



Záródolgozat

Készítették:

Szabó Judit Anna - Pejko Bálint - Várdai Balázs

Konzulens:

Farkas Zoltán

Miskolc

2024

Miskolci SZC Kandó Kálmán Informatikai Technikum

Miskolci Szakképzési Centrum

SZOFTVERFEJLESZTŐ- ÉS TESZTELŐ SZAK

Szakedolgozat

PrintFusion

Szabó Judit Anna - Pejko Bálint - Várdai Balázs

2024

Tartalomjegyzék

Témaválasztás indoklása.....	5
Felhasznált technológiák	6
HTML.....	6
CSS.....	6
JavaScript.....	6
Bootstrap.....	6
React	7
React Router.....	7
Jwt	7
MySQL	7
C#	8
Entity Framework Core	8
Asp.Net Core API.....	8
WPF	8
Fejlesztői környezetek	9
Visual Studio Code	9
Visual Studio 2022	9
Swagger	9
Xampp	10
Kommunikációs felületek	11
Github	11
Trello.....	11
Discord.....	11
Webes felület.....	12
Főoldal	12
Introduction.....	12
Shop.....	13
Regisztráció.....	14
Bejelentkezés.....	15
Backend	17
Controllers.....	17
Dto-s	17

Swagger	18
Authentication	18
Adatbázis	19
Aspnetusers tábla.....	19
Aspnetrole tábla.....	20
Hozzaszolasok tábla	20
Kategoriak tábla.....	20
Email	21
Asztali alkalmazás.....	22
Bejelentkezés.....	22
MVVM	23
Nézet.....	24
Modell.....	24
NézetModell.....	24
Összefoglalás	25
Frontend.....	26
Axios.....	26
Szerkezete	26
Források	27

Témaválasztás indoklása

A PrintFusion alkalmazás célja az, hogy egy egyszerű és felhasználóbarát platformot kínáljon a 3D nyomtatással foglalkozó cégeknek és az egyéni felhasználóknak. A fontosságát a 3D nyomtatás rohamos fejlődése adja, valamint az, hogy ez a technológia egyre inkább része lesz mindennapi életünknek és az ipari folyamatoknak is.

A 3D nyomtatás terén jelenleg sok kihívás és probléma merül fel, amelyeket az alkalmazásunk igyekszik megoldani. Ezek közé tartozhatnak:

- A 3D nyomtatáshoz kapcsolódó termékek és szolgáltatások széles skálájának nehézkes elérhetősége.
- A megfelelő nyomtató vagy anyag kiválasztásának bonyolultsága.
- Az egyéni igényekre szabott nyomtatási megoldások hiánya.
- A nyomtatással kapcsolatos folyamatok és szolgáltatások átláthatóságának és hozzáférhetőségének hiánya.

Ezekre a problémákra próbál megoldást nyújtani a PrintFusion alkalmazás, egy olyan platformot kínálva, amely segítségével könnyen és gyorsan megtervezhetők, megrendelhetők és nyomon követhetők a 3D nyomtatással kapcsolatos termékek és szolgáltatások.

Felhasznált technológiák

HTML

A HTML (1. ábra) a Hypertext Markup Language rövidítése. Leíró nyelv, kódnyelv és a weboldalak készítéséhez fejlesztették. A HTML nem programozási nyelv, hanem kódnyelv, aminek segítségével akár a Jegyzetomb segítségével is alkothatók weboldalak. A hipertext jelenti valójában az interneten található oldalakat (dokumentumokat), amelyek szöveg, kép, videó, hang, animáció, vagy ezek valamilyen kombinációjából áll. A HTML ezeknek a dokumentumoknak az elrendezését, formázását tartalmazza saját jelölőnyelvén.



1. ábra – HTML logó

CSS

A CSS (2. ábra) a Cascading Style Sheets rövidítése. Ez az egyik legfontosabb technológia a weboldalak kialakításában. Arra használják a CSS-t, hogy megváltoztassák a HTML dokumentumok kinézetét, stílusát. És lehetővé teszi, hogy különböző elemeket (képek, videók) tehessünk a webes dokumentumba. A webes böngészők megvizsgálják a dokumentum (weblap) CSS kódját, és ez alapján jelenítik meg a HTML elemeket.



2. ábra – CSS logó

JavaScript

JavaScript (3. ábra) a webfejlesztés egyik alapvető nyelve, amit böngészőkben futtatnak. Frontend és backend fejlesztésre is használható. Kliensoldali szkriptnyelv, amely interaktívvá teszi a weboldalakot, és Node.js segítségével szerveroldali alkalmazások is készíthetők vele.



3. ábra – Javascript logó

Bootstrap

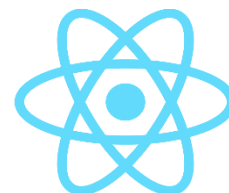
A Bootstrap (4. ábra) egy CSS és jQuery nyelveken megírt keretrendszer amely előre definiált utasításokkal és függvényekkel képes a weboldal dizájnját megjeleníteni. Az osztályokat a keretrendszer határozza meg, és hozzáadja a megfelelő HTML-elemekhez. A Bootstrapel fejlesztés nagyon gyors, de az, hogy előre megírt osztályokat használunk, nem jelenti azt, hogy nem szükséges a HTML, CSS és jQuery alapismerete. A Bootstrap a „mobile first” elvet követi, ami azt jelenti, hogy elsősorban reszponzív weboldalakot fejlesztünk, amelyek mobileszközökön jelennek meg, bár természetesen oldalaink minden eszközön megfelelően jelennek meg.



4. ábra – Bootstrap logó

React

A React (5. ábra) a nyílt forráskódú JavaScript-könyvtárak egyike. Interaktív felhasználói felületek létrehozására szolgál. Ez egy erőteljes, deklaratív és rugalmas könyvtár. Ez magában foglalja a Model-View-Controller (MVC) V (vagy nézet) elemét. Ez nem egy teljes keretrendszer, csak egy front-end könyvtár. Lehetővé teszi összetett felhasználói felületek létrehozását vagy felépítését kis, elszigetelt kódrészletek, úgynevezett komponensek felhasználásával. Az összetevők fő előnye, hogy egy komponens módosítása nem érinti a teljes alkalmazást.



5. ábra – React logó

React Router

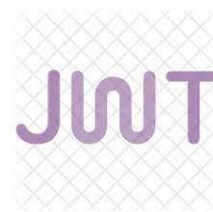
A React Router egy olyan könyvtár, amelyet a React alkalmazások útvonalvezetéséhez és navigációjához használunk. Ez annyit jelent, hogy az URL címünk mindig szinkronban van az alkalmazással és a megfelelő komponenst jeleníti meg. Lehet úgy is használni, hogy egy útvonal egyetlen egy komponenst jelenít meg vagy bizonyos részei az oldalunknak (Navbar, Footer stb.) statikusak maradnak az útvonaltól függetlenül, és így mindig csak a tartalmi rész fog render-elődni.



6. ábra – React-router logó

Jwt

A JWT (Json Web Token) az információ biztonságos továbbításának módja a felek között, általában az ügyfél és a szerver között. A felhasználók hitelesítésére és engedélyezésére szolgál anélkül, hogy munkamenetadatokat kellene tárolnia a szerveren, mivel információkat tartalmaz a felhasználóról, például azonosítóját, hozzáférési jogait és lejáratát. Annak ellenőrzésére szolgál, hogy a felhasználó az-e akinek mondja magát és hogy hitelesítette-e.



7. ábra – JWT logó

MySQL

A Mysql egy nyílt forráskódú relációs adatbázis-kezelő rendszer, melyet a szerveroldali adatok tárolására és kezelésére használnak. Az SQL segítségével lehet lekérdezéseket és módosításokat végrehajtani a táblákon. A MySQL egy népszerű RDBMS a webes alkalmazásokhoz, mert könnyen használható, gyors és megbízható. Ezenkívül a szoftver nagy felhasználói közösséggel rendelkezik, és számos kereskedelmi támogatási lehetőséget kínál. A MySQL emellett nagy rugalmasságot kínál, lehetővé téve az adatok különböző formátumokban történő tárolását és saját sémák meghatározását.



8. ábra – MySQL logó

C#

A C# egy objektumorientált programozási nyelv. Az objektumorientált programozás négy alapelve:

- **Absztrakció:** Az entitások releváns attribútumainak és interakcióinak modellezése osztályokként a rendszer absztrakciós ábrázolásának meghatározásához.
- **Beágyazás:** Elrejt az objektumok belső állapotát és funkcióit, és csak nyilvános függvénykészleten keresztül engedélyezi a hozzáférést.
- **Öröklési képesség:** új absztrakciók létrehozására a meglévő absztrakciók alapján.
- **Polimorfizmus:** Az öröklött tulajdonságok vagy metódusok implementálása különböző módokon több absztrakcióban.

Entity Framework Core

Az Entity Framework Core egy modern objektum-adatbázis-leképező a .NET-hez. Leegyszerűsíti a különböző adatbázisok (például az Azure Cosmos DB, a MySQL, az SQL Server és az SQLite) használatát erősen gépelt .NET- objektumokkal és a LINQ támogatásával.

Asp.Net Core API

A Web API egy alkalmazásprogramozási felület webalkalmazásokhoz vagy webszerverekhez. HTTP-protokollt használ az ügyfelek és a webhelyek közötti kommunikációhoz az adatokhoz való hozzáférés érdekében.

WPF

A WPF a Windows Presentation Foundation rövidítése, és a Microsoft legújabb grafikus felhasználói felülete, amely a .NET keretrendszerrel használható. A GUI a grafikus felhasználói felület rövidítése. A Windows operációs rendszer grafikus felhasználói felülettel rendelkezik, amely lehetővé teszi a számítógépen való munkát és az azzal való interakciót, az esetleg használt böngészőnek is van egy grafikus felhasználói felülete, amely lehetővé teszi az internetezést.

Fejlesztői környezetek

Visual Studio Code

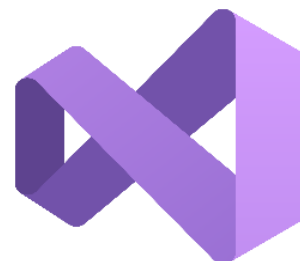
Visual Studio Code egy könnyűsúlyú, ingyenes, és nyílt forráskódú fejlesztői környezet, amely platformfüggetlen, tehát használható Windowson, macOS-en és Linuxon is. Ez egy "kódszerkesztő" funkciókkal, amelyek lehetővé teszik a fejlesztők számára a kódszerkesztés, hibakeresés és verziókezelés egyszerű kezelését. Visual Studio Code alkalmas különböző programozási nyelvekhez és keretrendszerekhez, és a fejlesztők egyedi bővítményeket is telepíthetnek, hogy testreszabják a munkakörnyezetüket.



9.ábra - Visual Studio Code logó

Visual Studio 2022

Visual Studio 2022 egy fejlesztői környezet, amit elsősorban Windows operációs rendszerre terveztek. Segítségével könnyen és hatékonyan készíthetünk különböző típusú alkalmazásokat, beleértve a Windows alkalmazásokat, webalkalmazásokat és szolgáltatásokat. Ebben a fejlesztői környezetben lehetőségünk van kódírássra, hibakeresésre, vizuális tervezésre és az alkalmazások tesztelésére. Emellett a Visual Studio 2022 támogatja a több programozási nyelvet, például a C#, C++, és Visual Basic használatát.



10.ábra – Visual Studio 2022 logó

Swagger

A Swagger egy nyílt forráskódú eszközkészlet, amely lehetővé teszi a fejlesztők számára a RESTful API-k dokumentálását és tesztelését konzisztens, könnyen olvasható formátumban. Ez a dokumentum JSON vagy YAML formátumban készült, és bemutatja az API végpontjait, a végpontok által elfogadott paramétereket, valamint az API által visszaadott válasz- és hibakódokat. Mivel több programozási nyelvet és keretrendszert támogat, használata lehetővé teszi a programozók számára, hogy gyorsabban és hatékonyabban dolgozzanak, és csökkentsék a hibák számát az alkalmazásfejlesztés során.



Swagger™
Supported by SMARTBEAR

11.ábra – Swagger logó

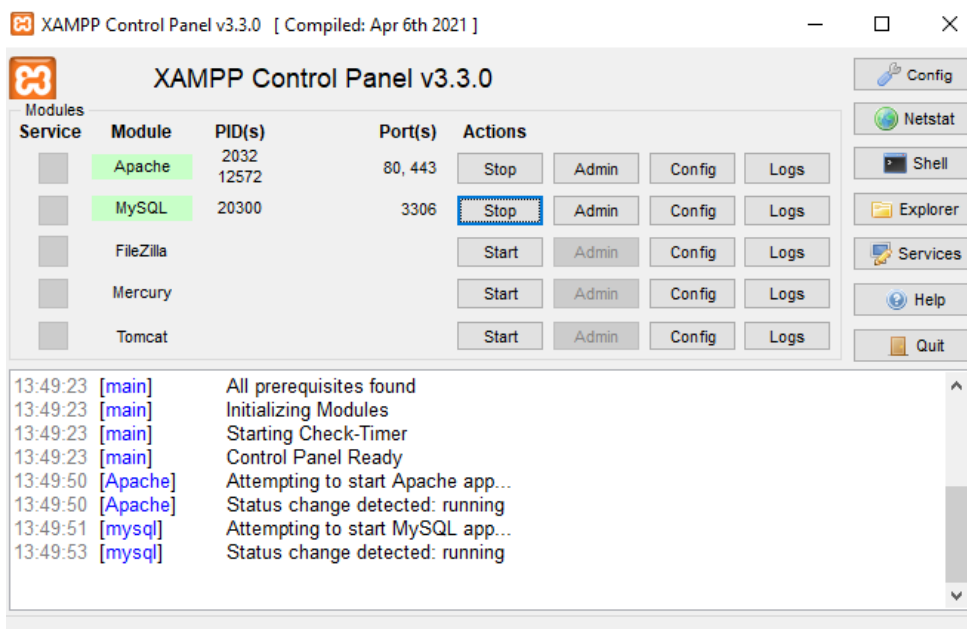
Xampp

XAMPP egy olyan szoftvercsomag, amely az Apache webservert, MySQL adatbáziskezelőt, PHP-t és más szükséges eszközöket tartalmaz, hogy könnyen telepíthetővé és üzembe helyezhetővé tegye egy fejlesztői környezetet. Az XAMPP segítségével a fejlesztők helyi gépükön tudnak webalkalmazásokat fejleszteni és tesztelni. A XAMPP minden szükséges komponenst egy helyre csomagol, így a webalkalmazásokhoz szükséges környezet könnyen elindítható.



12. ábra – XAMPP logó

A XAMPP használatával könnyedén létre tudunk hozni egy teljes webkiszolgáló környezetet a saját számítógépünkön, és tesztelhetjük a webalkalmazásokat és weboldalkat. Az XAMPP-ban az Apache és MySQL szerverek könnyen kezelhetők. Ezért az adatbázisunkat ennek segítségével készítettük el.



13. ábra - XAMPP Control Panel

Kommunikációs felületek

Github

A GitHub, Inc. egy egyesült államokbeli nemzetközi vállalat, amely a Git segítségével szoftverfejlesztési verziókövetés-szolgáltatást nyújt. 2018-ban a Microsoft leányvállalata lett 7,5 milliárd dollárért. Saját funkcióin felül a Git elosztott verziókövetését és forráskódkezelését teszi elérhetővé.



14. ábra – Github logó

Trello

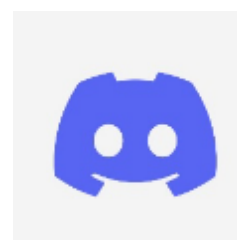


A Trello egy webalapú, Kanban stílusú listakészítő alkalmazás, amely az Atlassian leányvállalata. A felhasználók különböző oszlopokkal hozhatják létre a feladattábláikat, és áthelyezhetik a feladatokat közöttük. Az oszlopok általában olyan feladatállapotokat tartalmaznak, mint a Teendő, Folyamatban, Elkészült. Az eszköz személyes és üzleti célokra használható, ideértve az ingatlankezelést, a szoftverprojekt-menedzsmentet, az iskolai hirdetőtáblákat, az óratervezést, a könyvelést, a webtervezést, a játékokat és az ügyvédi irodák esetkezelését.

15. ábra – Trello logó

Discord

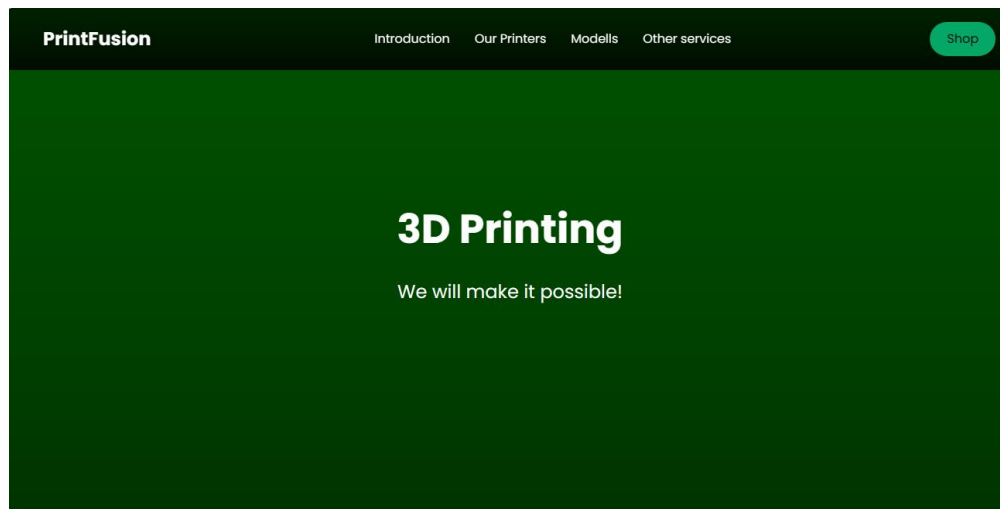
A Discord egy ingyenes VoIP-alkalmazás és digitális terjesztési platform, amelyet legfőképp videójáték-közösségek számára terveztek, de nem zár ki semmilyen más témájú közösséget sem. A Discordot úgy tervezték, hogy nagy rendszerigényű programok futtatása mellett is gördülékenyen lehessen használni.



16. ábra – Discord logó

Webes felület

Főoldal



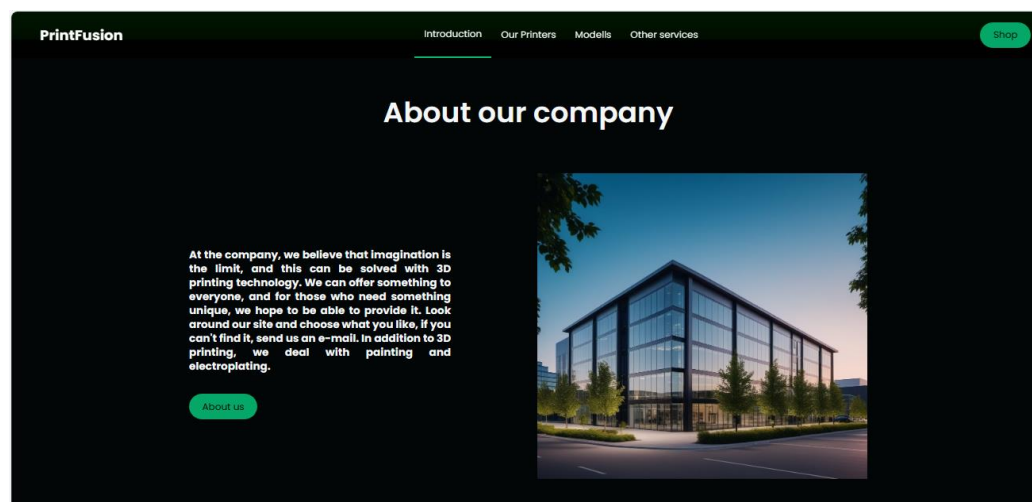
17.ábra - főoldal

A navigációs sáv tartalma:

A navigációs sáv, vagy navbar, általában a főoldal tetején található, és menüpontokat tartalmaz, amelyekre kattintva a felhasználók a megfelelő oldalra juthatnak:

- Introduction – Bevezetés
- Our printers – Nyomtatóink
- Modells – Modellek
- Other services – További szolgáltatásaink
- Shop – Bolt

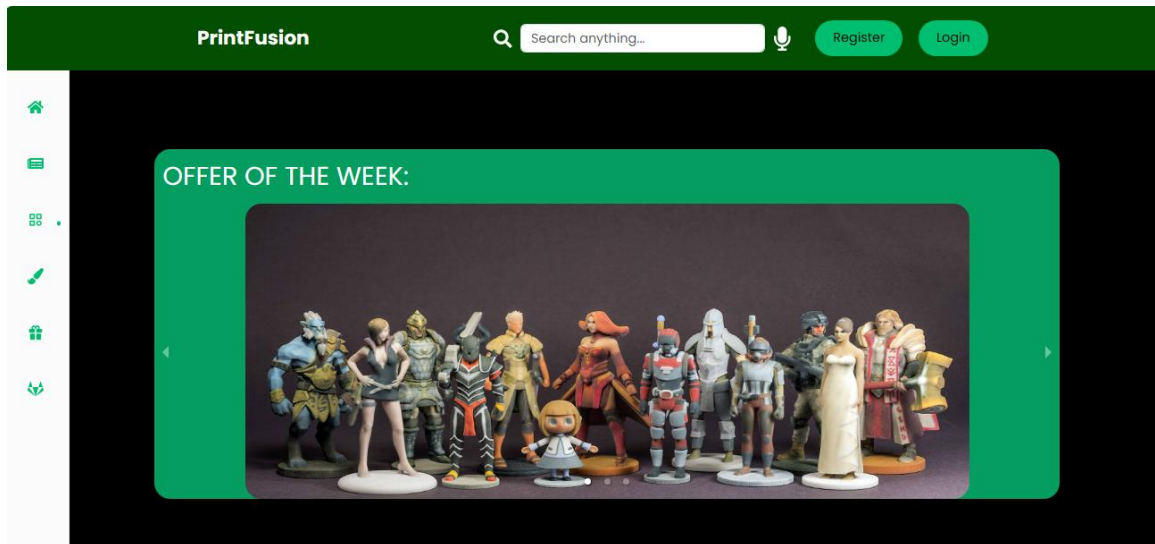
Introduction



18.ábra – Bevezetés aloldal

Shop

A shop gomb megnyomásával a felhasználók regisztráció, illetve bejelentkezés után tudnak 3D nyomtatott tárgyakat rendelni.

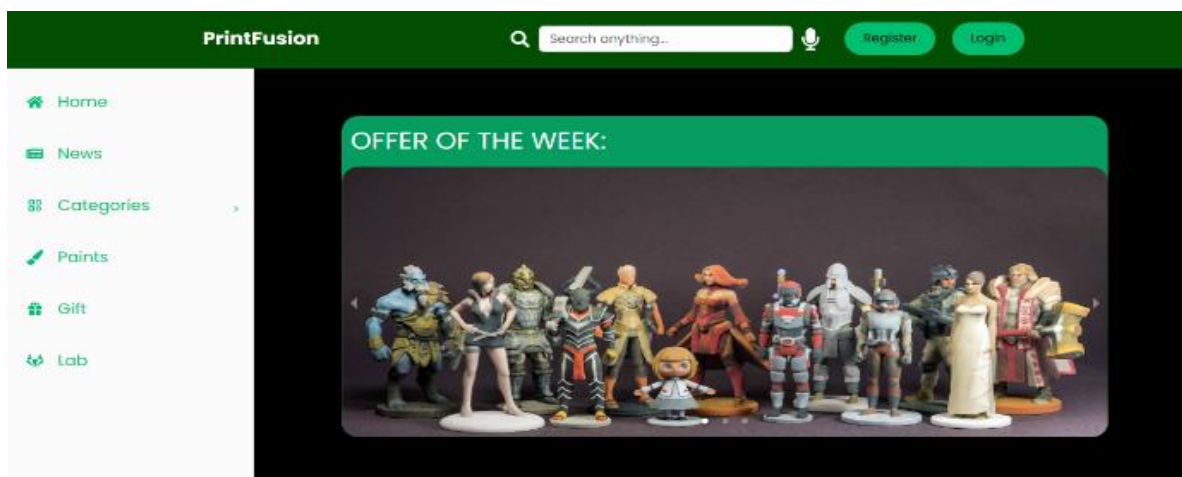


19. ábra – Shop aloldal

OFFER OF THE WEEK: Az éppen aktuális 3D híreket jeleníti meg egy slide carousel-ben.

Az oldalsó navbar tartalma:

- Home - gombra kattintva a főoldalra irányít vissza
- News
- Categories
- Paints
- Gift
- Lab

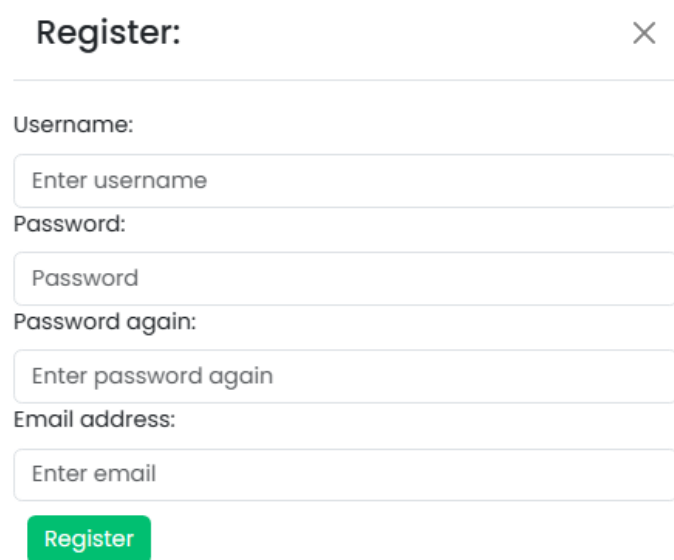


20. ábra – oldalsó navbar

A navbar-ban található egy kereső, amely hang alapján is tud keresni a kereső felület melletti mikrofon ikonra kattintva.

Regisztráció

A register gombra kattintva megjelenik ez a modal ablak, amely lehetővé teszi az új felhasználók számára a regisztrációt a weboldala. Ez a modal ablak egy űrlapot tartalmaz (21. ábra), amelyben az új felhasználók megadják a szükséges információkat, mint például a felhasználónevüket, email címüket, illetve a jelszavukat.



21. ábra – regisztráció modal

A Register gombra kattintva ha a felhasználó helyesen minden mezőt kitöltött egy automatikus visszaigazoló emailt kap a megadott email címére. Ez a visszaigazoló email egy 4 számjegyű kódot tartalmaz, a kétfaktoros hitelesítéshez. Ezt a 4 számjegyű kódot beírva történik meg a hitelesítés és a regisztráció.

A jelszavakat HASH-é alakítva adjuk át a frontend, illetve a backend részére, és így tároljuk az adatbázisunkban a felhasználók biztonsága érdekében.

Felépítése:

- Az username(felhasználónév) beviteli mező, a beírt szöveg betű vagy szám lehet,
- A password(jelszó) beviteli mező, a beírt szövegnek tartalmaznia kell számot és betűt is,
- Az enter password again (jelszó újra) beviteli mező, a beírt szövegnek egyeznie kell a jelszóéval, Az email beviteli mező, a beírt szövegnek tartalmaznia kell “@” és “.” karaktereket,
- A Register(regisztráció) gomb.

Bejelentkezés

A bejelentkezési oldalon a regisztráció során megadott helyes felhasználónév és jelszó páros megadásával a Login gombra kattintva tud majd bejelentkezni

Login: ×

Username:

Password:

Login

22.ábra – Login modal

Reset Password: ×

Email address:

Email code:

New Password:

Reset Password

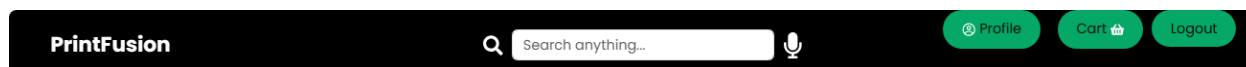
Ha a felhasználó elfelejtette a jelszavát, akkor van lehetőség jelszóváltoztatásra is. Ehhez szükség van:

- A regisztrációnál használt email címre
- A megadott emailer érkező 4 számjegyű kódra
- Az új jelszó beírására

Ha megtörtént a jelszóváltás, akkor az új jelszóval tud bejelentkezni a felhasználó

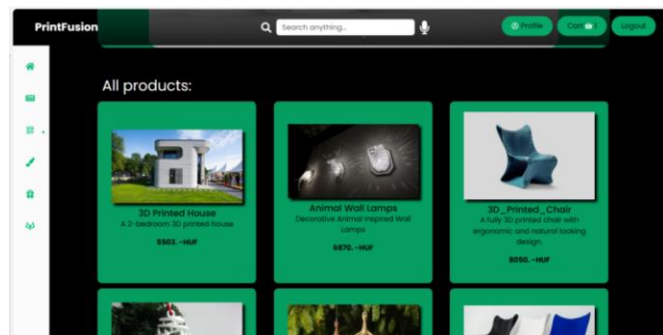
23.ábra – Elfelejtett jelszó modal

Amint a felhasználó sikeresen bejelentkezett, a navigációs sávon megjelenik egy Profil gomb, egy Kosár és a Kijelentkezés gomb.



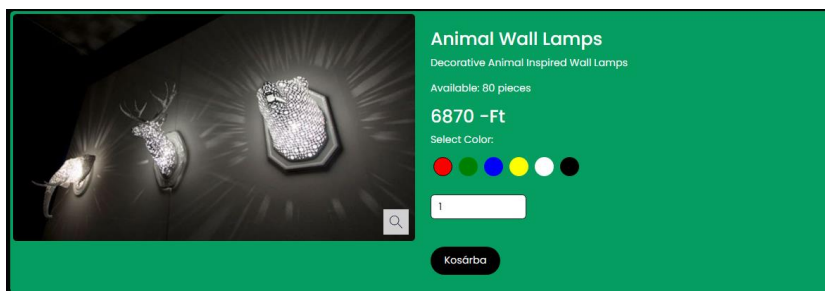
24.ábra – Navigációs sáv

Lentebb pedig a megvásárolható 3D nyomtatott termékek:



25.ábra – Termékek felsorolása

A termékekre kattintva kiválasztható a megvásárolni kívánt termék színe és mennyisége is.



26.ábra – Egy termék kártya

A felhasználók itt tudnak hozzászólni, értékelni és megosztani véleményüket.

Fontos, hogy a hozzászólásoknak nem szabad személyes adatokat tartalmazniuk, és a felhasználóknak tisztelettel kell viselkedniük egymással szemben.



27.ábra - Komment

Kosár tartalma:

- A kiválasztott termék(ek) neve,
- A kiválasztott termék(ek) ára,
- A kiválasztott termék(ek) darabszáma,
- A kiválasztott termék(ek) színe,
- A fizetendő összeg

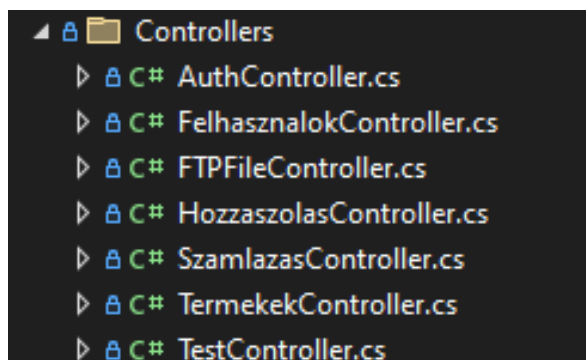


28.ábra – Kosár tartalma

Ha meggondolta magát a felhasználó és mégsem szeretné megrendelni az adott terméket, akkor az “X” gombbal kitörölheti a kosárból.

Backend

Controllers

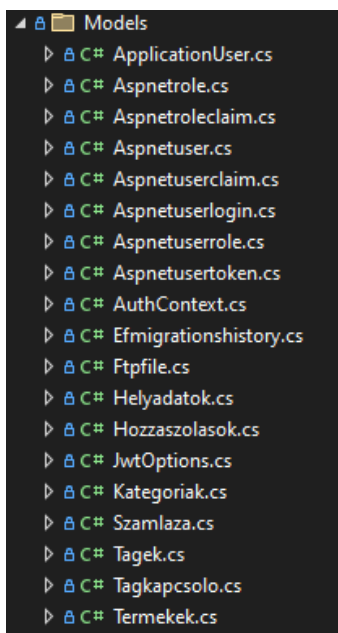


29.ábra – Controllers

Dto-s

Az adatátviteli objektumok (Data Transfer Objects) a különböző rétegek közötti adatátvitelre szolgáló osztályok. DTO-k segítségével adatokat lehet átvinni a kliens és a szerver között. A DTO-kat használhatja arra, hogy csak olyan entitásadatokat továbbítsanak, amelyek szükségesek a kliens és a szerver közötti konkrét műveletek végrehajtásához. Webalkalmazásokban a DTO-k segítenek elszigetelni a kliensoldali kódot a szerveroldali adatstruktúráktól.

Models



A models mappában található osztályokat általában az alkalmazásban használt adatmodelleknek nevezik. Ezek az osztályok az adatstruktúrákat és azok tulajdonságait határozzák meg.

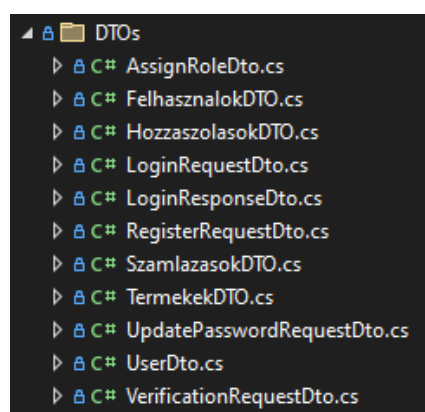
Az Entity Framework Core (EF Core) egy népszerű ORM (Object-Relational Mapping) keretrendszer, amely lehetővé teszi az alkalmazások számára, hogy könnyedén kommunikáljanak az adatbázisokkal. A scaffold egy olyan folyamat, amely során az EF Core automatikusan generálja az adatbázis sémáját a meglévő adatbázisból.

A következő Scaffold db paranccsal generáltuk le a backendet:

```
"SERVER=localhost;PORT=3306;DATABASE=database_name;  
USER=root;PASSWORD=;SSL MODE=none;"  
mysql.entityframeworkcore -outputdir Models -f
```

31.ábra – Models

Egy program vagy weboldal backendje a felhasználó számára láthatatlan része az, ahol a tényleges számítások zajlanak. A backend feladata, hogy feldolgozza a frontendről (a felhasználóval interakcióba lépő rész) érkező adatokat és az eredményeket visszaküldje a frontendnek. A mi backendünk az ASP.NET Core Web Api keretrendszerre épült (29.ábra). A modellt egy adatbázisból generáltuk a 'Scaffold-DbContext' parancs segítségével.



30.ábra- DTO-s

Swagger

Authentication

Auth		^
POST	/Auth/register	✓
POST	/Auth/login	✓
POST	/Auth/updatepassword	✓
POST	/Auth/AssignRole	✓ 🔒
POST	/Auth/verify-email	✓

32.ábra – Swagger authentication

```
namespace ProjectBackend.DTOs
{
    1 reference
    public class RegisterRequestDto
    {
        1 reference
        public string Username { get; set; }

        1 reference
        public string Password { get; set; }

        2 references
        public string Email { get; set; }
    }
}
```

- A kód a C# programozási nyelvet használja.
- A névtér a “ProjectBackend.DTOs” nevű névtér deklarálja.
- A “RegisterRequestDto” osztály három nyilvános tulajdonságot tartalmaz: “Username”, “Password” és “Email”. Mindegyik tulajdonság típusa string, és mindegyikhez van getter és setter metódus.

33.ábra – Visual Studio Code

Termékek	
GET	/Termékek
POST	/Termékek
GET	/Termékek/Kategoriak
GET	/Termékek/Termek/{kategoriaId}
GET	/Termékek/EgyTermek/{id}
PUT	/Termékek/{id}
DELETE	/Termékek/{id}

34.ábra -Swagger termékek

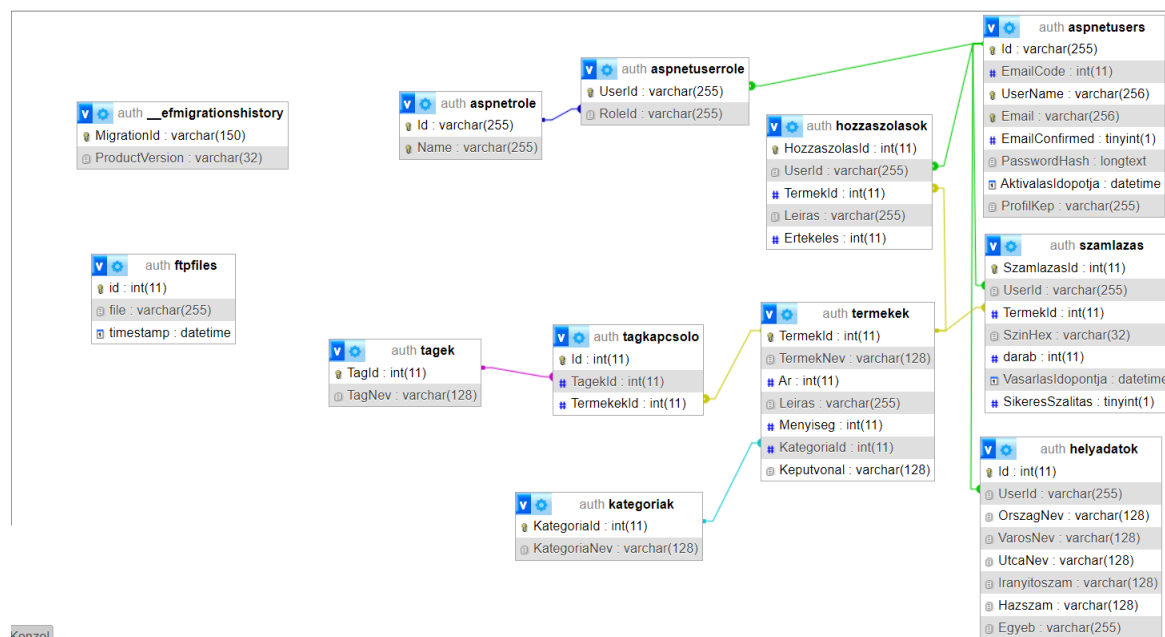
- **GET:** A GET metódust arra használjuk, hogy információt szerezzünk a megadott szerverről egy adott URI segítségével. A GET kérések csak az adatok rekérdezésére szolgálnak és nem gyakorolnak más hatást az adatokra.
- **POST:** A POST kérés arra szolgál, hogy adatokat küldjünk a szervernek, például ügyfél-információkat, fájlfeltöltést stb. HTML űrlapok segítségével.
- **PUT:** A PUT kérés arra szolgál, hogy az aktuális cél erőforrás minden jelenlegi reprezentációját a feltöltött tartalommal helyettesítse.
- **DELETE:** A DELETE kérés arra szolgál, hogy az aktuális cél erőforrás minden jelenlegi reprezentációját a megadott URI alapján eltávolítsa.

Adatbázis

A MySQL adatbázisunkban (35. ábra) 12 tábla található, melyekben a következő adatokat tároljuk:

- A felhasználók regisztrációs adatait. Megkülönböztetés céljából a felhasználók más-más jogosultsági szinttel rendelkeznek
- A megvásárolható termékek adatait
- Számlázáshoz szükséges adatokat, például a megrendelt termék azonosítóját, színét, darabszámát, vásárlás időpontját
- Helyadatokat, amik a szállításhoz szükségesek
- Az adott termékekhez a hozzászólásokat, értékeléseket

Az adatbázis létrehozásához phpMyAdmin programot használtunk, A lenti ábrán (35. ábra) a tervezői nézetet és a táblák közötti kapcsolatot láthatjuk.



Konzol

35. ábra – Adatbázis

Aspnetusers tábla

Id: A felhasználó azonosítója, VARCHAR típusú, Primary Key

EmailCode: A felhasználó regisztráció után emailbe kap egy 4 számjegyű kódot, INT típusú

UserName: A felhasználó neve, amit regisztrációnál ad meg, VARCHAR típusú

Email: A felhasználó e-mail címe, amit regisztrációnál ad meg, VARCHAR típusú

EmailConfirmed: A 2 lépésös azonosításhoz használt 4 számjegyű kód megerősítése után lesz a regisztráció hitelesítve, TINYINT típusú

PasswordHash: A regisztrációnál megadott jelszó Hash-é alakítva, a nagyobb biztonság érdekében, LONGTEXT típusú

AspNetrole tábla

					Id	Name
<input type="checkbox"/>					81f1660f-8329-4f49-9b16-7a1592d43d7e	ADMIN
<input type="checkbox"/>					49d1592d-dc96-4868-834e-fa4e837e3d97	USER

36.ábra – AspNetrole tábla

id: A szerepkör azonosító száma, GUID típusú, Primary Key

name: A szerepkör megnevezése, VARCHAR típusú

Hozzaszolasok tábla

					HozzaszolasId	UserId	TermekId	Leiras	Ertekeles
<input type="checkbox"/>					53	03c66e48-f575-4ffb-9252-f307d1b473e0	21	Looks amazing on my desk, very sturdy.	5

37.ábra – Hozzászólások tábla

HozzaszolasId: A hozzászólás azonosító száma, INT típusú

UserId: A felhasználó azonosító száma, aki írta a hozzászólást, INT típusú

TermekId: A termék azonosító száma, amelyik termékhez lett hozzászólás írva

Leiras: A hozzászólás

Kategoriak tábla

					KategoriaId	KategoriaNev
<input type="checkbox"/>					1	Home decor
<input type="checkbox"/>					2	Gaming
<input type="checkbox"/>					3	Movies
<input type="checkbox"/>					4	Accessories
<input type="checkbox"/>					5	Festive decoration
<input type="checkbox"/>					6	Statues
<input type="checkbox"/>					7	Miniatures
<input type="checkbox"/>					8	Wall decor
<input type="checkbox"/>					9	Other
<input type="checkbox"/>					10	Furniture

KategoriaId: A termék kategóriájának azonosító száma, INT típusú

KategoriaNev: A termék kategóriájának megnevezése, VARCHAR

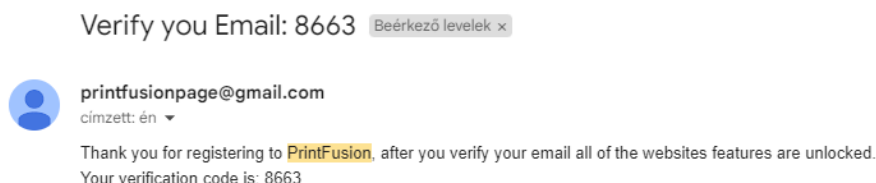
38.ábra- Kategóriák tábla

Email

Az automatikus email értesítések lehetővé teszik, hogy a felhasználók automatikusan értesítést kapjanak bizonyos eseményekről vagy tevékenységekről. Ezen értesítések elküldése automatikusan történik, amint az adott feltétel teljesül.

Regisztráció esetén:

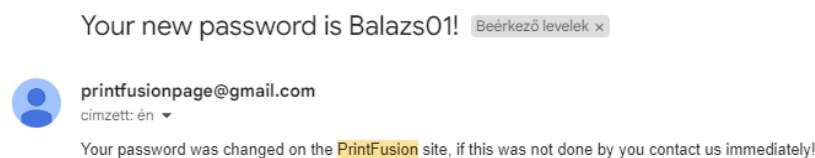
Küld egy 4 számjegyű kódot, melyet automatikusan generál le.



39. ábra – Regisztráció utáni email

Elfelejtett jelszó esetén:

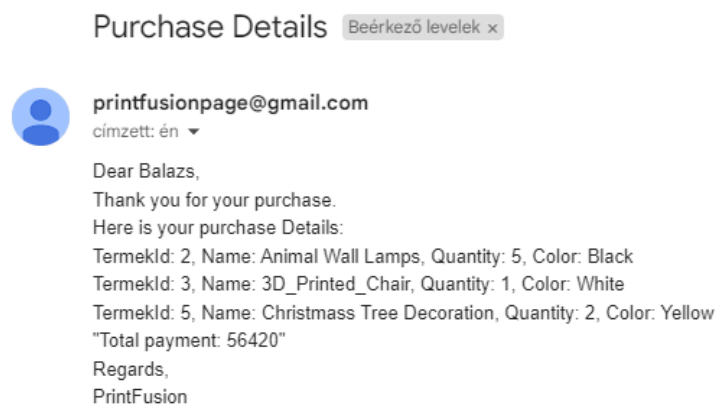
Ha elfelejtettük a jelszavunkat és jelszóváltoztatást kérünk, akkor a megadott email címre jön egy email, amiben leírja, hogy sikeresen változtattuk meg a jelszavunkat!



40. ábra – Elfelejtett jelszó utáni email

Megrendelés esetén:

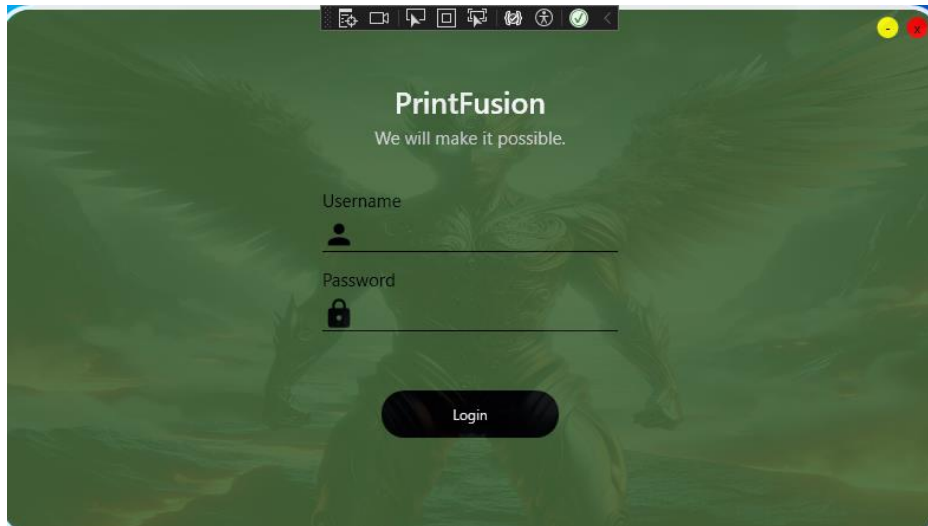
Ha leadtuk a rendelésünket, akkor automatikusan jön egy email, amely tartalmazza a megrendelt termékek azonosítóit, neveit, darabszámait, színeit és a fizetendő összeget.



41. ábra- Megrendelés utáni email

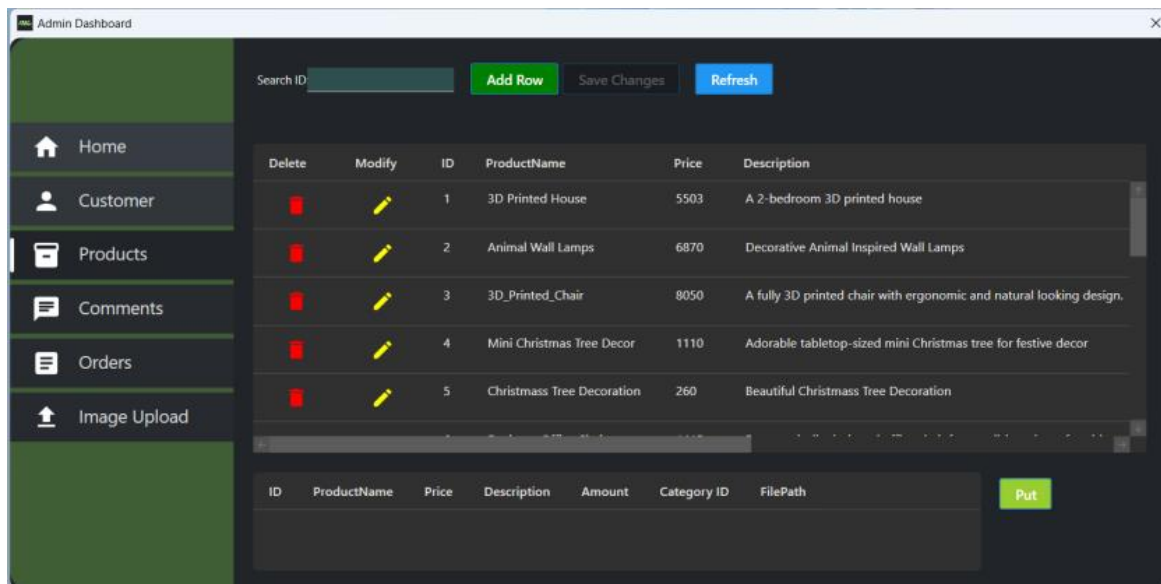
Asztali alkalmazás

Bejelentkezés



42.ábra – Asztali alkalmazás admin felület bejelentkezés

Az adminisztrátorok bejelentkezést követően itt módosíthatják a termékek adatait, például az árat, a leírást vagy a készletmennyiséget. Továbbá lehetőségük van a termékeket törölni, ha már nem elérhetőek vagy nem aktuálisak. A táblázat hasznos lehet a webhely adminisztrálásának bemutatásához, és segíthet az adminisztrátoroknak hatékonyan kezelni a termékekkel kapcsolatos információkat.

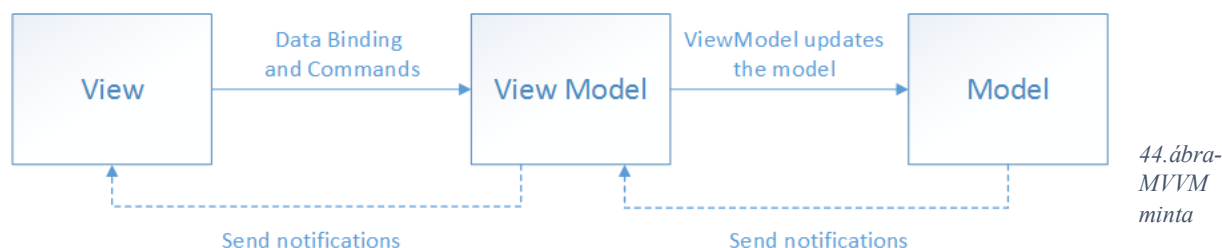


43.ábra – Asztali alkalmazás admin felület

MVVM

Az MVVM-minta segít tisztán elkülöníteni az alkalmazás üzleti és megjelenítési logikáját a felhasználói felülettől (UI). Az alkalmazáslogika és a felhasználói felület tiszta elkülönítése számos fejlesztési probléma megoldásában segít, és megkönnyíti az alkalmazások tesztelését, karbantartását és fejlesztését. Emellett jelentősen javíthatja a kód újrafelhasználási lehetőségeit, és lehetővé teszi a fejlesztők és a felhasználói felület tervezői számára, hogy könnyebben együttműködjenek az alkalmazás megfelelő részeinek fejlesztése során.

Az MVVM-mintának három alapvető összetevője van: a modell, a nézet és a nézetmodell. Mindegyik külön célt szolgál. Az alábbi ábra a három összetevő közötti kapcsolatokat mutatja be.



Az MVVM-minta használatának előnyei a következők:

Ha egy meglévő modell implementációja meglévő üzleti logikát foglal magában, nehéz vagy kockázatos lehet annak módosítása. Ebben az esetben a nézetmodell adapterként működik a modellosztályok számára, és megakadályozza, hogy jelentős módosításokat hajtson végre a modellkódon.

A fejlesztők a nézet használata nélkül hozhatnak létre egységteszteket a nézetmodellhez és a modellhez. A nézetmodell egységtesztjei pontosan ugyanazokat a funkciókat tudják gyakorolni, mint a nézet.

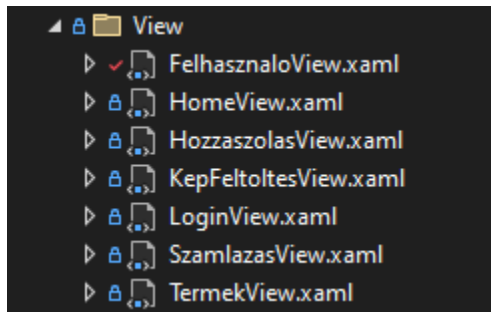
Az alkalmazás felhasználói felülete újratervezhető a nézetmodell és a modellkód érintése nélkül, feltéve, hogy a nézet teljes egészében XAML vagy C# nyelven van megvalósítva. Ezért a nézet új verziójának működnie kell a meglévő nézetmodellel.

A tervezők és a fejlesztők önállóan és egyidejűleg dolgozhatnak komponenseiken a fejlesztés során. A tervezők a nézetre összpontosíthatnak, míg a fejlesztők a nézetmodellel és a modellösszetevőkön dolgozhatnak.

Az MVVM hatékony használatának kulcsa annak megértésében rejlik, hogyan lehet az alkalmazáskódot a megfelelő osztályokba integrálni, és hogyan működnek együtt az osztályok.

Nézet

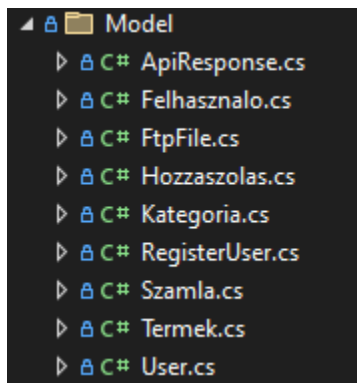
A nézet felelős a felhasználó által a képernyőn látottak szerkezetének, elrendezésének és megjelenésének meghatározásáért. Ideális esetben minden nézet XAML-ben van definiálva, korlátozott háttérkóddal, amely nem tartalmaz üzleti logikát. Bizonyos esetekben azonban a háttérkód tartalmazhat olyan felhasználói felületi logikát, amely XAML-ben nehezen kifejezhető vizuális viselkedést, például animációkat valósít meg.



45. ábra- MVVM nézet

Modell

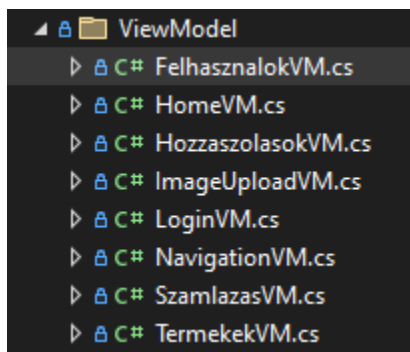
A modellosztályok olyan nem vizuális osztályok, amelyek az alkalmazás adatait foglalják magukban. Ezért a modell az alkalmazás tartománymodelljét képviseli, amely általában tartalmaz egy adatmodellt az üzleti és érvényesítési logikával együtt. A modellobjektumok közé tartoznak például az adatátviteli objektumok (DTO-k), az egyszerű régi CLR-objektumok (POCO-k), valamint a létrehozott entitás- és proxyobjektumok.



46. ábra- MVVM model

NézetModell

A nézet első kompozícióval az alkalmazás fogalmilag olyan nézetekből áll, amelyek kapcsolódnak azokhoz a nézetmodellekhez, amelyektől függenek. Ennek a megközelítésnek az elsődleges előnye, hogy megkönnyíti a lazán csatolt, egységesíthető alkalmazások létrehozását, mivel a nézetmodellek nem függenek maguktól a nézetektől. Az alkalmazás szerkezetét is könnyen megértheti a vizuális struktúrájának követésével, ahelyett, hogy nyomon kellene követnie a kód végrehajtását az osztályok létrehozásának és társításának megértéséhez.



47. ábra- MVVM nézetmodell

A nézetmodellek általában nyilvános tulajdonságokat tesznek elérhetővé a nézetből való kötéshez, amelyek megvalósítják a felületet. A gomb vezérlőelem az egyik leggyakrabban használt vezérlő, amely olyan parancstulajdonságot biztosít, amely a gombra kattintáskor hajtódik végre. ICommandCommand

Az alábbi példakód bemutatja, hogyan lehet összekötni a nézetet egy nézetmodellel:

Nézet:

```
<UserControl.DataContext>
    <local:FelhasznalokVM/>
</UserControl.DataContext>
```

48. ábra- MVVM nézet kód

```
<Button Grid.Row="0" Command="{Binding SaveCommand}" Content="Save Changes"
        x:Name="btn_save" Background="Black" HorizontalAlignment="Left"
        VerticalAlignment="Top" Margin="336,26,0,0"/>
```

49. ábra- MVVM nézet kód

NézetModell

```
public RelayCommand SaveCommand { get; private set; }

SaveCommand = new RelayCommand(execute => SaveItem(), canExecute => CanSave());
```

50. ábra- MVVM nézetmodell kód

Összefoglalás

Az MVVM-minta használatával az alkalmazás felhasználói felülete, valamint az alapul szolgáló megjelenítési és üzleti logika három külön osztályba sorolható: a nézet, amely magában foglalja a felhasználói felületet és a felhasználói felület logikáját; a nézetmodell, amely magában foglalja a megjelenítési logikát és állapotot; és a modell, amely magában foglalja az alkalmazás üzleti logikáját és adatait.

Frontend

Axios

Az Axios egy olyan könyvtár, amely segít a háttérrel való kommunikációban. Főként aszinkron HTTP-kérések küldésére REST-végpontokhoz használják. Tehát, ha az alkalmazásunknak adatokat kell lekérnie egy szerverről, az Axios segítségével könnyedén megtehetjük. Az Axios egyfajta kapcsolat a kliens (alkalmazás) és a szerver között. Amikor az alkalmazásunk adatokat szeretne lekérni, az Axios elküldi a kérést a szervernek, majd a szerver válaszol az adatokkal. Az Axios használata során a kéréseket promise-okkal kezeljük, ami azt jelenti, hogy a válasz érkezése előtt a kód nem blokkolódik, hanem folytatja a futását. Ez különösen hasznos, ha több kérést is egyszerre szeretnénk küldeni. Az Axios támogatja az interceptorokat, transzformátorokat és más hasznos funkciókat is.

Szerkezete

A szakdolgozatban bemutatjuk a **React**-ot, mint frontend keretrendszert, amelyet a webshopunk fejlesztéséhez használtunk. A React egy modern és hatékony eszköz a felhasználói felületek készítéséhez, és lehetővé teszi a komponens alapú fejlesztést és a könnyű újrahasznosíthatóságot.

A React fő jellemzője, hogy a legtöbb adatot lokálisan tárolja, ezzel minimalizálva a hálózati kérések számát. A fejlesztés során különböző package-eket is felhasználunk, hogy optimalizáljuk és kiegészítsük a folyamatot. Például a **react-router-dom** könyvtárat használjuk a navigáció kezelésére.



51.ábra – Frontend szerkezete

Források

<https://learn.microsoft.com/hu-hu/dotnet/csharp/fundamentals/tutorials/oop>

<https://learn.microsoft.com/hu-hu/shows/entity-framework-core-for-beginners/>

<https://attila.gludovatz.hu/posts/reszletesebb-ismerkedes-a-react-tel-routing-2-resz>

<https://blog.openreplay.com/using-jwt-for-authentication-in-react/>

<https://wpf-tutorial.com/hu/1/a-wpf-rol/mi-az-a-wpf/>

<https://tutorials.eu/data-transfer-objects-in-c-sharp/>

<https://www.geeksforgeeks.org/axios-in-react-a-guide-for-beginners/>

https://www.tutorialspoint.com/http/http_methods.htm?authuser=0

[Model-View-ViewModel - .NET | Microsoft Learn](#)

<https://drive.google.com/file/d/1y-Ro5HIOuAjrJHHkUCaDeaDulLu8t9/view>