Szakképesítés megnevezése:  
SZOFTVERFEJLESZTŐ ÉS -TESZTELŐ TECHNIKUS SZAKMA  
AZONOSÍTÓ SZÁMA: 5 0613 12 03

Kreatív Ötletcentrum Webshop

****

**Készítette:** Nagy Gábor

**Csapattagok:** Takáts Kristóf László, Domokos Dávid

Budapest, 2022.

**Nyilatkozat**

Alulírott **Nagy Gábor** kijelentem, hogy ez a vizsgaremek saját tudásom, önálló munkám terméke.

A vizsgaremek közös részeit Takáts Kristóf László, Domokos Dávid és Nagy Gábor 2/14S tanulói készítettük, ezeket pontosan jelöltük.

Aláírás Aláírás Aláírás  
 Nagy Gábor Takáts Kristóf László Domokos Dávid

Budapest, 2022.

**Tartalomjegyzék**

Tartalom

[1 Bevezetés 4](#_Toc101736563)

[1.1 Témaválasztás 4](#_Toc101736564)

[1.2 A projekt folyamata 5](#_Toc101736565)

[2 Megoldandó feladatok 5](#_Toc101736566)

[2.1 A fejlesztés előtt kitűzött célok 5](#_Toc101736567)

[3 Fejlesztői dokumentáció 7](#_Toc101736568)

[3.1 Alkalmazott fejlesztői eszközök 7](#_Toc101736569)

[3.2 Felületterv 9](#_Toc101736570)

[3.3 Adatmodell leírása 13](#_Toc101736571)

[4 4Részletes feladatspecifikáció, algoritmusok 16](#_Toc101736572)

[4.1 A program lényeges függvényeinek, eljárásainak, az osztályok metódusainak a specifikációja 16](#_Toc101736573)

[5 Tesztelési Dokumentáció 21](#_Toc101736574)

[5.1 Api végpontok tesztelése: 21](#_Toc101736575)

[6 A program ismert hibái, továbbfejlesztési lehetőségek 25](#_Toc101736576)

[6.1 A program ismert hibái 25](#_Toc101736577)

[6.2 Továbbfejlesztési lehetőségek 25](#_Toc101736578)

[7 Felhasználói dokumentáció 26](#_Toc101736579)

[7.1 Rendszerkövetelmények 26](#_Toc101736580)

[7.2 A program telepítésének és konfigurálásának leírása 27](#_Toc101736581)

[8 A program használatának részletes dokumentálása 50](#_Toc101736582)

# Bevezetés

## Témaválasztás

Az általam és a csapatom által választott téma egy kreatív termékeket, alapanyagokat forgalmazó üzlet komplett webshopjának elkészítése. A témaválasztás előtt nagyjából három hónappal megkeresést kaptam az üzlettől, hogy elvállalnám-e ennek a boltnak a webshop készítését. Mivel egyedül nem sikerülhetett volna ezért először visszautasítottam az ajánlatot. Később a projekt óra keretei között megtudtuk, hogy csapatunknak készítenie kell egy komplex alkalmazást (RestApi technológiát használva) így egyből a fent említett ajánlat jutott eszembe. A csapattagokkal és az ügyféllel konzultálva, végül elvállaltuk a Kreatív Ötletcentrum üzletnek a webshop készítését. A vizsgaremek elvárásai alapján egy weblapot, egy asztali alkalmazást és egy telefonos alkalmazás készítését kell teljesítenie a csapatunknak a sikeres vizsgához. A bolt tulajdonosával konzultálva felvetettük neki az ötletet egy asztali raktárkezelő alkalmazás, és egy Android telefonos alkalmazás elkészítésére. Az ügyfél természetesen támogatta ezt az ötletet, az áruforgalom nyomon követése érdekében.

A tervezés alatt felosztottuk a frontend alkalmazásokat, mindig is szerettem az asztali programozási technológiákat, így az én részem az asztali alkalmazás lett, mivel az asztali alkalmazásban nagyobb hangsúlyt kap a programozás rész, mintsem a dizájn így nagy örömmel tölt el, hogy ez a feladat került hozzám. A fejlesztés során valós adatokkal, termékekkel dolgozunk, így nagyban megnöveli a vizsgamunka komolyságát. Valós adatokkal dolgozni részben könnyebb, részben pedig sok nehézség is van mögötte, például: a termék adatait adatbázisba rendezni, a termékek lefotózása, photoshopolás, nagy adatbázis kezelése. Ugyanakkor könnyebbség, hogy megkímél a rengeteg adat, képgyűjtéstől, feldolgozástól. Ez a „munka” tekinthető egy megrendelésnek is mind az ügyfél mind az iskola részéről.

## A projekt folyamata

Az alábbi folyamatábra szemlélteti a projekt folyamatát a kezdetektől az átadásig.



1. *ábra – folyamatábra*

# Megoldandó feladatok

## A fejlesztés előtt kitűzött célok

A fejlesztés megkezdése előtt az ügyféllel és a tanárokkal is konzultáltunk az elvárások iránt. Rengeteg saját, illetve nagyon sok ötlet merült fel a csapaton belül is, hogy milyen irányban induljunk el, milyen funkciói legyenek a webes, asztali, telefonos alkalmazásban. Az alap cél kitűzése, egy már meglévő boltnak komplex alkalmazáscsomag készítése, ami képes egy weboldalt, egy asztali alkalmazást és egy telefonos alkalmazást kezelni. A projekt fő célja egy webshop elkészítése, amelyen a vásárlók tudnak böngészni a termékek között, megrendelést leadni, felhasználó fiókot létrehozni.

A weboldal mellett elengedhetetlen egy alkalmazás, ami képes a termékek kezelésére, ezt egy asztali alkalmazásban valósítom meg. Ez az alkalmazás kizárólag az üzlet személyzet számára készül. Az alkalmazás célja, hogy a webshopon kívül is tudják a bolt értékesítői, tulajdonosai kezelni a boltban eladott termékek számát, a raktáron megmaradt termékeket. Emellett az asztali alkalmazásban a bolt tulajdonosa fel tud venni új termékeket, ha bővülne a kínálat, vagy ha változtatni szeretne az áron, mennyiségen vagy a kategóriába soroláson akkor ezt is meg teheti az alkalmazáson belül, a már nem forgalmazott termékeket törölni is tudja, ha szeretné. A termékek megtalálása érdekében egy keresés funkció is jelen van az alkalmazásban.

A biztonság érdekében az asztali alkalmazásba egy autentikáció (bejelentkezés) is bekerült. Bejelentkezés hiányában egy rossz szándékú ember percek alatt romba tudja dönteni az üzlet dolgozóinak hosszas munkáját, de a bejelentkezésnek hála, csak a saját, az adatbázisban egyedileg adminisztrátori jogosultsággal regisztrált felhasználók tudnak belépni az asztali alkalmazásba.

Az asztali alkalmazás maximálisan felhasználóbarát, egyszerű kinézetével, átláthatóságával nem okoz gondot egy idősebb dolgozónak sem, ez külön elvárás volt felénk, hogy az idősebb korosztály se zavarodjon bele a villogó, túlszínezett, értelmetlen funkciók közé, ezért egy egyszerűen kezelhető, átlátható, minimalista de mégis esztétikus alkalmazás elkészítése mellett döntöttem.

A webböngészőből megnyitható alkalmazás mellett a projekt társam készít egy applikációt, amelynek célja, hogy egy Androidos okostelefonra applikáció formájában letölthető legyen a weboldal kinézetére épülő blog alkalmazás, ahol a felhasználók egymással megoszthatják az általuk elkészített kreatív díszeket, termékeket. Ez az alkalmazás segít népszerűsíteni az üzletet, az emberek az ott vásárolt alapanyagokból kreatívkodhatnak, ötletet meríthetnek.

A webes alkalmazásban a vásárlás, kosárba rakás regisztrációhoz kötött míg a termékek böngészése regisztráció nélkül is elérhető. A telefonos alkalmazás használata regisztrációhoz kötött és egy bemutató viszi végig a felhasználót a funkciókon a felhasználói élmény fokozása érdekében.

# Fejlesztői dokumentáció

## Alkalmazott fejlesztői eszközök

Leíró és programozási nyelvek

* PHP
* JAVA
* CSS
* FXML

Felhasznált keretrendszerek

* Laravel (https://laravel.com/)
* JavaPackager (<https://github.com/fvarrui/JavaPackager>)

Felhasznált fejlesztői környezetek

* Microsoft Visual Studio Code
* IntelliJ Idea

Felhasznált adatbáziskezelő rendszer

* MySQL (phpMyAdmin)

Verziókezelő rendszer

* Git
* GitHub

Bővítmények

* Thunder Client
* Live Server
* Gson
* JavaFx
* Scene Builder

Lokális webszerver

* Apache
* MySQL

Kép-, videó-, szövegszerkesztő, egyéb használt programok

* Xampp
* Notepad++
* Paint
* Adobe Photoshop
* Adobe XD
* Microsoft Word
* Microsoft Excel
* Microsoft PowerPoint
* Opera GX

## Felületterv

Az alább megtalálható táblázat bemutatja az alkalmazás funkcióit.

|  |  |
| --- | --- |
| **Funkció** | **Részletes leírás** |
| **Bejelentkezés** | Belépés az alkalmazásba, kizárólag azok számára, akik admin jogosultsággal rendelkeznek. |
| **Hozzáadás** | Termék hozzáadása az adatbázishoz a szükséges mezők kitöltésével. |
| **Módosítás** | Az adatbázisban lévő termék módosítása, a szükséges mezők kitöltésével. |
| **Törlés** | A kiválasztott termék törlése az adatbázisból. |
| **Keresés** | Az adatbázisban megtalálható termékek között keresés „live keresés” funkcióval. |
| **Rendelések** | A webshopon rendelt megrendelések listázása. |
| **Szűrés funkció** | Ár, kategória szerint tud szűrni. |
| **Felhasználók kezelése** | Felhasználók listázása, törlése, tiltása. |
| **Rendelések kezelése** | A weboldalon leadott rendelések kezelése, listázása az asztali alkalmazásban. |

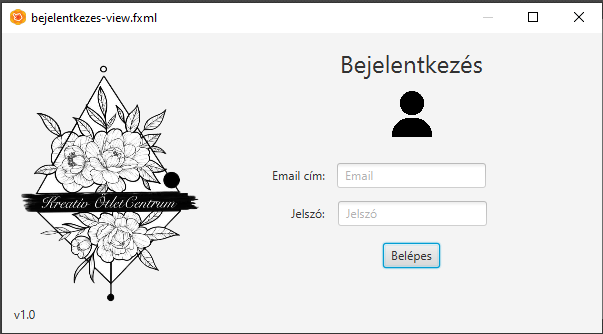
A tervezési fázisban fontos volt, hogy összegyűjtsem a programban megvalósítani kívánt funkciókat, és a jövőre nézve a továbbfejlesztési funkciókat. A kinézet összerakása előtt készítettem egy látványtervet a bejelentkezésről és a termékkezelő kinézetéről. Az egyszerűségre törekedtem, felesleges képpel, animációval nem raktam tele, ezáltal meggyorsítva a program futását.

**Bejelentkezés látványterv**

****

1. *ábra – Bejelentkezés-terv*

**Bejelentkezés végleges kinézet**

****

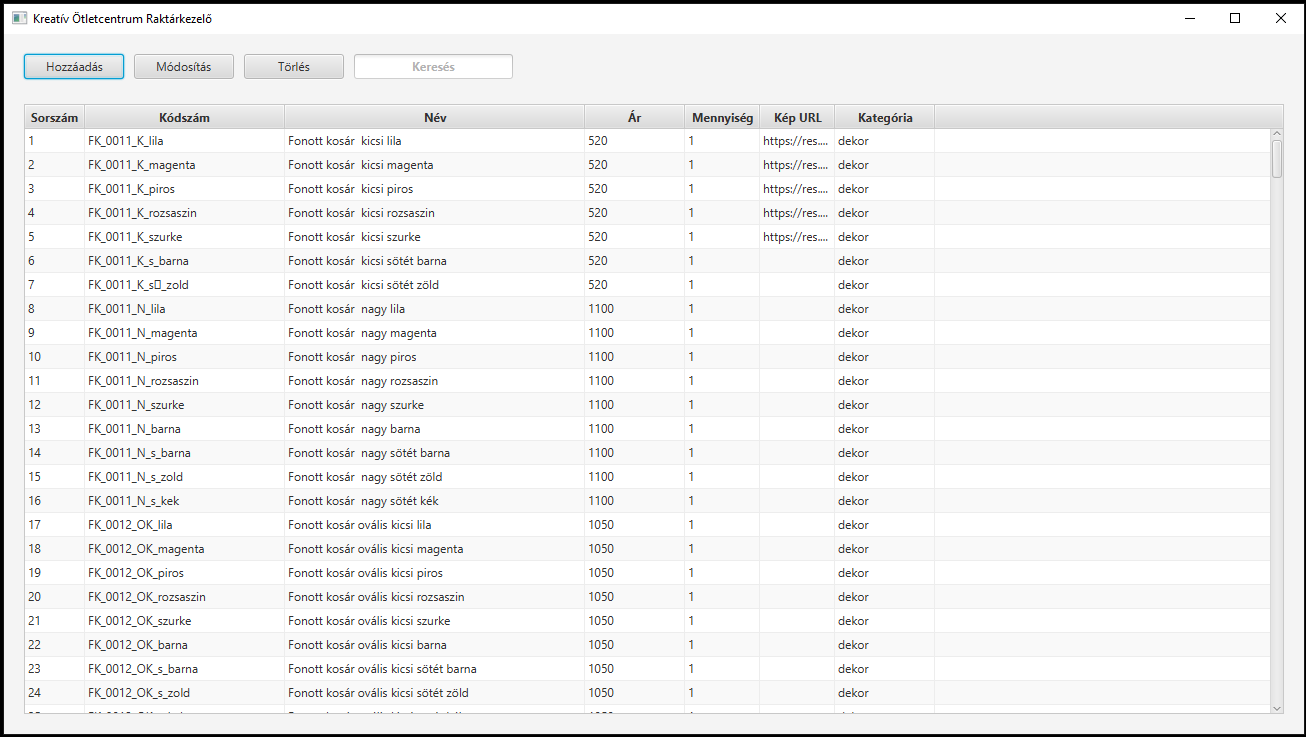
1. *ábra – Bejelentkezés-végleges*

**Termékek listázása, műveletek látványterv**

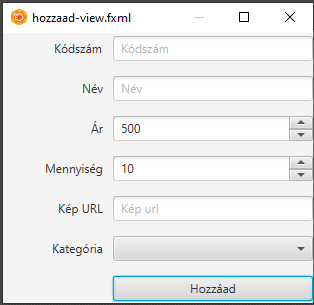
****

1. *ábra – Termékek, műveletek-terv*

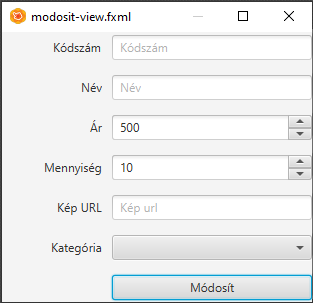
**Termékek listázása, műveletek végleges kinézet**

****

1. *ábra – Termékek, műveletek-végleges*

**Hozzáadás látványterv Hozzáadás végleges kinézet**

1. *ábra – Termék hozzáadása – terv 7. ábra – Termék hozzáadása - végleges*

**Módosítás látványterv Módosítás végleges kinézet**

1. *ábra – Termék módosítása – terv**9. ábra – Termék módosítása – végleges*

## Adatmodell leírása

webshop

1. *ábra – Adatmodell kinézete*

Az adatbázisunk webshop néven lett létrehozva UFT\_8\_MB4\_hungarian\_ci kódolással. Az adatbázis **négy** táblát tartalmaz. Az **users** (felhasználók) tábla tartalmazza a weboldalra regisztráló felhasználók adatait. Minden felhasználó egy azonosítóval van ellátva, másnéven **id**, ami egy egész típusú szám (int), ez egy egyedi azonosító, ami a táblázat elsődleges kulcsa, és automatikusan növekszik a felhasználók számától függően (AUTO\_INCREMENT). Továbbá tartalmaz még egy **name** mezőt (név), ahol a felhasználó vezeték és keresztneve szerepel szöveges típusban. Az **email** mező tartalmazza a regisztráció során megadott email címet szöveges formában, ezt követi az **email\_verified\_at**, ami automatikusan dátum formában (timestamp) adja vissza a regisztráció megerősítésének időpontját, mivel regisztráció után a rendszer egy megerősítő levelet küld a regisztrálni kívánt címre. Ha a mező null értéket ad vissza, a felhasználó nem tud bejelentkezni, mert még nem erősítette meg a regisztrációt. A megerősítés után az **email\_verified\_at** dátum megjelenik. A **password** (jelszó) mező a fiókhoz tartozű jelszót tárolja titkosított szöveges formában (hashelve). Ezen kívül a jogosultság kezeléshez létre lett hoztunk egy **permission** mezőt, aminek az alapértelmezett értéke 0 (**felhasznalo**), ez manuálisan átállítható 1-re, ami az **admin** jogosultságot jelenti.

A **kosar** táblában található a bejelentkezett felhasználó által kosárba helyezett termékek listája, amit le tud menteni a fiókjához, később pedig onnan tudja folytatni a vásárlást, ahol abbahagyta. Tartalmaz egy **id** egész szám típusú egyedi azonosítót, ez a kosárnak az azonosítója. A **mennyiseg** egy egész típusú oszlop, ami a kosárba helyezett termékek mennyiségét adja vissza. A **kosarba\_helyezes**, ami egy timestamp típusú mező, dátum és idő formájában adja vissza a kosárba helyezés időpontját. Ezen kívül a tábla tartalmaz egy **rendeles\_id** egész szám típusú egyedi azonosítót, ami megegyezik a rendeles tábla azonosítójával. A **termek\_id** egy egész típusú azonosító, ami a kosárba rakott termék azonosítóját tartalmazza, ez megegyezik a termekek táblában szereplő id-vel.

A harmadik tábla a **termekek**, ami tartalmazza a honlapon megjelenő termékek adatait. Minden termék el van látva egy **id**, egész szám típusú egyedi azonosítóval. Továbbá el van látva egy **kodszam**-al, ami az id oszloppal ellentétben szöveges típusú, betűkből és számokból áll, ez a termékekhez tartozó egyedi kódszám. A **name** egy szöveges típusú mező, a termék nevét tartalmazza. A **price** mező egy egész szám típusú oszlop, ahol az összes termék ára fel van sorolva. A **quantity** oszlop szintén egy egész típusú oszlop, itt található a termékek elérhető mennyisége. Amennyiben a mennyiség nulla, a weboldalon megjelenik a „Jelenleg nem kapható” üzenet a terméknél, ellenkező esetben az elérhető mennyiség látható. Az **url** mező egy szöveg típusú mező, itt található a termékekhez kapcsolódó fotó linkje, ez jelenik meg a weboldalon a termékeknél. A **kategoria** mező egy szöveg típusú oszlop, ahol a termékek kategóriája van felsorolva, ennek az oszlopnak a segítségével tettük lehetővé a termékek kategóriákra bontását. A **created\_at** oszlop egy timestamp, dátum típusú mező, ahol a termék adatbázishoz adásának időpontját jeleníti meg, ezt követi az **updated\_at** mező, ami szintén dátum formában adja vissza a termék adatbázisban történő módosításának időpontját.

|  |  |
| --- | --- |
| Termekek tábla | |
| Mező neve | Típusa, hossza, jellemzői |
| id | bigint (20) AUTO\_INCREMENT PRIMARY KEY |
| kodszam | varchar (255) |
| name | varchar (255) |
| price | int (11) |
| quantity | int (11) |
| url | varchar (255) |
| kategoria | varchar (255) |
| created\_at | timestamp  DEFAULT: NULL |
| updated\_at | timestamp  DEFAULT: NULL |

A **rendeles** tábla tartalmazza az összes rendelés adatait. Minden rendeléshez tartozik egy **id**, egész szám típusú egyedi azonosító. Az **user\_id** mező egy egész szám típusú azonosító, ez a mező tartalmazza a rendeléshez kötött felhasználó azonosítóját. Az utolsó oszlop pedig az **idopont**, ami egy timestamp típusú mező, a rendelés leadásának időpontját mutatja.

|  |  |
| --- | --- |
| Users tábla | |
| Mező neve | Típusa, hossza, jellemzői |
| id | bigint (20) AUTO\_INCREMENT PRIMARY KEY |
| name | varchar (255) |
| email | varchar (255) |
| email\_verified\_at | timestamp  DEFAULT: NULL |
| password | varchar (255) |
| permission | tinyint (4)  DEFAULT: 0 |
| remember\_token | varchar (100) DEFAULT: NULL |
| created\_at | timestamp  DEFAULT: NULL |
| updated\_at | timestamp  DEFAULT: NULL |

|  |  |
| --- | --- |
| Kosar tábla | |
| *Mező neve* | Típusa, hossza, jellemzői |
| *id* | bigint (20) AUTO\_INCREMENT PRIMARY KEY |
| *mennyiseg* | *int (11)* |
| *kosarba\_helyezes* | *timestamp current timestamp()* |
| *rendeles\_id* | *bigint (20)* |
| *termekek\_id* | *bigint (20)* |

|  |  |
| --- | --- |
| *Rendeles tabla* | |
| *Mező neve* | *Típusa, hossza, jellemzői* |
| *id* | bigint (20) AUTO\_INCREMENT PRIMARY KEY |
| *user\_id* | bigint (20)  FOREIGN KEY |
| *idopont* | *timestamp current\_timestamp ()* |

# 4Részletes feladatspecifikáció, algoritmusok

## 4.1 A program lényeges függvényeinek, eljárásainak, az osztályok metódusainak a specifikációja

Az asztali alkalmazás program a Laravel backend keretrendszer és a Java nyelv, JavaFx segítségével lett létrehozva, az alkalmazás így grafikus felületű, így egy „laikusnak” is könnyen kezelhető.

A program felépítése alapjában a „clean code” elveire épül, a többször használt metódusok kiszervezése egy külön metódusba, később így a függvény nevét meghívva átláthatóságot, könnyen értelmezhetőséget sugall a kód. A „Java” nyelvnek szépsége, hogy az alapja az OOP (Objektum Orientált Programozás), ami számos lehetőséget kínál a fejlesztő számára.

Lényegesebb függvények:

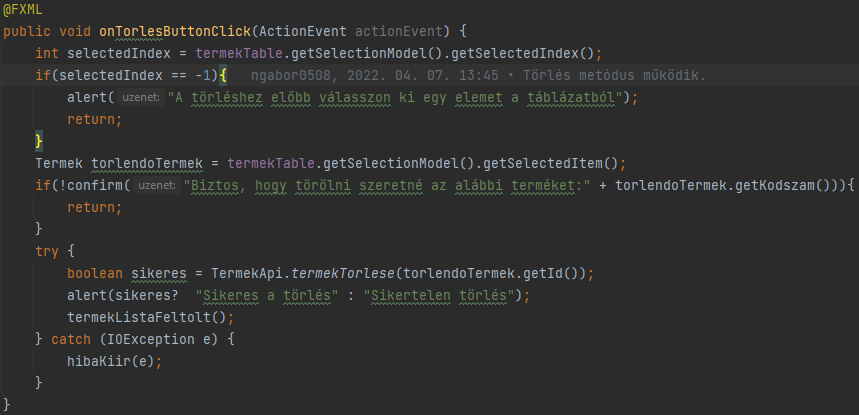
Backend:

* Migrációk:
  + **Termekek, kosar, rendeles, users:**
* A migrációk segítségével tudjuk létrehozni a táblákat az üres adatbázisban. Ez a metódus mindegyik migrációs osztályban megtalálható. **(termekek, users, kosar, rendeles):**

**up ()**

* + Az adatbázisban a adott tábla és az oszlopainak létrehozásáért felelős metódus. A metódusból kapja meg a táblázat, hogy melyik oszlop milyen adat típussal rendelkezzen.
* Seederek:
  + TermékekSeeder - run()
    - Egy script ami a .csv kiterjesztésű fájlt feltölti egy adatbázisban lévő táblába.
* UserSeeder – run()
  + Az asztali alkalmazás zavartalan működése érdekében, a metódus egy „admin” és egy „általános” felhasználót seedel a users táblába.
* Api:
* AuthController – login ()
  + A bejelentkezésért felelős backend rész, ha a bejelentkezés hibás akkor a hibaüzenetet innen adja vissza a frontend alkalmazásnak. A bejelentkezés email, password alapján történik.
* TermekekController – index (), store (), show (), update (), destroy()
  + A listázást, hozzáadást, módosítást, törlést megvalósító metódusok találhatóak a TermekekController.php-ban a frontend ezeket a függvényeket hívja meg, a legtöbbnél id alapján.

Frontend:

* TermekApi– getTermekek(), termekHozzaadasa(), termekModosítasa(),  
  termekHozzaadasa()
  + A listázás, hozzáadás, módosítás, törlés metódusokat valósítja meg.
  + Az url alapján kéri le az adatokat Json formátumban, ezeket átalakítja, listába teszi, így később a listában a kezelhető adatok lesznek benne.
  + A böngészőben „/termekek” url megadásával érhető el az összes termek Json formátumban-.
* BejelentkezesApi – getBejelentkezes(), postBejelentkezes(),  
  getBejelentkezesAdatok()
  + A bejelentkezéshez szükséges adatokat kéri le az adatbázisból a backenden keresztül.
* ApiError getMessage()
* A backendben lévő hibaüzenetet adja vissza.
* Controller – alert()
  + Alertbox metódus, ahol alertboxot szeretnék használni, ott a függvény nevét meghívva, tudtunk új alertboxot létrehozni.
  + Egyedi figyelmeztetés megadása szükséges.
  + Az ablak egyedi ikonnal rendelkezik.
* Controller – confirm()
  + Megerősítést kérő művelet előtt ez a függvény van meghívva, ezzel tudjuk megerősíteni a módosítás, törlés műveletet.
  + Egyedi figyelmeztetés megadása szükséges.
  + Az ablak egyedi ikonnal rendelkezik.
* Controller – hibaKiir()
* A backend által kapott hibaüzenetet tudjuk kiíratni vele, ezt a művelet Excption esetén érdemes használni, ha a rendszer által dobott hibát szeretnénk kiíratni.
* Egyedi figyelmeztetést megadni nem kell.
* Az ablak egyedi ikonnal rendelkezik.
* Controller – ujAblak()
  + A metódus célja, hogy ennek meghívásával tudunk új ablakot létre hozni.
  + Ez a függvény három bemeneti paramétert vár meghíváskor, amikor meghívjuk akkor meg kell adni azt az fxml fájlt amit meg akarunk nyitni, az oldal címét, és az oldal méreteit.
  + Új oldal megnyitáskor minden esetben egyedi ikon helyettesíti az alapértelmezett ikont.
* MainController – termekListaFeltolt()
  + A TermekApi-ból a gettermekek() metódust meghívva egy listába feltölti az az adatbázisban található termékeket.
  + A grafikusan megjelenő táblázat feltöltésre kerül a listában található termékekkel.
  + Sikertelen feltöltés esetén a backend által átadott hibát adja vissza.
* MainController – keres()
  + A termékeket tartalmazó listát feltölti egy **Observable** listába, egy szűrt listába (**filteredList**) fogjuk elmenteni azokat az elemeket, ahol egyezést találtunk a keresett szó és a termékek tulajdonságai között.
  + A keresőmezőbe beírt szavakat, karaktereket átalakítjuk kis betűssé a pontos egyezés érdekében, csakúgy, mint a keresendő termékeket is.
  + Többágú elágazással megvizsgáljuk, hogy van e egyezés a kereső mezőben szereplő szó, karakterlánc és a listában található elemek bármelyike között.
  + Egyezés esetén a visszatérési érték „**true**” lesz, ha nincs egyezés akkor pedig „**false**” értékkel fog visszatérni az elágazás.
  + A találatokat (**filteredList-et**) átadjuk egy rendezett listának (**sortedList**).
  + A grafikusan megjelenő táblázat kiürítésre került, ezután feltöltjük a táblázatot a rendezett listával.
* MainController – szures()
  + A szűrés funkció hasonló működési elvre épül mint a keres() metódus.
  + Az adminnak lehetősége van kategóriákra szűrni a termékek között. A három kategória: **alap, dekor, kész,** ezek a kategóriák megjelennek egy Choicebox-ban, ahol választani tud, hogy melyikre szeretne rászűrni.
  + Ezen felül, ha mégis az összes terméket szeretné látni, az Összes termék kategóriára kattintva az eredeti listát fogja látni.
  + A termék listát, feltöltjük egy Observable listába.
  + Egy többágú elágazásban megnézzük, hogy éppen melyik kategóriára szeretne szűrni az **admin** jogosultsággal rendelkező felhasználó.
  + Az elágazáson belül egy ciklus segítségével végig megyünk a termékek kategória oszlopán.
  + A ciklus magon belül létrehozunk egy egyágú elágazást, ez figyeli az egyezést a listában lévő kategóriák között, ha egyezést talál hozzáadja az Observable listához.
  + A végén ezzel a listával töltjük fel a táblázatot.
  + Ha az Összes termék kategóriára kattintunk, úgy a termekListaFeltolt() metódust fogjuk meghívni, ez feltölti a listát az összes elemmel.
* MainController – onTorlesButtonClick()
  + A termék törlés metódus id alapján tud megvalósulni.
  + Létrehozunk egy helyi selectedIndex változót ami az éppen kiválasztott terméknek az id-jét adja vissza.
  + Egy egyágú elágazásban megnézzük, hogy az admin felhasználó választott e ki elemet, ha nem választott ki elemet akkor egy alert hibaüzenettel fog visszatérni a program.
  + Egy másik egy másik egyágú elágazással megnézzük, hogy ha választott ki elemet akkor melyik elemet választotta ki, és egy megerősítő (confirm) ablakot dobunk a felhasználónak.
  + Egy boolean változó létrehozásával megnézzük, hogy sikeres volt e a törlés.
  + Ha sikeres volt akkor egy visszajelző alert üzenetet írunk ki, ha sikertelen akkor pedig egy sikertelenségre utaló hibaüzenetet írunk ki a felhasználó számára.
  + A végén a termekListaFeltolt() metódussal feltöltjük a módosított listát.

# Tesztelési Dokumentáció

A tesztelési eljárás egy program készítési folyamatának elengedhetetlen része, itt lehetőségünk nyílik feltárni a még ismeretlen hibákat vagy sikeres tesztesetek után megbizonyosodni a program helyes működéséről. Ezt természetesen az adminisztráció ügyében dokumentálni is szükséges.

## Api végpontok tesztelése:

Helyi szerver kiszolgáló felépítése

1. XAMPP Control Panel megnyitása, helyi belső szimulált hálózat.
2. Az „Apache” és a „MySQL” modulok elindítása.
3. A XAMPP/htdocs mappában hozzunk létre egy új mappát „Teszt” néven.
4. Visual Studio Code fejlesztői környezet megnyitása, új ablak létrehozásához kattintson a menüsoron a File >> New Window menüpontra.
5. A repository klónozásához válasszuk a bal oldali sávban 3. lehetőséget a Source Control-t. Ezután két menüpont közül van lehetőségünk választani, a „Clone Repository” lehetőséget válasszuk.
6. Ezután a fenti keresőmezőbe illesszük be az alábbi repository url címét: [**https://github.com/ngabor0508/projekt\_webshop\_weblap.git**](https://github.com/ngabor0508/projekt_webshop_weblap.git)
7. A mappa kiválasztásánál, ha van lehetőségünk akkor a XAMPP/htdocs/Teszt mappát válasszuk ki.
8. Most nyissuk meg a Visual Studio-ban az általunk le klónozott mappát.
9. A XAMPP Control Panelt megnyitva a MySQL stop lehetőség mellett kattintsunk az „Admin” gombra. Ekkor a phpMyAdmin oldalát hozza elő nekünk a böngésző, itt a bal oldali sávban válasszuk az „Új” feliratú gombot.
10. Webshop néven, a kódolást állítsuk „utf8mb4\_hungarian\_ci” kódolásúra. Ezután a létrehozás gombot válasszuk ki.
11. A Visual Studio Code – ra visszatérve nyissunk egy új terminált, a terminálba az alábbi parancsot gépeljük be:

cd projekt\_webshop\_weblap

cd KreativOtletcentrum-master

1. A bal oldali listában válasszuk a .env.example fájlt és készítsünk róla egy másolatot. Ha ezt a lépést kihagyjuk, a következő commitolásnál a fájl eltűnik és újra kell generálni.
2. A le másolt .env mappát nevezzük át .env -re.
3. A .env mappában a 14. sorban ez szerepeljen: DB\_DATABASE=webshop.
4. Ezután a terminálba írjuk be sorban a parancsokat, figyeljünk arra, hogy időbe telhet amíg az egyes parancsok befejezik a generálást.
   * composer install
   * php artisan key: generate --asni
   * php artisan migrate
   * php artisan db:seed
   * php artisan serve
5. Ha ezekkel a lépésekkel meg vagyunk akkor a Visual Studio Code -ba visszatérve és annak baloldalán van egy olyan menüpont, hogy „Extensions” kiegészítők, a keresőbe beírva a „Thunder Client” kifejezést egy api kérés szimulációt találunk, kattintsunk a kék install gombra.
6. A telepítést követően a bal oldai sáv egy bekarikázott villám ikonnal bővült, erre rákattintva hozhatjuk elő a szimulációs környezetet.
7. A „New Request” gombra kattinva tudunk új kérést készíteni.
8. Az ezután megnyitott ablakban a kérés típusát állítsuk „GET” kérésre. A mögötte található mezőbe illesszük be ezt az url-t: <http://localhost:8000/api/termekek>.
9. A Send gomb lenyomása után a program JSON formátumban kilistázza az adatbázisban található összes terméket.
10. Ha a felhasználókat szeretnénk megnézni akkor a <http://localhost:8000/api/termekek>. helyet a <http://localhost:8000/api/users> linket használjuk.

Asztali alkalmazás telepítése tesztelésre:

1. IntelliJ megnyitása
2. „Get from VSC” gomb megnyomása, az url mezőbe az alábbi link beillesztése, clone gomb megnyomása, https://github.com/ngabor0508/Projekt\_Webshop\_Raktarkezelo\_Java.git
3. A fejlesztői környezet ezután leklónozza a projektet és meg is nyitja, a rendszer meg szokta kérdezni, hogy bízunk e a projektben és megnyitjuk-e, a „Trust project” gombra kattingva betöltődnek a fájlok.
4. Bal oldalt a fájlok között egy „Main osztály” megtalálható, erre a fájlra 2x rákattintva megnyílik.
5. A sorok mellett balra van sorszámozás lefele, ott a zöld nyilacskára kell rákattintani és a „run main” opciót kiválasztani.
6. Ezután a program elindul, egy bejelentkezési panel jön először elő.
7. A tesztelés érdekében az adatbázis 2 tesztfelhasználót tartalmaz, az egyik admin jogosultságú a másik sima felhasználó. Admin: [admin@example.com](mailto:admin@example.com) : Jelszó: admin jogosultság: 1 - admin  
   Felhasználó: [felhasznalo@example.com](mailto:felhasznalo@example.com) Jelszó: felhasznalo jogosultság: 0 - felhasznalo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Művelet | Bemenet | Várt eredmény | Tapasztalt eredmény | Teszt eredmény |
| Bejelentkezés nem létező adatokkal | Email: [teszt@example.com](mailto:teszt@example.com)  Jelszó: teszt | Hiba üzenet kiírása, bejelentkezési hiba. | Hiba üzenet rendben volt, bejelentkezési hiba megvolt. | Sikeres |
| Bejelentkezés nem admin adatokkal | Email:  [felhasznalo@example.com](mailto:felhasznalo@example.com) Jelszó: felhasznalo | Hiba üzenet kiírása: Nincs jogod bejelentkezni. | Kiírta a hiba üzenetet. | Sikeres |
| Bejelentkezés admin adatokal | Email: [admin@example.com](mailto:admin@example.com)  Jelszó: admin | A rendszer be kell, hogy engedjen. | A rendszer beléptet a felüleltre. | Sikeres |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Művelet | Bemenet | Várt eredmény | Tapasztalt eredmény | Teszt eredmény |
| Bejelentkezés nem létező adatokkal | Email: [teszt@example.com](mailto:teszt@example.com)  Jelszó: teszt | Hiba üzenet kiírása, bejelentkezési hiba. | Hiba üzenet rendben volt, bejelentkezési hiba megvolt. | Sikeres |
| Bejelentkezés nem admin adatokkal | Email:  [felhasznalo@example.com](mailto:felhasznalo@example.com) Jelszó: felhasznalo | Hiba üzenet kiírása: Nincs jogod bejelentkezni. | Kiírta a hiba üzenetet. | Sikeres |
| Bejelentkezés admin adatokal | Email: [admin@example.com](mailto:admin@example.com)  Jelszó: admin | A rendszer be kell, hogy engedjen. | A rendszer beléptet a felüleltre. | Sikeres |

# A program ismert hibái, továbbfejlesztési lehetőségek

## A program ismert hibái

Az asztali alkalmazás össze funkciója tökéletesen működik azonban van egy dolog, ami nem a program hibája, de egy felhasználó annak tekintheti. Az adatbázisba egy külső .csv kiterjesztésű fájl kerül importálásra a backend elindításakor, ez a fájl tartalmazza a termékek adatait. A csv készítésekor valami hiba történhetett, ezáltal minden termék nevének a harmadik szavánál két darab „space” van, így ha valaki keresni szeretne, a két „space miatt” nem ad jó eredményt a program. Ez a hiba nagyon gyorsan és könnyen javítható, de sajnos későn fedeztük fel ezt a hibát.

A másik ismert hiba, hogy amikor a három kategória közül kiválasztunk egyet, pl.: dekor, és közben felveszünk egy új terméket akkor a lista automatikusan nem frissül, így hiába a dekor van kiválasztva akkor is bekerül pl.: egy „alap” kategóriába, a kategória kiválasztásakor természetesen ez frissül és minden a jó kategóriába került. Ennek a fejlesztésén még problémamegoldó fázisban vagyok.

## Továbbfejlesztési lehetőségek

A vizsgát követően szeretnénk ezt a komplett webshop alkalmazás csomagot egy valós boltnak átadni, szolgálatunkra kínálni, ehhez azonban még szükséges, hogy további fejlesztéseket tegyünk, mivel az asztali alkalmazás az én területem, így annak továbbfejlesztésében tudok gondolkodni, az első és az egyik legfontosabb, hogy a rendelések valós idejű kezelése. Ennek megvalósítása még csak ötletelés. Az egyik megoldás lehet egy frissítés gomb, aminek hatására, az asztali alkalmazás frissül. Így az újonnan beérkezett rendelések is bekerülnének a rendelések táblába.

# Felhasználói dokumentáció

## Rendszerkövetelmények

|  |  |  |
| --- | --- | --- |
| **Hardver követelmények** | | |
|  | Minimum gépigény | Ajánlott gépigény |
| Processzor: | Intel Pentium Dual-Core | Intel Core i3-7100 |
| RAM: | 4GB | 8GB |
| Tárhely (szabad): | 2GB | 4GB |
| Internetkapcsolat: | szükséges, min.: 50mbps | szükséges, min.: 100mbps |

|  |  |
| --- | --- |
| **Szoftver követelmények** | |
| Minimum operációs rendszer követelmények | Ajánlott operációs rendszerek |
| Windows 7 Linux Ubuntu 14 | Windows 10  Windows 11 |

## A program telepítésének és konfigurálásának leírása

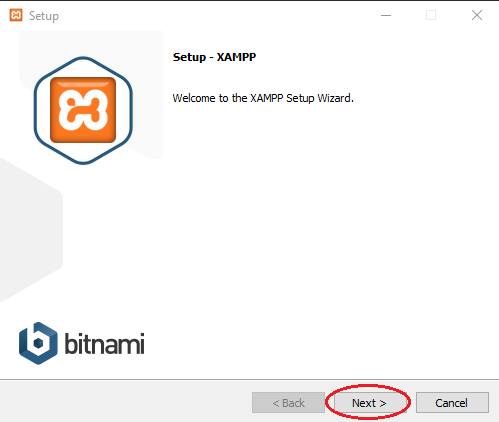
Az asztali alkalmazás jelenleg nem csatlakozik online webkiszolgálóhoz, így jelenlegi állapotban csak helyi hálózaton futtatható. Természetesen ez később feltöltésre kerül a webtárhelyre, domain címre. A domain szerverre feltöltés után az első néhány lépés kihagyása szükséges. A jelenlegi belső hálózat szimulálásához a XAMPP program lesz segítségünkre.

A XAMPP program az alábbi linken tölthető le:  
 <https://www.apachefriends.org/hu/index.html>

A letöltést követően nyissuk meg a letöltött telepítő fájlt. Ezután a XAMPP telepítéséhez kövessük az alábbi lépéseket.

A helyes, ajánlott telepítés érdekében, a piros jelöléseket követve telepítse az alkalmazást.

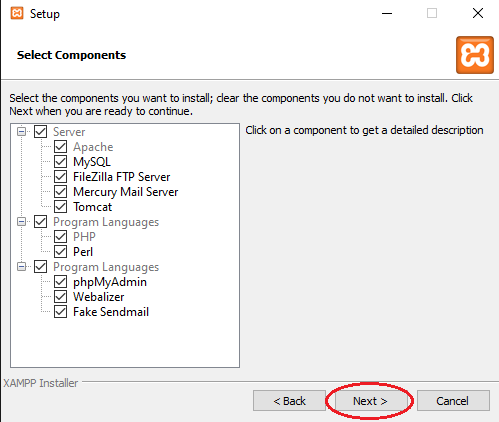
Az alábbi lépések sorrendben vannak így mindig az a segítő ábra fog következni amilyen sorrendben a telepítés zajlik.

1. lépés

árba, xampp telepítő bevezető

1. lépés

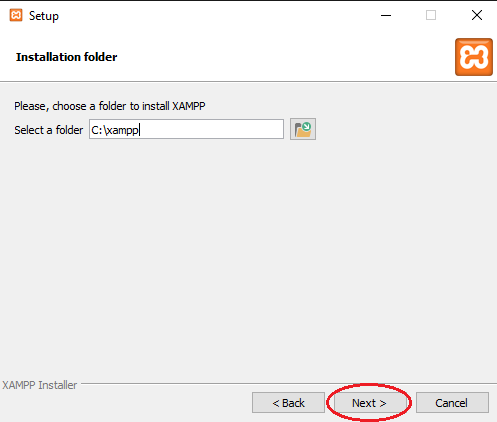
Ennél a lépésnél választhatjuk ki, hogy melyik komponenseket szeretnénk feltelepíteni, ezt hagyjuk alapértelmezetten, így mindent fel fog telepíteni, ami a használathoz szükséges. Ha ezt megváltoztatjuk, előfordulhat, hogy befolyásolja a program működését.



ábra, xampp komponensek

1. lépés

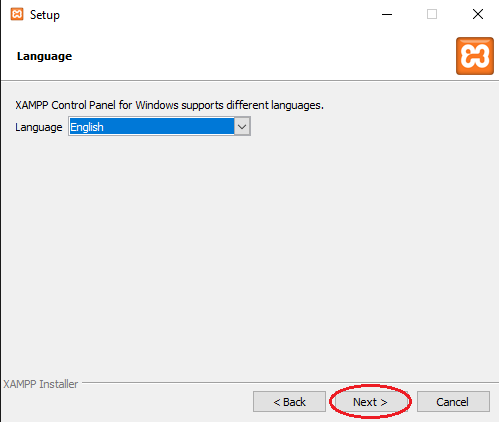
Ennél a lépésnél ki kell választani a telepítési könyvtárat (alapesetben a **C:\** meghajtóra telepíti), bárhova feltelepíthetjük, nem befolyásolja a program működését. A hely kiválasztása után kattintsunk a „next” gombra.



ábra, xampp telepítési helyének kiválasztása

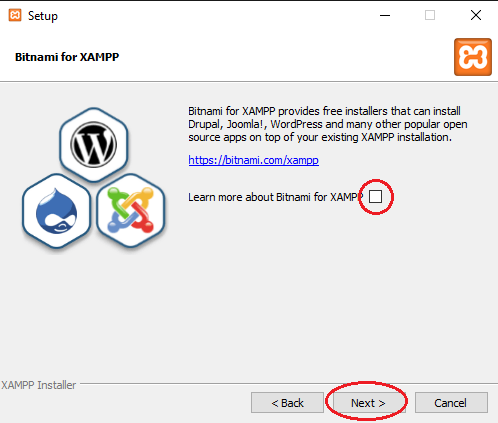
1. lépés

Nyelv kiválasztása, itt lehetőségünk van angol és német nyelv között választani, ezt hagyjuk alapértelmezetten. A nyelv kiválasztása után kattintsunk a „next” gombra.

ábra, nyelv kiválasztása

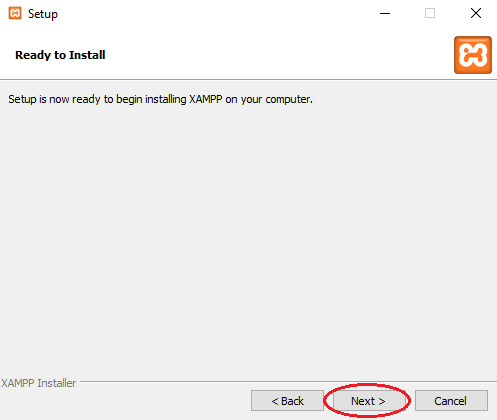
1. lépés

Ha a jelölőnégyzet be van pipálva, még egy kattintással ez kikapcsolható, így telepítés után nem nyílik meg automatikusan a XAMPP oldala. A pipa kiszedése után kattintsunk a „next” gombra.

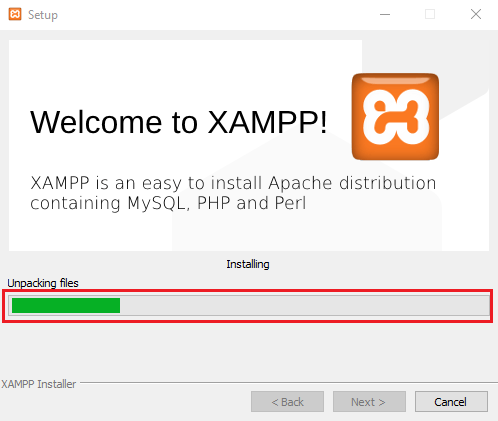
ábra, XAMPP ajánló oldala

1. lépés

Ez az utolsó lépés a telepítés előtt, a „next” gomb megnyomásával megerősíthetjük a telepítés véglegességét. Itt kattintsunk a „next” gombra.

 ábra, telepítés megerősítése

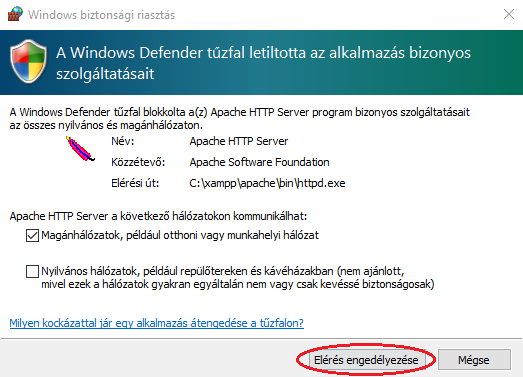
1. lépés

A telepítés folyamata a piros jelölőben található, ennek a végeztével települ fel a XAMPP.

ábra, XAMPP telepítése

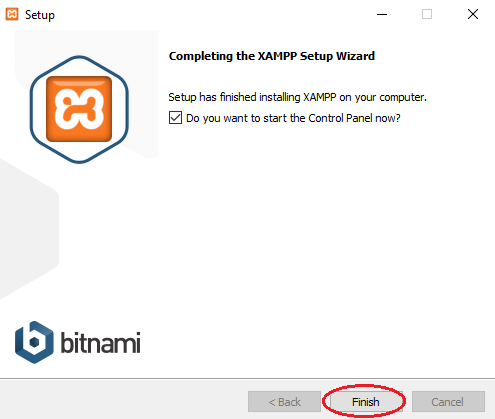
1. lépés

A XAMPP engedélyezése a Windows számára. Itt mind a kettő lehetőségek pipáljuk be. Ezután kattintsunk az elérés engedélyezésére.

ábra, elérés engedélyezése a Windows számára

1. lépés

Utolsó lépés, itt hagyjuk bepipálva a jelölő négyzetet és kattintsunk a „finish” gombra, ezzel befejeződik a XAMPP telepítése.

ábra, XAMPP telepítésének befejezése.

Backendhez szükséges program telepítése

A backend futtatásához szükségünk lesz egy fejlesztői környezetre, amin keresztül le tudjuk futtatni, ezt a MS Visual Studio Code alkalmazás segítségével fogjuk megtenni.

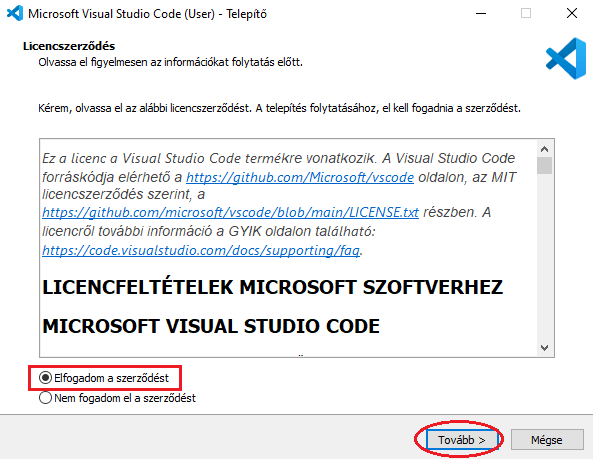
Ezt az alkalmazás az alábbi linken tölthető le a „download for windows” feliratú gombra kattintva:

<https://code.visualstudio.com/>

Ennek telepítési lépései:

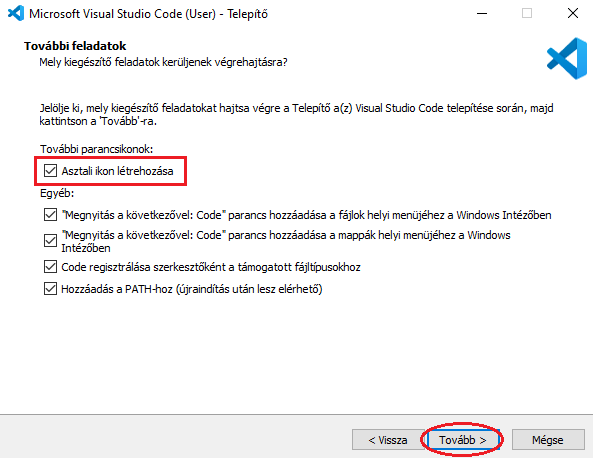
1. lépés

A telepítéshez kattintsunk az „elfogadom a szerződést” feliratra, ezután a tovább gombra.

ábra, licencszerződés elfogadása

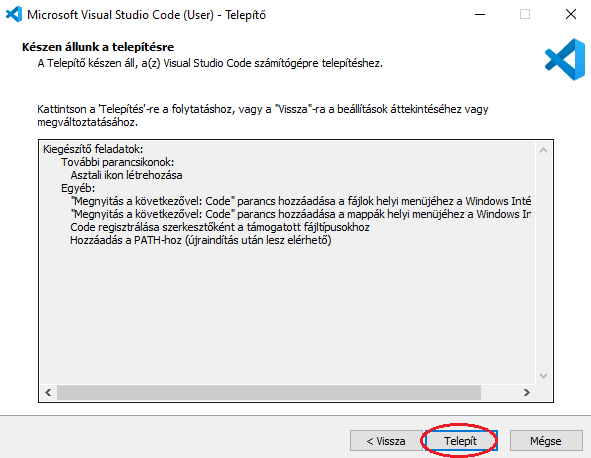
1. lépés

Ennél a lépésnél tudjuk kiválasztani a Windowsba integrálandó lehetőségeket, ezt hagyjuk alapértelmezetten és mellé jelöljük be a pirossal jelzett lehetőséget, így könnyebben meg fogjuk találni a programot.

ábra, Windows lehetőségek

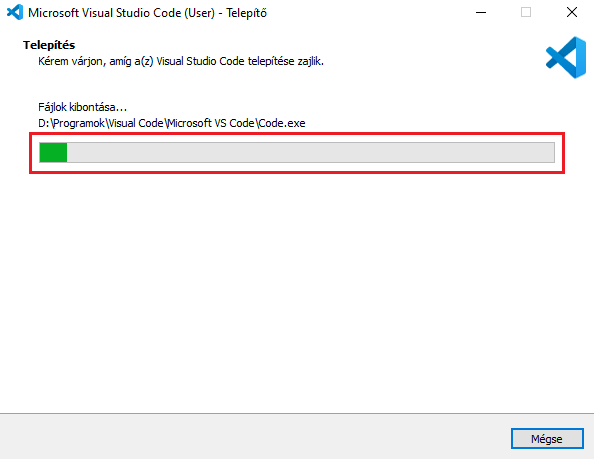
1. lépés

A telepítés előtt ez az utolsó megerősítő ablak, itt a „telepít” gombra kattintva, a program elkezdi a telepítést.

ábra, program telepítés megerősítése

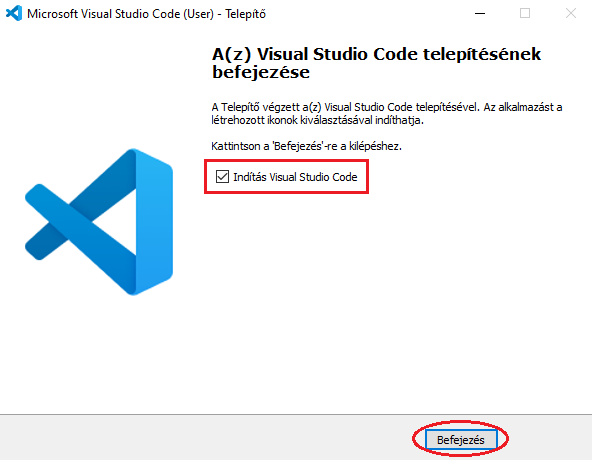
1. lépés

Itt a program feltelepíti magát, itt nincsen semmi teendőnk.

ábra, program telepítése

1. lépés

A program telepítésének befejezése, alapértelmezetten a „Visual Studio Code” elindítása be van jelölve, ezt hagyhatjuk bejelölve, ezután kattintsunk a befejezés gombra.

ábra, telepítés befejezése

Az asztali alkalmazás telepítése

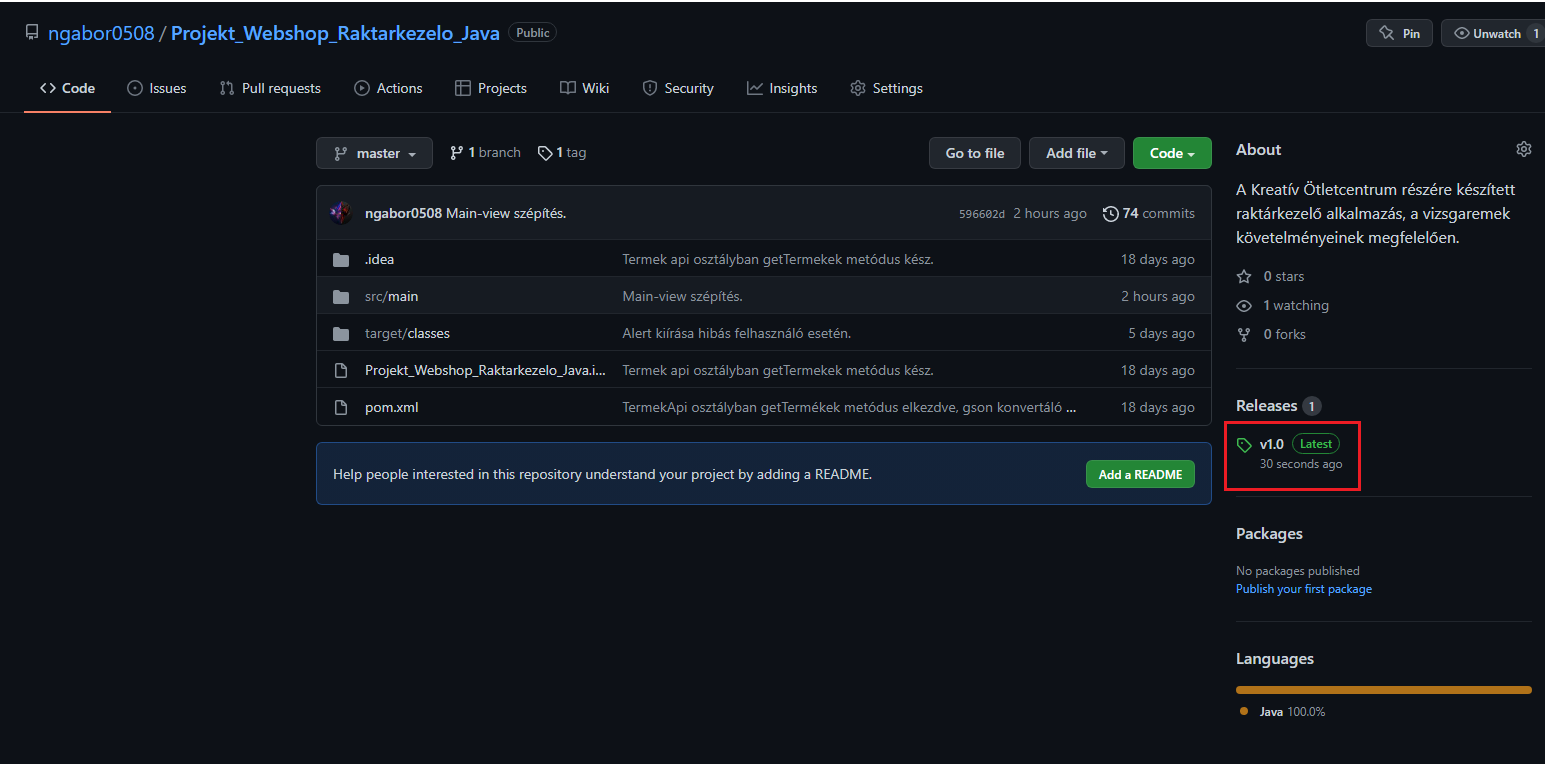
Az asztali alkalmazás verziói GitHubra lettek publikálva, így az onnan letölthető.

Az alábbi linkre rákattintva a Github felület nyílik meg:

<https://github.com/ngabor0508/Projekt_Webshop_Raktarkezelo_Java>

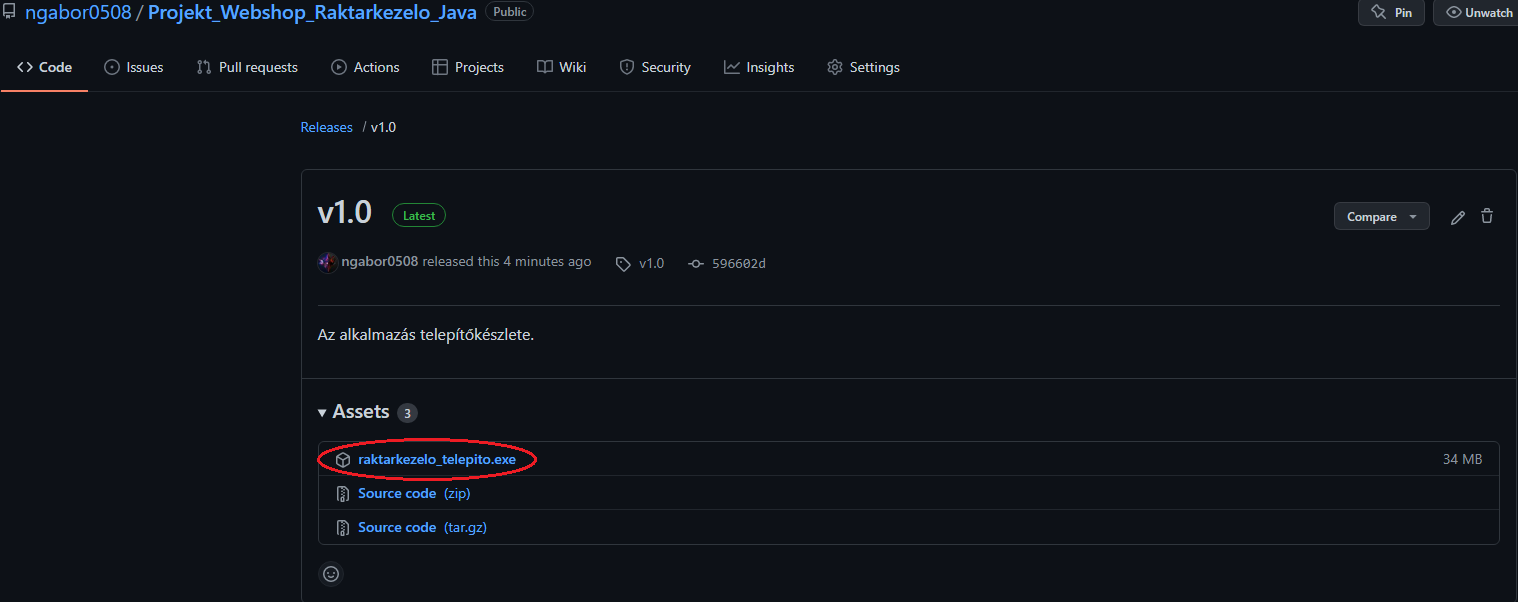
1. lépés

A link megnyitása után ez az oldal nyílik meg, itt a pirossal jelölt verzióra rákattintva tudjuk letölteni az alkalmazás telepítőjét.

ábra, GitHub asztali alkalmazás telepítője

1. lépés

Erről az oldalról tudjuk letölteni a telepítőkészletet. Itt a pirossal jelölt .exe állományra kattintva letöltődik a program. Előfordulhat, hogy a licencszerződés miatt meg kell erősíteni, hogy biztos le szeretnénk e tölteni.

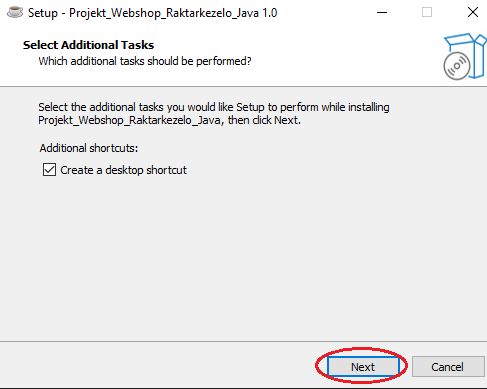
ábra, telepítőkészlet letöltése GitHubról

1. lépés

A program telepítőjének megnyitásakor egy biztonsági figyelmeztetés ugrik fel, itt válasszuk a „További információ” lehetőséget. Ezután lehetőségünk lesz a „Futtatás mindenképpen” gombra rákattintani. Kattintsunk rá. Az újabb hibaüzenetnél válasszuk az „Igen” lehetőséget.

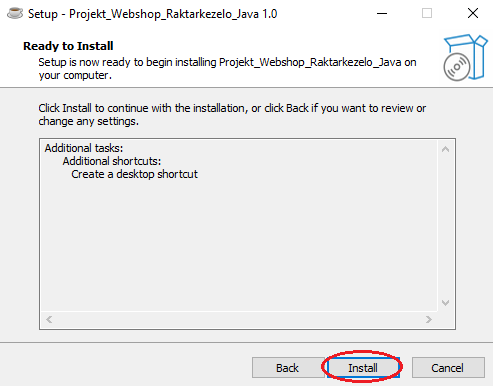
1. lépés

Ennél a lépésnél hagyjunk mindent alapértelmezetten és kattintsunk a „Next” gombra.

ábra, telepítőkészlet asztali alkalmazás

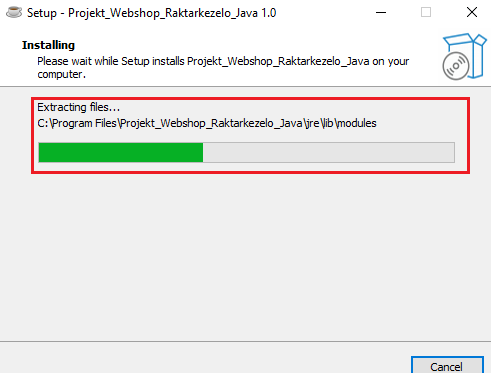
1. lépés

A telepítés előtt ez az utolsó megerősítő ablak, itt az „Install” gombra kattintva, a program elkezdi a telepítést.

ábra, megerősítő ablak telepítés előtt

1. lépés

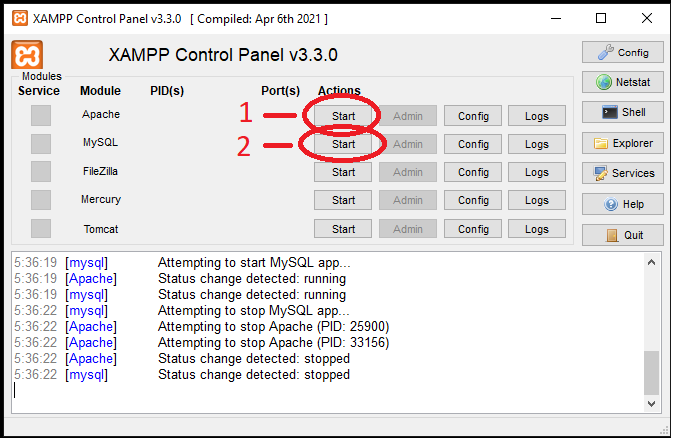
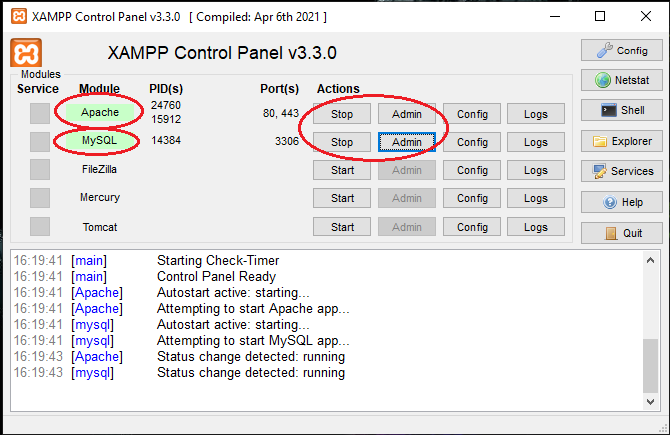
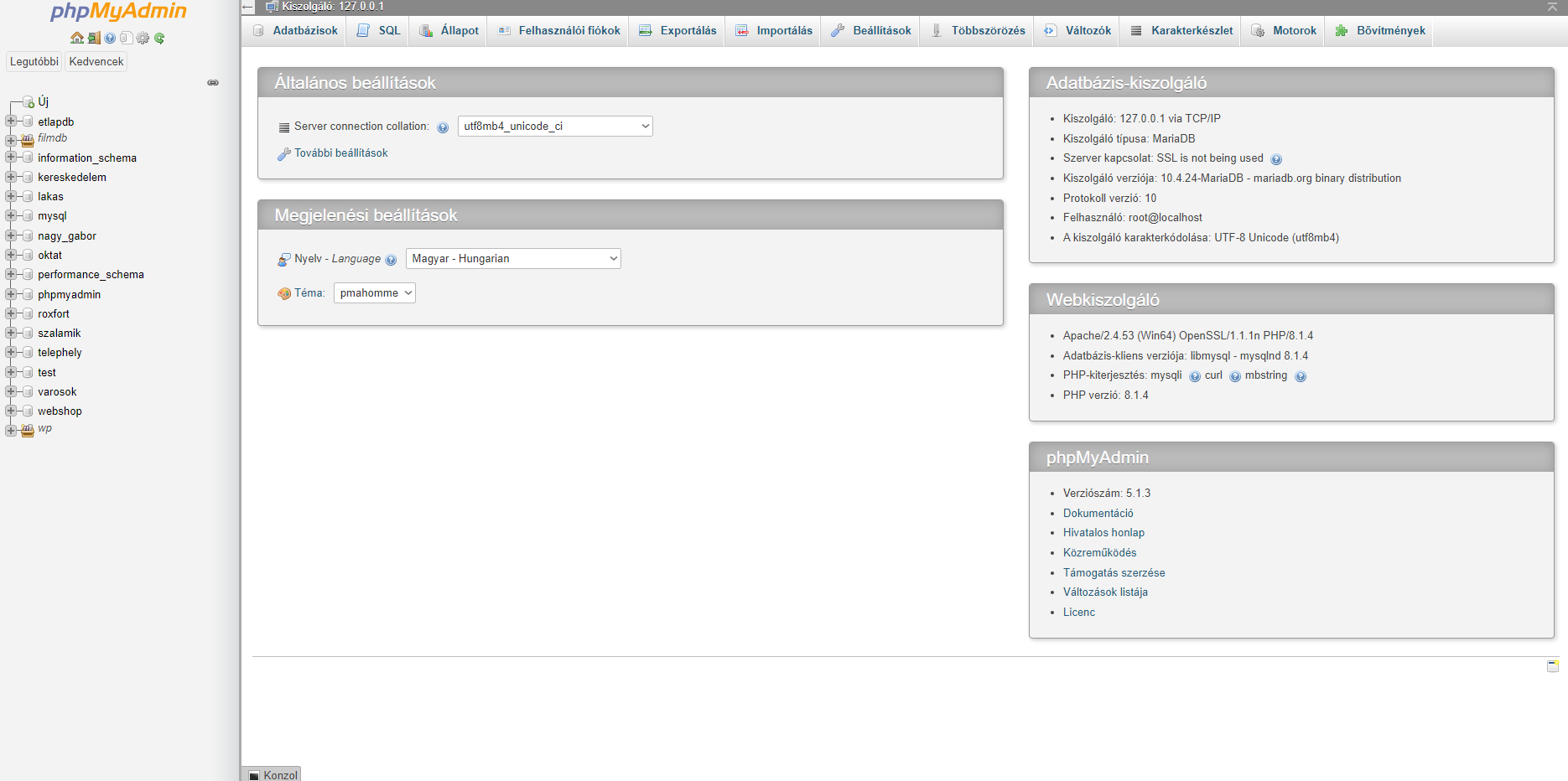
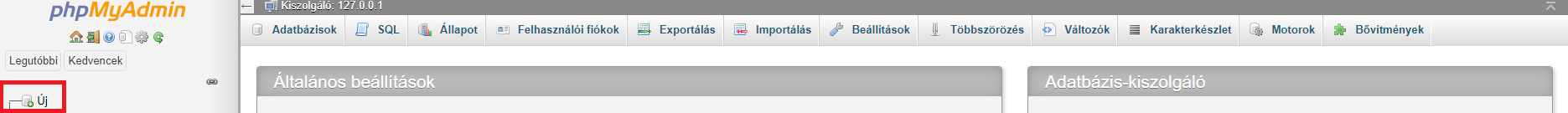
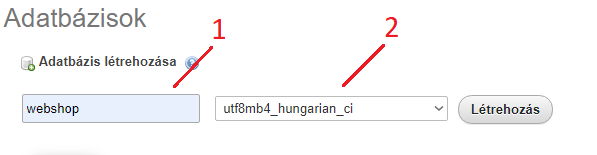
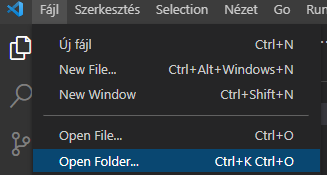
Ennél a lépésnél a program telepítésének folyamata látható, itt nincs semmi teendőnk, a program telepítésének befejezésével az asztalon megjelenik a feltelepített alkalmazás.

ábra, program telepítés befejezése

A program elindítása

Minden szükséges programot feltelepítettünk, ezek után az alkalmazások elindítása maradt.

Backend:

1. XAMPP program megnyitása, admin, Apache, MySQL elindítása.
2. A XAMPP programon belül kattintsunk MySQL sorban lévő „Admin” feliratra. (A képen kék kerettel van jelölve). Ekkor megnyílik a böngészőben egy oldal.  
   ábra, XAMPP admin panel
3.  Itt a bal oldali sávban található egy olyan menüpont, hogy „Új”, piros jelölővel jelölve. Erre kattintsunk rá.
4. Az „Új” menüpontra kattintás után a következő felület kerül elénk.  
   Itt elsőként írjuk be a mezőbe az adatbázis nevét, ami „webshop”, és a listából válasszuk ki a kódolást ami „utf8mb4\_hungarian\_ci”. Ezután kattintsunk a létrehozás gombra. Ezután több teendőnk a XAMPP alkalmazással nincs.
5. Hozzunk létre egy üres mappát az asztalra, pl.: „backend” néven.
6. Visual Studio Code megnyitása.
7. Visual Studio-n belül Fájl <Open Folder lehetőség kiválasztása.

ábra, Visual Studio code, mappa megnyitása

1. Az asztalon létrehozott „backend” mappa tallózása, megnyitása.
2. Új terminál nyitása, az eszközsávon Terminal<New Terminal

ábra, Új terminál nyitása

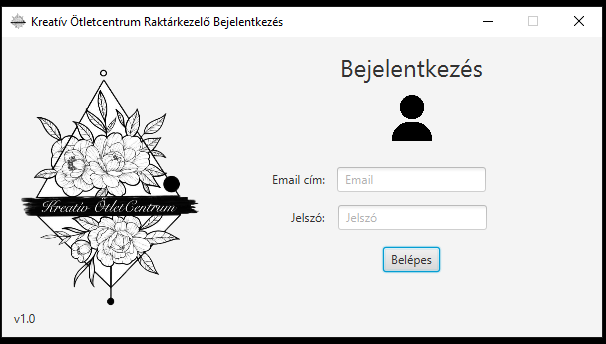
1. Terminálba az alábbi kódsort írjuk be. Kijelölés és beillesztéssel jóval egyszerűbb, mint beírni. Ez a kód leklónozza a projektet, így a mappába kerül.

git clone <https://github.com/ngabor0508/projekt_webshop_weblap.git>

1. A backend mappán belül létrejött egy „projekt\_webshop\_weblap” nevű mappa, ezt kibontva található benne egy „KreativOtletcentrum-master” nevű mappa.
2. A „KreativOtletcentrum-master” mappába belépve a fájlok között megtalálható egy **.env.example** nevű fájl, ezt a fájlt jobb gombbal megnyitva „Átnevezés”/”Rename” nevezzük át **.env -re,** a pontra figyeljünk mert fontos, hogy ott legyen.   
   Az átnevezést követően egy fogaskerék ikonnak kell megjelennie az előző ikon helyett.
3. Ezt követően kattintsunk vissza a terminálba és írjuk be az alábbi parancsokat sorban. A parancsok után enter megnyomásával futtathatjuk le őket. Mindegyik parancs után célszerű várni nagyjából másfél percet.  
   Célszerű másolással, beillesztéssel beírni őket.  
   cd projekt\_webshop\_weblap  
   cd KreativOtletcentrum-master  
   composer install  
   php artisan migrate  
   php artisan db:seed  
   php artisan serve
4. Ha mindent jól csináltunk a végén ezt kell látnunk.  
   ábra, szerver elindítva.
5. Ezután indítsuk el az asztalra feltelepített alkalmazást.

# A program használatának részletes dokumentálása

Bejelentkezés

A program megnyitását követően egy bejelentkezés oldal tárul elénk. Ebbe az alkalmazásba kizárólag adminok tudnak belépni.   


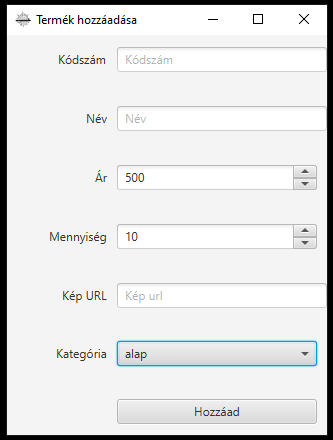
A program kipróbálásához létrehoztam 2 teszt felhasználót, egy „admint” és egy sima „felhasználót”. A sima felhasználónak nincs jogosultsága belépni, így a program elutasítja a bejelentkezést. Ha admin felhasználóval próbálunk bejelentkezni úgy sikeresen beenged az alkalmazás.

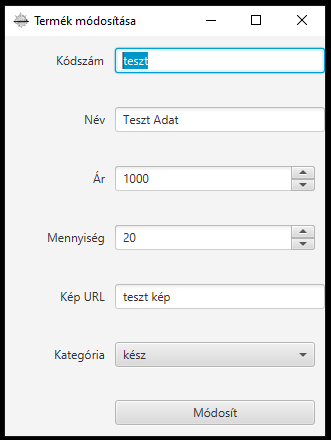
* Admin:
  + Email: [admin@example.com](mailto:admin@example.com)
  + Jelszó: admin
* Felhasználó:
  + Email: [felhasznalo@example.com](mailto:felhasznalo@example.com)
  + Jelszó: felhasznalo

Termék kezelő

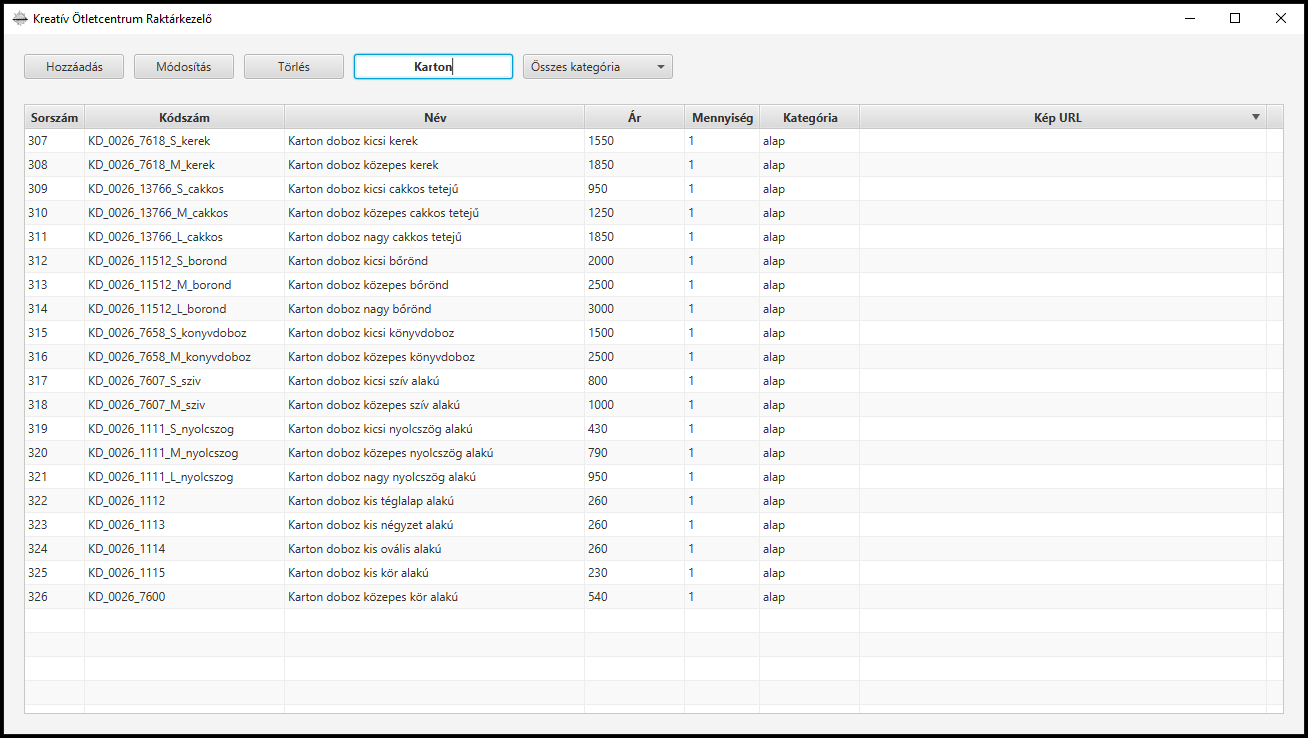
Bejelentkezés után a termék kezelő oldal jelenik meg a bejelentkezés panel ezzel egyidőben bezáródik. A termék kezelő főoldalon vannak listázva a termékek és azoknak tulajdonságaik.

A termékeket az általunk kiválasztott oszlopra kattintva lehet rendezni, növekvő/csökkenő sorrendbe. A termék kezelőnek három módosító művelete van, ami módosítja a termékek táblázatát.

Hozzáadás, ezzel a funkcióval lehet hozzáadni termékeket az adatbázishoz, itt minden mező kitöltése kötelező, a „Kódszámhoz” a terméknek az üzletben is megtalálható kódját kell írni. A „Név” mezőbe a termék teljes megnevezését a beazonosíthatóság érdekében. Az árat egy „görgő” segítségével lehet megadni, de manuálisan is beírható a kívánt összeg, az ár minimum 1ft lehet és alapértelmezetten 500Ft van megadva. A „Mennyiség” mezőhöz a boltban az adott pillanatban megtalálható termék mennyiségét kell írni. Ide 0 is írható, ha jelenleg nincs belőle vagy elfogyott. Ez az érték alapértelmezettem 10 darab. A „Kép URL” megadása is kötelező így ha egy új termék érkezik a boltba akkor képet is lehet hozzá társítani. És a végén a „Kategória” egy választódoboz, ahol a 3 kategória közül lehet kiválasztani, hogy az adott termék melyik a kategóriába tartozik. A Hozzáadás ablak alján egy „Hozzáad” gomb található aminek segítségével lehet hozzáadni a terméket a többi közé.

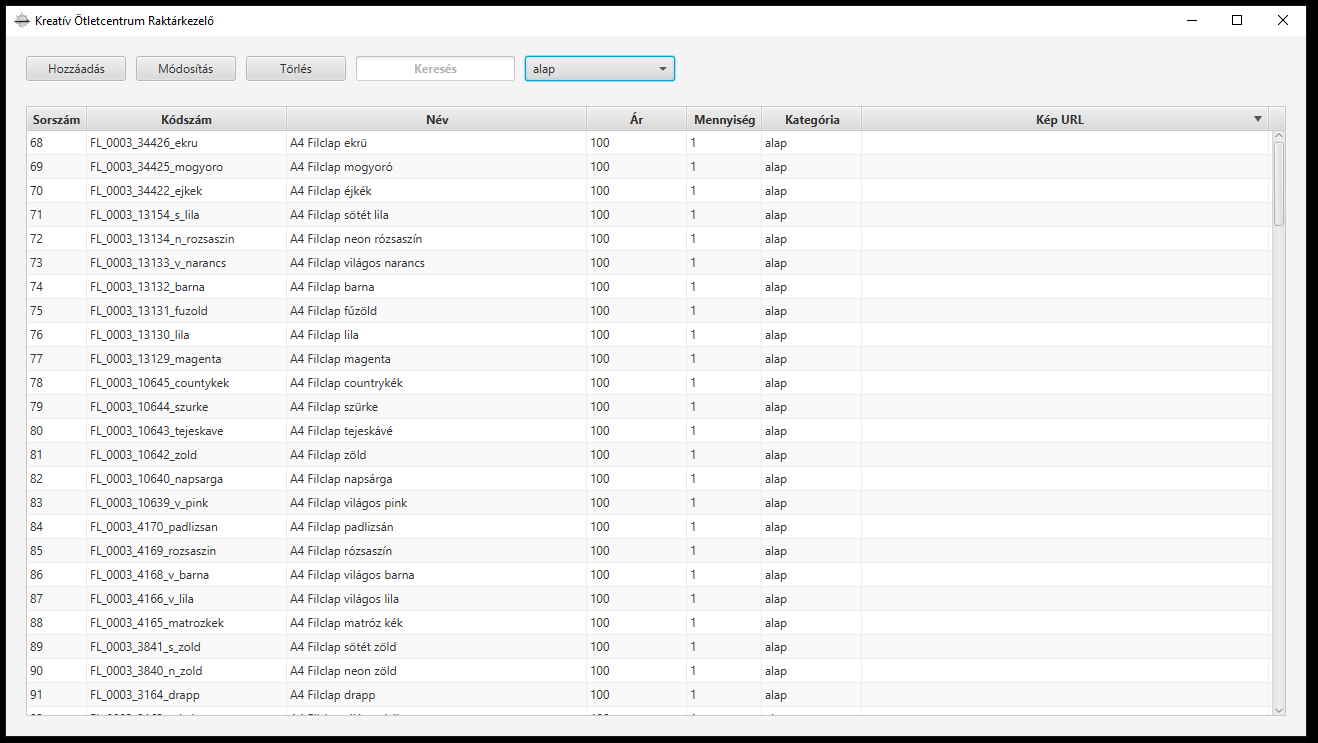
Módosítás, ezzel a funkcióval lehet egy már meglévő terméket módosítani. Ez a funkció arra szolgál, ha a boltban a vásárló vásárol egy termékből például 5 darabot, akkor a bolti eladó le tudja vonni a raktár készletből az eladott mennyiséget, így nyomon követhető az eladás folyamata. A másik funkciója, hogy ha egy terméknél drágulás történik akkor az ár is könnyedén megváltoztatható. A termék módosításához először ki kell választani egy terméket, amit módosítani szeretnénk, ezután tudunk rákattintani a módosít gombra, ahol a lentebbi árba is szemlélteti, hogy betölti a módosítani kívánt adatokat, így nem kell újra beírnunk, kitöltenünk az egész módosítás ablakot, csak a módosítani kívánt mezőt/mezőket. A módosítás mezőire ugyanaz a formátum érvényes, mint a hozzáadás mezőire, mindegyik mezőt kötelező kitölteni, és a mennyiség lehet 0 ha már elfogyott az adott termék. Így nyomon követhető, hogy miből kell rendelni.  
ábra, Termék módosítása

Törlés, ez a funkció azért került bele az alkalmazásba, mert sok olyan termék van, ami csak időszakos, átmeneti így a szezon után már nincs szükség arra, hogy ez benne legyen a termékek között, a másik lehetséges opció, hogy 2x lett felvéve egy termék így az egyiket ezzel a funkcióval kitörölhetjük. Törlés előtt a program megkérdez minket, hogy biztos törölni szeretnénk-e a terméket az OK megnyomásával tudjuk törölni a terméket, a „cancel” gomb megnyomásával megszakíthatjuk ez a folyamatot.

Keresés, ez a funkció a felhasználói élmény érdekében került bele, nagyon sok időd lehet vele nyerni, ha nem kell kikeresni az adatbázisban, hogy melyik termék mennyibe került, hány darab van belőle, ezzel egyszerűen, és gyorsabban lehet termékeket megtalálni. Ahogy a keresendő szó bevitelre kerül, folyamatosan változik a keresési eredmény, nem kell se „enter” se „keresés” gombot lenyomni ennek a neve: „live search”.

ábra, Keresés

Szűrés, a termékeket három nagy csoportra lehet osztani „kész”(termék), „alap” és „dekor”. Ez szintén egy felhasználói élményt javító funkció. A három kategória közül kiválasztva egyet, csak azokat a termékeket adja ki amelyiknél ugyanaz a kategória szerepel mint amit kiválasztott a felhasználó.

ábra, Kategória szűrés.