELÉS

4.1.1 Az oldal tesztelése

Az oldal tesztelését minden egyes funkció létrehozásánál vagy változtatásánál teszteltem, a megjelenítés változtatásánál is folyamatosan frissítettem az oldalt.

4.1.2 Jelszó tesztelése

A jelszó módosításának tesztelését a <https://md5decrypt.net/en/Sha1/> oldal segítségével végeztem el az alábbi módon:

Az adatbázisban kiválasztottam egy jelszót, majd azt az admin felület Jelszókezelő menüjében megváltoztattam, ebben az esetben Kovács László Nem123 jelszavát.

„Igen456” -ra módosítottam le, így létrejött egy új titkosított jelszó:
c416fdad7a712710f399444b1a590f830063bd03

id	lastname	firstname	title	username	password	email	phone_number	status	deleted
1	Kovács	László	2	pacsu77	Nem123	pacsu007@gmail.com	703144573	1	0
5	Fulajtár	Mónika	2	mnika9	da39a3ee5e6b4b0d3255bfef95601890afd80709	fulajtarmonika@gmail.com	703110744	0	0
9	Nagy	Magdolna	5	magdus	9484b46d5b4a760acb900627c21621dd94b5c799	magdalena@gmail.com	0	1	1
11	Pjjesovszki	Roland	3	prooland	8abccda2dba9a5c5c674e659333828582122c5f56	prolika@gmail.com	306541240	0	1
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



A fent említett oldal segítségével legeneráltam az új jelszót egy Sha1 jelszóra, majd a két jelszót összehasonlítottam:

Adatbázisban lévő jelszó:

id	lastname	firstname	title	username	password	email	phone_number	status	deleted
1	Kovács	László	2	pacsu77	c416fdad7a712710f399444b1a590f830063bd03	pacsu007@gmail.com	703144573	1	0
5	Fulajtár	Mónika	2	mnika9	8ffb5db5b8ad6e28cd16f8bea01dac07dd86f7da	fulajtarmonika@gmail.com	703110744	0	0
9	Nagy	Magdolna	5	magdus	9484b46d5b4a760acb900627c21621dd94b5c799	magdalena@gmail.com	0	1	1
11	Pjjesovszki	Roland	3	prooland	8abccda2dba9a5c5c674e659333828582122c5f56	prolika@gmail.com	306541240	0	1
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

c416fdad7a712710f399444b1a590f830063bd03

Legenerált jelszó:

Sha1(Igen456) = c416fdad7a712710f399444b1a590f830063bd03

Jól látszik, hogy a két jelszó megegyezik, azaz a jelszómódosítás sikeres.

4.1.3 Responsivitás ellenőrzése

A responsivitást legegyszerűbben az F12 gomb segítségével, a fejlesztői felületen lehet tesztelni, én is ezt használtam.

5 A PROGRAM TOVÁBBI FEJLESZTÉSÉNEK TERVE

A program teljes befejezésével szeretném megvalósítani a vendégházak működtetésével járó információk karbantartását, a költségek és elvégzett feladatok nyomon követését, a vendégek adatait.

A további fejlesztés során kívánom megvalósítani az alábbi feladatokat:

- a felhasználók jogosultságainak a kialakítása
- a Log rendszer naplózása a felhasználókra
- az adatbázis felöltése
- a fejlesztés alatt álló oldalak befejezése
- kapcsolati űrlap kialakítása
- online időpontfoglalások

6 Mellékletek

6.1 Az adatbázis modell diagramja

6.2 Az _install script file.

1. számú melléklet

