

Vizsgaremek

Árajánlat gyorsan és egyszerűen
Quote quickly and easily

Kammerer Gabriella
Zibrinyiné Vendel-Mohay Luca Mária
Varjasi Tamás

Vizsgaremek adatlap

A Vizsgaremek készítői:

Neve: Kammerer Gabriella
E-mail címe: kammerer.gabriella@blathy.info

Neve: Zibrinyiné Vendel-Mohay Luca Mária
E-mail címe: vendel-mohay.luca@blathy.info

Neve: Varjasi Tamás
E-mail címe: varjasi.tamas@blathy.info

A Vizsgaremek témája:

Árajánlat készítő szoftver készítése webshopszerű megoldással, amely egyszerű, átlátható, de ügyfélközpontú szerkezetével időt és energiát takarít meg valamennyi vállalkozás részére. Jelen szoftverünk egy biztonságtechnikai telepítő cég számára van testreszabva. Külön-külön felület kerül kialakításra az ügyfeleknek és az ügyféllátogató munkatársaknak, illetve az irodai dolgozóknak.

A Vizsgaremek címe:

Árajánlat gyorsan és egyszerűen - Quote quickly and easily

Konzulens tanár: Duffka Erik Martin

Kelt: Budapest, 2023. szeptember 09

Kammerer Gabriella

Zibrinyiné Vendel-Mohay Luca
Mária

Varjasi Tamás

Duffka Erik Martin

Eredetiség nyilatkozat

Alulírott tanuló kijelentem, hogy a vizsgaremek saját és csapattársa(i)m munkájának eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült vizsgaremek részét képező anyagokat az intézmény archiválhatja és felhasználhatja.

Budapest, 2023. szeptember 09

Zibrinyiné Vendel-Mohay Luca Mária
Tanuló aláírása

Kammerer Gabriella
Tanuló aláírása

Varjasi Tamás
Tanuló aláírása

1.	Projekt ismertetése	5
2.	Felhasználói dokumentáció	5
2.1.	Webalkalmazás.....	5
2.2.	Asztali alkalmazás C# programnyelven	13
3.	Fejlesztői dokumentáció.....	15
3.1.	Munkamegosztás.....	15
3.2.	Fejlesztői környezet.....	15
3.3.	Telepítési dokumentáció	17
3.3.1.	Laravel	17
3.3.2.	React.....	17
3.3.3.	C#.....	20
3.4.	Adatbázis dokumentáció	20
3.4.1.	Diagram	20
3.4.2.	Adattáblák létrehozása.....	21
3.4.3.	Mezők részletes ismertetése.....	22
3.4.4.	Kapcsolatok indoklása, ismertetése	23
3.5.	Api.....	23
3.6.	Backend - Laravel.....	24
3.7.	Frontend - React.....	30
3.8.	Asztali alkalmazás - C#.....	33
4.	Teszt dokumentáció	44
5.	Továbbfejlesztési lehetőségek	46
6.	Összegzés.....	46
7.	Források.....	47

1. Projekt ismertetése

Az easyQuote egy árajánlat készítő webalkalmazás webshopszerű megoldással. Célja, hogy az árajánlat a lehető legrövidebb idő alatt elkészüljön és minél kevésbé terhelje a vállalkozás erőforrásait.

Az árajánlat készítése szerves része egy vállalkozásnak. Sok esetben az árajánlatból nem lesz megrendelés, ugyanakkor jelentős idő ráfordítással jár. Pontosnak, precíznek kell lennie, mivel az esetleges megrendelés elszámolásánál az árajánlat az elszámolás alapja és amennyiben nem elég precíz, a vállalkozásnak veszteséget okozhat.

Fő szempont az átláthatóság és a felszerelésre kerülő eszközök pontos ismertetése is, annak érdekében, hogy az ügyfél értelmezni tudja az árajánlatot és pontosan össze tudja hasonlítani egy másik ajánlattal.

Az eszközök megtekinthetősége is fontos szerepet játszik egy árajánlatban, mivel nagyon sok esetben lakberendezők/belsőépítészek tervei alapján kerülnek kialakításra a belső terek és ez befolyásolja az eszközök választását például a kezelőegységek tekintetében.

Az árajánlat tehát nemcsak pénzügyi aspektusból fontos, hanem a vállalkozás teljes működésének és stratégiájának szerves részét képezi. Megfelelő árajánlat készítésével és kezelésével a vállalkozás hatékonyan versenyezhet a piacon, növelheti a bevételét, és hosszú távú sikerre tehet szert.

2. Felhasználói dokumentáció

Az alkalmazás használatával a felhasználó árajánlat kérést tud indítani, nyomon tudja követni az árajánlat folyamatát, illetve a helyszínen lévő kolléga segíteni tud a megfelelő eszköz kiválasztásában.

A cég alkalmazottjai viszont egy gyors és egyszerű felületen, pár kattintással tudják adminisztrálni a helyszíni felmérést és a kiválasztott eszközöket.

Az alkalmazás további előnye, hogy csak egy böngészőre van szükség a használatához, így szinte bármilyen eszközön használható.

Az irodában dolgozó kollégák egy felületről tudják kezelni a megrendeléseket és termékeket.

2.1. Webalkalmazás

Honlapunkat felkeresve az alábbi felület fogadja az érdeklődőket, ügyfeleket és az alkalmazottakat. Rövid bemutatkozás után a főbb termékköröket ismertetjük.



Biztonsági rendszerek

Üdvözljük az **easyQuote** weboldalán! Mi vagyunk az Ön biztonsága és védelme szakértői. Cégünk elkötelezett amellett, hogy magas minőségű kamerarendszerekkel és riasztóberendezésekkel segítsük ügyfeleinket otthonuk és vállalkozásuk biztonságának megőrzésében. Vállaljuk komplett riasztó és kameraberendezések teljes körű telepítését és karbantartását. Kérje **ajánlatunkat!**



Kamerák

Kiemelkedő minőségű kameráink segítségével lehetősége lesz 24/7 figyelni és rögzíteni az eseményeket. Legyen szó otthonról vagy irodáról, kameráink éles képminőséget és éjszakai látást kínálnak, hogy mindig tudja, mi történik. Intelligens funkcióink, mint a mozgásérzékelés és a távoli hozzáférés, segítenek még hatékonyabb és kényelmesebb védelmet biztosítani. Fedezze fel kamerarendszereink széles választékát, amelyek alkalmazkodnak az Ön egyedi igényeihez, és biztonságban érezheti magát bárhol is legyen.



Riasztók

Fejlett és megbízható riasztórendszereket forgalmazunk, amelyek védelmet nyújtanak a betörések, tűz és egyéb veszélyek ellen. Intelligens érzékelőink és távoli hozzáférhetőségük révén bármikor figyelheti és vezérelheti riasztórendszerét, akár távolról is. Rugalmas és testreszabható megoldásainkkal minden egyedi igényt kielégítünk. Válassza riasztórendszereinket, és aludjon nyugodtan éjszaka, tudván, hogy otthona és értékei védelmet élveznek.



Garancia

Szolgáltatásaink magukba foglalják a konfigurációt, telepítést és a rendszerek karbantartását is. Minden telepítésre **2 év** teljes körű garanciát vállalunk. Szakértőink mindig rendelkezésre állnak, hogy válaszoljanak kérdéseire és tanácsot adjanak az optimális megoldások kiválasztásában.

Ajánlatot kérek!

Linkek

- › Főoldal
- › Ajánlat
- › Rólunk
- › Kapcsolat

easyQuote

Magyarország
Budapest 1234
Elfelejtetem utca 5.
☎ +36 20 123 4567
✉ info@easyquote.hu

A navigációs sáv (navbar) bal oldalán látható a program logója, jobboldalon a látogató, illetve bejelentkezett felhasználó szerepkörének megfelelő menüpontok láthatók.

A lábjegyzetben (footer) található az oldalon a könnyebb eligazodást segítő linkek, míg jobb oldalán az elérhetőségek.

Linkek <ul style="list-style-type: none"> > Főoldal > Ajánlat > Rólunk > Kapcsolat 	easyQuote <p>Magyarország Budapest 1234 Elfelejtetem utca 5. ☎ +36 20 123 4567</p>
---	---

Az ajánlat menüpontban az ajánlatkérés menetét ismertetjük.

Az ajánlatkérés menete:

- Regisztráljon oldalunkra a **Regisztráció** linkre kattintva
- Kattintson a **Rendelések** menüpontra
- Kattintson az **Új ajánlatkérés** gombra, majd töltsse ki az űrlapot
- Kollégáink hamarosan felveszik önnel a kapcsolatot további adategyeztetésre
- A telefonos megbeszélés után a **Rendelések** menüpont alatt láthatja a rendelés részleteit

Rólunk szekcióban bemutatjuk a 404-es hiba kezelését amennyiben hibás, nem létező oldalt keresne fel a látogató.



Error - 404

A kért oldal nem található

Vissza főoldalra

Regisztráció a felületen egy nagyon gyors és egyszerű folyamat. Minden felhasználónak szüksége van egy felhasználói fiókra, szerepkört az asztali alkalmazásban lehet változtatni.

easyQuote

 [Regisztráció](#)  [Bejelentkezés](#)

Regisztráció

Felhasználónév:*

E-mail cím:*

Jelszó: * **

Jelszó megerősítése:*

* A csillaggal jelölt mezők kitöltése kötelező!



** Min. 8 karakter, min. egy speciális karakter

Regisztráció

Már regisztrált? [Jelentkezzen be!](#)

Bejelentkezés az email címmel és jelszóval történik.

easyQuote

 [Regisztráció](#)  [Bejelentkezés](#)

Bejelentkezés

E-mail cím:

Jelszó:

Bejelentkezés

Még nem regisztrált? [Regisztráljon!](#)

Regisztrációnál hibás adatok megadása esetén hibaüzenet jelenik meg az oldalon.

Regisztráció

Felhasználónév:*

E-mail cím:*

Jelszó: **

Jelszó megerősítése:*

* A csillaggal jelölt mezők kitöltése kötelező!

** Min. 8 karakter, min. egy speciális karakter

Regisztráció

Már regisztrált? [Jelentkezzen be!](#)

A mezők kitöltése helytelen!

Bejelentkezés esetén hibás e-mail cím és/vagy jelszó esetén szintén figyelmeztetést kapunk.

Bejelentkezés

E-mail cím:

Jelszó:

Bejelentkezés

Még nem regisztrált? [Regisztráljon!](#)

A megadott e-mail cím, vagy jelszó helytelen!

Sikeres belépést követően csak a menü változik meg a korábban bemutatott módon. Felhasználó módosíthatja az adatait, indíthat új ajánlatkérést, illetve lekérdezheti a saját ajánlatait.

Adatok megváltoztatása.

easyQuote

Rendelések

Adataim

Kijelentkezés

Adatok megváltoztatása:

Felhasználónév:

E-mail cím:

Új jelszó:

Új jelszó megerősítése:

Mentés

Rendelések lekérdezése.

easyQuote

Rendelések

Adataim

Kijelentkezés

Rendelések:

Új ajánlatkérés

Nincs megjeleníthető rendelés.

Új ajánlat kérése.

easyQuote

Rendelések

Adataim

Kijelentkezés

Új ajánlat igénylése:

Irányítószám:*

Város:*

Cím:*

Telefonszám:*

A csillaggal jelölt mezők kitöltése kötelező!

Küldés

Ajánlatkérés elküldése után az ügyfélnek semmilyen más teendője nincs, kollégánk keresni fogja adategyeztetés miatt, majd ezután kerül sor a helyszíni felmérés és megrendelés összeállítására.

Technikus kollégának megjelenik a termékek menüpont is, rendeléseknél az összes visszaigazolt megrendelés látszik egymás alatt.

easyQuote

Termékek Rendelések Adataim Kijelentkezés

Rendelések:


Felhasználó:	tomi@mail.com
IRSZ:	1011
Város:	Budapest
Cím:	Fő utca 1
Telefonszám:	0611111111
Fizetési határidő:	
Felmérés:	-
Teljesítve:	-
Létrehozva:	2023-09-14 09:52:42
Módosítva:	2023-09-14 09:52:42
Árkalkuláció:	

Termékek oldalon van lehetőség a megrendelni kívánt termékek összeállítására a megrendelést kiválasztva.

easyQuote

Termékek Rendelések Adataim Kijelentkezés

Módosítani kívánt rendelés: Válasszon...
Keresés: név szerint Válasszon...
tomi@mail.com - Fő utca 1




Kezelő - TM50

Multifunkciós érintőképperrys kezelő 1 címezhető z...

83 983 Ft

Hozzáadás




Kezelő - TM70

Paradox TM70 multifunkciós érintőképperrys kezelőe...

107 206 Ft

Hozzáadás




Kezelő - K656 W

Érintógombos LCD kezelő LED háttérvilágítással, mo...

62 212 Ft

Hozzáadás




Kezelő - K32LCD+

Két particiós 32-zónás LCD kezelő. Programozható c...

44 432 Ft

Hozzáadás




Kezelő - K641+

Egy vagy több particiós 32-zónás LED kezelő (max 8) buszo...

56 406 Ft

Hozzáadás



Kezelő - K32+

Két particiós 32-zónás LED kezelő, 1 kezelő zóna b...

34 272 Ft

Hozzáadás

Keresés funkcióval gyorsan le lehet szűkíteni a keresést akár kategória, alkategória és név szerint is.

easyQuote

Termékek
Rendelések
Adataim
Kijelentkezés

Módosítani kívánt rendelés: tomi@mail.com - Fc

Keresés:
név szerint
hátter

Háttértárak - Western Digital HDD 3,5" ...

Western Digital HDD 3,5" WD SATA3 Purple 1TB WD10...

26 543 Ft

1

Hozzáadás

Részletek

Háttértárak - Western Digital HDD 3,5" ...

Western Digital HDD 3,5" WD SATA3 Purple 3TB WD30...

49 022 Ft

1

Hozzáadás

Részletek

Háttértárak - Western Digital WD42PURZ 4...

Western Digital WD42PURZ 4TB HDD 3,5" Purple, HDD...

51 395 Ft

1

Hozzáadás

Részletek

Háttértárak - Western Digital WD63PURZ 6...

Western Digital WD63PURZ 6TB HDD 3,5" Purple, HDD...

64 611 Ft

1

Hozzáadás

Részletek

A kiválasztott termék részleteire kattintva a termékről bővebb információ jelenik meg.

easyQuote

Termékek
Rendelések
Adataim
Kijelentkezés

Keresés: név sze

Háttértárak - Western Digital HDD 3,5" ...

Western Digital HDD 3,5" WD SATA3 Purple 1TB WD10...

26 543 Ft

1

Hozzáadás

Részletek

Háttértárak - Western Digital HDD 3,5" WD SATA3 Purple 1TB WD10PURZ

Kategória: riaszto

Alkategória: hattertarak

Leírás: Western Digital HDD 3,5" WD SATA3 Purple 1TB WD10PURZ, HDD, 5400 rpm, SATA III, 64 MB, Surveillance.

26 543 Ft

Bezáras

A kiválasztott termékek megjelennek a megrendelés árkalkuláció részére kattintva.

easyQuote

Termékek
Rendelések
Adataim
Kijelentkezés

Megrendelt termékek:

Terméknév	Mennyiség	Ár	Törlés
Háttértárak - Western Digital HDD 3,5" WD SATA3 Purple 1TB WD10PURZ	1	26 543 Ft	
Kezelő - TM50	1	83 983 Ft	
Központ - EVO192	1	53 140 Ft	
Szíréná - VL-640L	1	14 984 Ft	
Összesen:		178 650 Ft	

* az árak tájékoztató jellegűek, az árváltozás jogát fenntartjuk

A megrendelések oldalon, felmérés és teljesítés után változik a státusz, a megrendelés ilyenkor kikerül a módosítható megrendelések listájából és a megrendelés véglegesnek tekinthető.

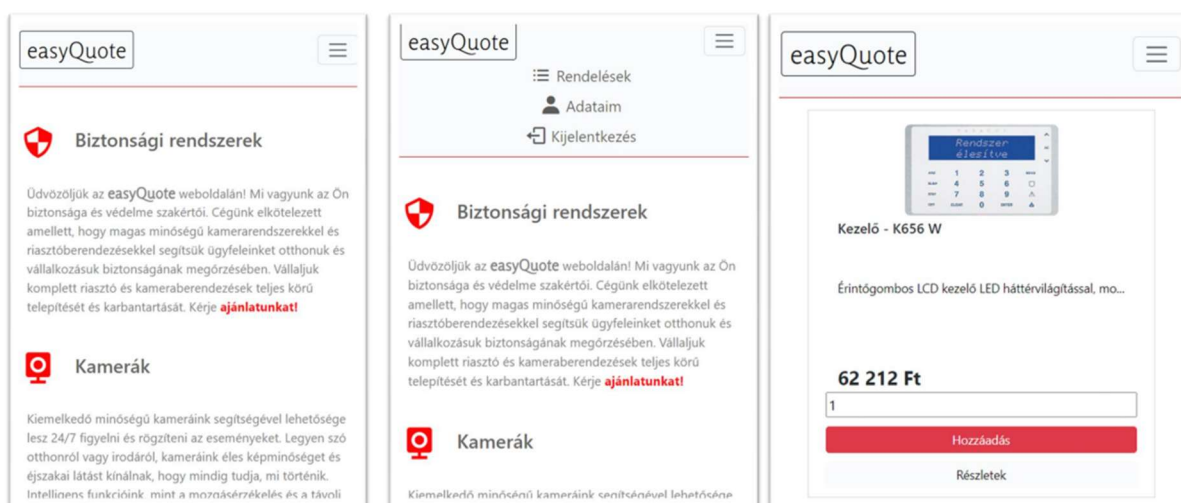
easyQuote

Termékek
Rendelések
Adataim
Kijelentkezés

Rendelések:

Felhasználó:	tomi@mail.com
IRSZ:	1011
Város:	Budapest
Cím:	Fő utca 1
Telefonszám:	0611111111
Fizetési határidő:	
Felmérés:	Elvégezve
Teljesítve:	-
Létrehozva:	2023-09-14 09:52:42
Módosítva:	2023-09-14 09:52:42
Árkalkuláció:	

A kor elvárásainak és a „Mobile First” szemléletnek megfelelően a teljes felület reszponzívan került kialakításra.



2.2. Asztali alkalmazás C# programnyelven

Felület ismertetése

A projektnek ez a része az irodában ülő alkalmazott számára készült, aki ezen a felületen tud bizonyos funkciókat jóváhagyni, termékeket feltölteni. Mivel a programunk nem egy webáruház, hanem egy árajánlat készítő szoftver melyet a telepítő a helyszínen tablet segítségével használ így szükségét éreztük egy irodai alkalmazásnak is, mely magasabb biztonsági tényezőkkel rendelkezik és a termékek is könnyebben feltölthetők így. Projektünk ezen része .NET keretrendszeren alapuló Windows Presentation Foundation (WPF)

technológiát használ a felhasználói felület kialakításához. Az alkalmazás összesen 5 ablakból, 3 lapból (page) és a hozzájuk tartozó '.cs' kiterjesztésű C# kódot tartalmazó fájlokból, és 4 db szintén C# fájlokban található segédosztályból áll.

Ablakok:

- **MainWindow.xaml + MainWindow.xaml.cs:** Ezen a felületen tud a felhasználó az alkalmazásba bejelentkezni.
- **easyQuoteDashboard.xaml + easyQuoteDashboard.xaml.cs:** Ez a program fő ablaka és kerete.
- **NewUser.xaml + NewUser.xaml.cs:** Ez az ablak szolgál az új irodai felhasználók felvitelére.
- **AddNewProduct.xaml + AddNewProduct.xaml.cs:** Itt lehet az alkalmazáshoz új termékeket hozzáadni.
- **About.xaml + About.xaml.cs:** Megjeleníti a az alkalmazást fejlesztő névjegy információit és verziószámát.

Lapok (pages):

1. **Orders.xaml + Orders.xaml.cs:** A bejelentkezés után alapértelmezetten betöltődő ablak, ahol a megrendeléseket lehet szerkeszteni, törölni vagy a számlázást elindítani.
2. **Products.xaml + Products.xaml.cs:** Menüben a **termékek** kiválasztása után megjelenő lap, mely a bal oldalintéblázatban az összes termék nevét, kategóriáját, alkategóriáját, kép URL-ét, árát, státuszát és leírását tartalmazza. Termékeket itt lehet feltölteni és módosítani.
3. **Users.xaml + Users.xaml.cs:** Ezen a lapon a felhasználókat lehet kezelni jogosultági szinttől függően csak nevet és e-mailcímet vagy akár új felhasználót felvenni, törölni és jelszót módosítani (csak a dolgozóknak).

3. Fejlesztői dokumentáció

3.1. Munkamegosztás

Csaptunk három tagból állt, Kammerer Gabriella Frontend, Zibrinyiné Vendel-Mohay Luca Mária C# és Varjasi Tamás Backend részért felelt.

Közös munka során a Microsoft Office által biztosított eszközökkel dolgoztunk.

Kapcsolatot, egyeztetéseket a Teams nevű programmal végeztük, megosztott közös dokumentummal pedig nyomon tudtuk követni, hogy ki mit módosított a bemutatón. Észlelt hibákat a OneNote alkalmazás segítségével szintén egy megosztott jegyzetben kezeltük.

A kód közös részén a GitHub segítségével tudunk dolgozni.

3.2. Fejlesztői környezet

A weboldalhoz a Visual Studio Code kódszerkesztőt, az asztali alkalmazáshoz a Visual Studio fejlesztőkörnyezetet használtuk. A Visual Studio Code alkalmas a Laravel backend és a React frontend fejlesztéséhez. Visual Studio alkalmas az asztali alkalmazás megírásához. Továbbá az adatbázis kezeléshez a XAMPP-ot használtuk.

Visual Studio Code



A Visual Studio Code (rövidítve: VSCode vagy VS Code) ingyenes, nyílt forráskódú kódszerkesztő, melyet a Microsoft fejleszt Windows, Linux és macOS operációs rendszerekhez. Támogatja a hibakeresőket, valamint beépített Git támogatással rendelkezik, továbbá képes az intelligens kódkiegészítésre az IntelliSense segítségével. A VSCode-ban a felhasználók megváltoztathatják a kinézetet (témát), megváltoztathatják a szerkesztő gyorsbillentyű-kiosztását, az alapértelmezett beállításokat és még sok egyebet. Támogatja a kiegészítőket, melyek segítségével további funkciók, testreszabási lehetőségek érhetőek el.

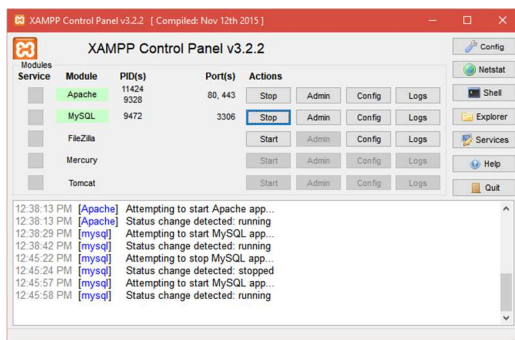
A programot 2015. április 29-én jelentették be a Build 2015 konferencián (a Microsoft éves fejlesztői konferenciája), november 18-án pedig MIT licence alatt nyilvánossá tették a forráskódját.

Visual Studio



A Visual Studio a Microsoft több programozási nyelvet tartalmazó fejlesztőkörnyezete, amely az évek során egyre több új programnyelvvel bővült. Jelenleg a F#, C++, C# és Visual Basic programozási nyelveket, valamint az XML-t támogatja. A csomag része még a MASM (Microsoft Macro Assembler) is, ami részleges assembly támogatást biztosít.

XAMPP



A XAMPP egy szabad és nyílt forrású platformfüggetlen webservert-szoftvercsomag, amelynek legfőbb alkotóelemei az Apache webservert, a MariaDB (korábban a MySQL[2]) adatbázis-kezelő, valamint a PHP és a Perl programozási nyelvek értelmezői (végrehajtó rendszerei). Ez a szoftvercsomag egy integrált rendszert alkot, amely webes alkalmazások készítését, tesztelését és futtatását célozza, és ehhez egy csomagban minden szükséges összetevőt tartalmaz. A rendszer egyik nagy előnye az összehangolt elemek könnyű telepíthetősége.

A XAMPP egyetlen tömörített (zip, tar, 7z vagy exe formátumú) állományba van csomagolva, telepítéséhez mindössze egy a fájlt kell letölteni és futtatni. A telepítés elvégzi az alapbeállításokat, azokon csak nagyon keveset vagy éppen semmit nem kell változtatni, ezután a rendszer készen áll a webservert és a mintaalizások futtatására. A XAMPP-csomagot rendszeresen frissítik, így az mindig az Apache, MariaDB, PHP és Perl legújabb változatát tartalmazza, valamint más kiegészítőket is, mint pl. az OpenSSL és a phpMyAdmin. Verziószáma a csomagban található PHP-értelmező verziójával egyezik meg (jelenleg az 5.5.x, 5.6.x és 7.0.x PHP-verziókhoz egyaránt elérhető különálló csomag).

A XAMPP-csomag több példányban is telepíthető a gazdagépre, a különálló telepítések képesek önmagukban, a többi csomag megzavarása nélkül működni, ráadásul az installált példányok egyszerűen átmásolhatók egy másik gépre.

A szoftver három változatban is elérhető: teljes, általános és kisméretű csomagok léteznek.

Hivatalosan a XAMPP tervezői az eszközt egy fejlesztőrendszernek szánták, amellyel a web-tervezők és programozók internetes kapcsolat nélkül fejleszthetik és tesztelhetik alkalmazásaikat. Ennek érdekében több fontos biztonsági funkció alapértelmezésben ki van kapcsolva a csomagban, ennek ellenére a XAMPP szoftvert valódi webes szolgáltatóként is használják. A csomag egy külön eszközt tartalmaz a legfontosabb részek jelszavas védelmének beállítására.

A XAMPP többféle adatbázis-kezelő használatát is támogatja, ilyenek pl. a MySQL és az SQLite és mások.

A XAMPP telepítése után a helyi gép (a localhost) hálózati gépként is hozzáférhetővé válik, pl. FTP kliensprogrammal elérhető. Alkalmazható pl. FileZilla fájlkezelő, telepíthetők tartalomkezelő rendszerek, mint a WordPress vagy a Joomla!. A localhost közvetlenül is kezelhető egyes HTML vagy általános szövegszerkesztőkből, az FTP protokoll használatával. Az alapbeállítás szerinti FTP felhasználónév a „newuser”, jelszava „wampp”. Az alapbeállítás szerinti MySQL felhasználónév a „root”, jelszó nélkül.

3.3. Telepítési dokumentáció

3.3.1. Laravel

Terminál segítségével az alkalmazás könyvtárában az alábbi műveleteket kell elvégezni, illetve parancsokat kell kiadni.

`-.env.example átnevezése .env -re`

Composer telepítése:

`-composer install`

Alkalmazás használatához szükséges kulcs generálása:

`-php artisan key:generate --ansi`

Adatbázis létrehozása migrálás segítségével:

`-php artisan migrate`

Indítás

`-php artisan serve`

3.3.2. React

Terminál segítségével a React könyvtárában az alábbi műveleteket kell elvégezni, illetve parancsokat kell kiadni.

`-.env.example átnevezése .env -re`

npm telepítése:

`-npm install`

axios telepítése:

`-npm install axios`

indítás:

-npm run dev

Az easyquote weboldal megnyitásához egy webböngészőt (browser) kell alkalmaznunk, itt külön telepítésre nincs szükség. A weboldal URL címét csak be kell írni az URL címsorba és ezt követően megnyílik a weboldal.

A legnépszerűbb böngészők jelenleg:

Google Chrome



A Google Chrome a Google által fejlesztett ingyenesen használható, számítógépeken, táblagépeken és okostelefonokon futó webböngésző, ami a Blink (iOS-en a WebKit) böngészőmotort használja. 2008 szeptemberében jelentették meg az első publikus bétaverziót, az első stabil verziót 2008. december 11-én adták ki. A Chrome nevét a böngészők grafikus felületének nem létfontosságú részeiről (angolul „chrome”, kb. csicsa) kapta; a névválasztás ironikus, mivel épp arra törekedtek, hogy ezek aránya minimális maradjon. 2016 decemberében a Chrome világszinten a legnépszerűbb böngésző volt, 53,61%-os globális részesedéssel.

Mozilla Firefox



A Mozilla Firefox (hivatalos rövidítéssel Fx, vagy gyakran FF) egy nyílt forráskódú, ingyenes webböngésző program, amit a Mozilla Alapítvány fejleszt. Magyarország piacvezető böngészője volt 2009-től, míg a Chrome 2013 októberében át nem vette a piacvezető szerepét. Globális szinten pedig a harmadik legtöbbet használt böngésző az Internet Explorer után, 13,29%-os részesedéssel. A nyílt forráskódú Gecko böngészőmotort használja, amely számos jelenlegi internetes szabványt támogat, továbbá néhány olyan elemet is, ami ezen szabványok része lehet a közeljövőben.

A Firefox funkciói között szerepel a füles böngészés, a helyesírás-ellenőrző, az azonnali keresés, az RSS-kliens (élő könyvjelzők), a privát böngészés és a helymeghatározás (Geolocation). Tartalmaz letöltéskezelőt, továbbá beépített keresőrendszert is, ami a felhasználó által megadott keresőmotort használja. Képességei bővíthetők külső fejlesztők által készített kiegészítők segítségével. A kiegészítők széles választéka és mennyisége számos Firefox-felhasználó legfontosabb érve a program használata mellett.

A Firefox többplatformos alkalmazás, a Microsoft Windows több változatán, OS X-en, Linuxon és számos más Unix-alapú operációs rendszeren is fut. Legutóbbi stabil kiadása, a 26.0-s verzió 2013. december 10-én jelent meg. Forráskódja szabad, a GPL/LGPL/MPL hármas licenc alatt érhető el. A Mozilla Corporation által kiadott, Firefox 3.0.6-os verziónál régebbi

telepítőkészletek telepítése előtt el kell fogadni a Mozilla Corporation végfelhasználói licencszerződését (EULA). A Firefox 3.0.6 és az újabb verziók telepítőkészletei az MPL alatt jelennek meg. A Firefox név és logó, egyéb nevek és logók mellett, védjegyoltalmat élvez; ezek felhasználása korlátozott, nem esnek a fenti licencek hatálya alá.

Microsoft Edge



A Microsoft Edge (gyakran egyszerűen Edge) a Microsoft által kiadott kereskedelmi, platformfüggetlen böngésző. Első változatát 2015-ben adták ki a Windows 10 és az Xbox One részeként, majd később más platformokra (Androidra, iOS-re, macOS-re, régebbi Windows-verziókra (a Windows 7, Windows Server 2008 R2 és későbbi változatainak 2023 elejéig voltak támogatottak), valamint legutóbb Linuxra) is elérhetővé tették a Google Chromium projektjének részeként. Az Edge az Internet Explorer utódjaként jött létre.

Az Edge eredetileg a Microsoft saját EdgeHTML böngésző- és Chakra JavaScript-motorjával jött létre. 2018 végén bejelentették, hogy az Edge-nek egy új, Chromium-alapú változata fog megjeleni Blink böngésző- és V8 JavaScript-motorral. Ez 2020 januárjában jelent meg számítógépeken, 2021-ben Xboxon. A Microsoft azóta megszüntette az eredeti böngésző támogatását, és a Windows 11-ben ez az alapértelmezett böngésző a Google Chrome-mal való kompatibilitásért.

2022 májusában a StatCounter szerint a Microsoft Edge volt a második legtöbbször használt böngésző a Safari előtt (egyes országokban, például az Amerikai Egyesült Államokban a harmadik, ahol az asztali változatának 14% volt a részesedése a Safari 16%-os részesedéséhez képest). 2022 szeptemberében a világon 11% volt a részesedése.

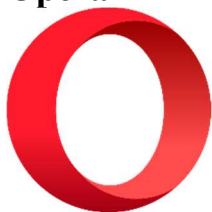
Safari



A Safari egy webböngésző, melyet az Apple Inc. fejleszt a OS X és iOS operációs rendszeréhez. Első kiadása egy nyilvános béta volt 2003. január 7-én. Ez az alapértelmezett böngésző a Mac OS X 10.3 Panther és a későbbi Mac OS X változatokban. A Safari iOS-en szintén natív böngésző.

A Safari Microsoft Windows operációs rendszerre írt első kiadása 2007. június 11-én jelent meg, és támogatta a Windows XP Service Pack 2 vagy újabb Windows változatokat, de ez már megszűnt. A 2012. május 9-én megjelent Safari 5.1.7 volt az utolsó Windows operációs rendszerre készült változat.

Opera



Az Opera egy ingyenes, igen kis méretű, több platformon használható internetes tevékenységeket összefogó program: elsősorban egy webböngésző, de ezen kívül tartalmaz beépített levelező klienst, hírolvasó klienst, címjegyzéket, IRC alapú csevegő klienst, RSS és Atom hírolvasót, minialkalmazásokat (widgeteket), valamint hagyományos és BitTorrent alapú letöltésvezérlőt is. Ezek a kiegészítő funkciók mindaddig rejtve maradnak az Operában, amíg a felhasználó nem kezdi el használni valamelyiket.

Az Operát az Opera Software, egy Oslóban jelenleg is működő norvég cég készíti. A böngésző korábban a Presto elnevezésű motorját (rendering engine) használta, később azonban a Google által fejlesztett Blink motorra váltott. Az Opera vezető pozíciót szerzett az okostelefonok és PDA készülékek böngészői között a kis képernyős megjelenítés technológiájának köszönhetően. Az Operát és az Opera Minit C++ nyelven írták.

Az Opera zárt forráskódú, de ingyenesen letölthető és használható program. A jelentősebb böngészőkhöz mérten igen kis méretű – annak ellenére, hogy funkcionalitása sokkal szélesebb körű, magas szinten követi a webes szabványokat. Ezt az irányvonalat jól jellemzi, hogy több alkalmazottja tagja a W3C konzorciumnak. A böngésző biztonság tekintetében is kiemelkedő eredményeket ért el, a többi böngészőhöz képest jóval kevesebb biztonsági hibát fedeztek fel benne, és ezeket is nagyon gyorsan kijavították.

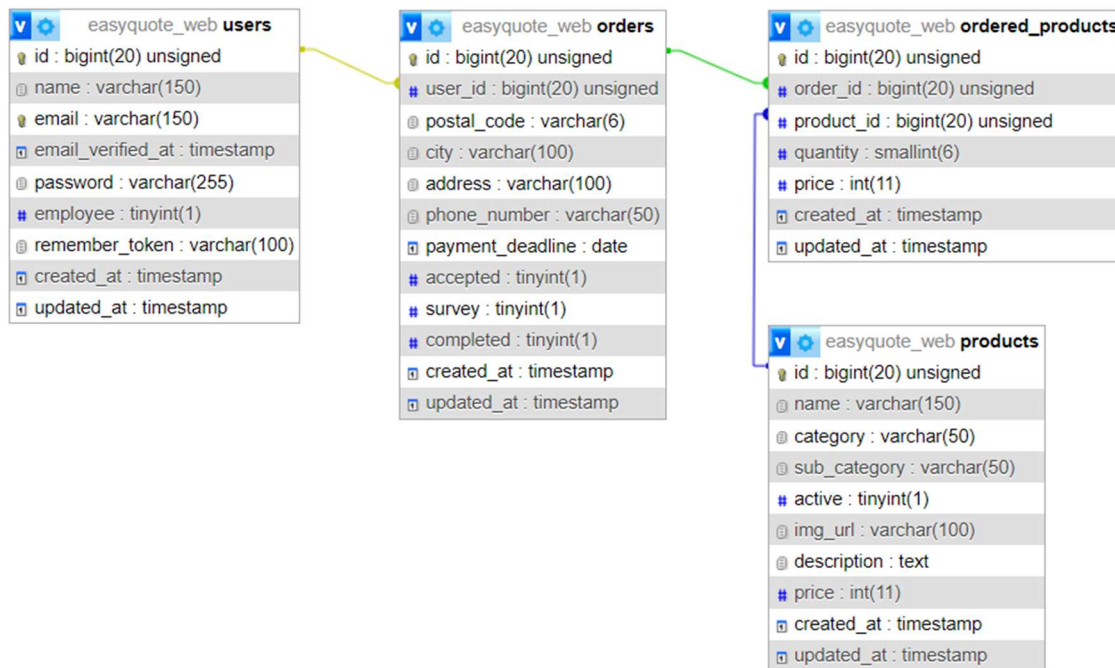
Az Operának léteznek speciális verziói mobiltelefonokra és más eszközökre is. A böngésző mobil változata, az Opera Mini minden Java-képes mobiltelefonon elérhetővé teszi a böngészést, használata ingyenes.

3.3.3. C#

Az asztali alkalmazást nem szükséges telepíteni, használatához az easyquoteDesktop\bin\Debug mappában található easyquoteDesktop.exe fájlt kell elindítani. A program működéséhez elengedhetetlen az adatbázis elérésének biztosítása és az easyQuote backend (laravel) futtatása az API-k működéséhez.

3.4. Adatbázis dokumentáció

3.4.1. Diagram



3.4.2. Adattáblák létrehozása

Létrehozás előtt szükséges beállítani az adatbázis elérést a .env fájlban.

```

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=easyquote_web
DB_USERNAME=root
DB_PASSWORD=
  
```

Az adattáblák létrehozása, módosítása a „php artisan migrate” paranccsal történik, ilyenkor a migrációs fájlok sorrendben futnak le, ezért fontos a nevekben szerepeltetni az időpontot, amennyiben ezt nem tesszük előfordulhat, hogy hibát kapunk, mert még nem hozott létre olyan táblát amelyekre egy másik hivatkozik.

Adattábla migrációs fájlok: database/migrations

Következőket állíthatjuk be („orders” tábla a példa):

- mezők típusa, kulcs mező vagy sem, felvehet-e NULL értéket, default értéket, adat max. hossza

- amennyiben tartalmaz kulcsmezőt, abban az esetben melyik tábla melyik mezőjével kösse össze és mi legyen vele törlés után

```
public function up(): void
{
    Schema::create('orders', function (Blueprint $table) {
        $table->id();
        $table->unsignedBigInteger('user_id')->nullable();
        $table->foreign('user_id')->references('id')->on('users')->onDelete('SET NULL');
        $table->string('postal_code', 6);
        $table->string('city', 100);
        $table->string('address', 100);
        $table->string('phone_number', 50);
        $table->date('payment_deadline')->default(null)->nullable();
        $table->boolean('accepted')->default(false);
        $table->boolean('survey')->default(false);
        $table->boolean('completed')->default(false);
        $table->timestamps();
    });
}
```

3.4.3. Mezők részletes ismertetése

users

id	bigint(20)	UNSIGNED	NOT NULL
name	varchar(150)	NOT NULL	
email	varchar(150)	NOT NULL	
password	varchar(255)	NOT NULL	
employee	tinyint(1)	NOT NULL	DEFAULT 0
remember_token	varchar(100)	DEFAULT NULL	

Az **employee** mező értéke szabályozza, hogy egy-egy felhasználónak milyen jogosultságai vannak a webalkalmazásban, pl, megjelenik-e a termékek menüpont illetve módosíthatja-e a megrendelést, asztali alkalmazásnál nincs szükség ilyen szabályozásra, mert azt csak az arra jogosult személyek használhatják.

orders

id	bigint(20)	UNSIGNED	NOT NULL
user_id	bigint(20)	UNSIGNED	DEFAULT NULL
postal_code	varchar(6)	NOT NULL	
city	varchar(100)	NOT NULL	
address	varchar(100)	NOT NULL	
phone_number	varchar(50)	NOT NULL	
payment_deadline	date	DEFAULT NULL	
accepted	tinyint(1)	NOT NULL	DEFAULT 0
survey	tinyint(1)	NOT NULL	DEFAULT 0
completed	tinyint(1)	NOT NULL	DEFAULT 0

Az **accepted**, **survey**, **completed** mezők tárolják a megrendelés státuszát.

products

id	bigint(20)	UNSIGNED NOT NULL
name	varchar(150)	NOT NULL
category	varchar(50)	NOT NULL
sub_category	varchar(50)	NOT NULL
active	tinyint(1)	NOT NULL DEFAULT 1
img_url	varchar(100)	DEFAULT NULL
description	text	NOT NULL
price	int(11)	NOT NULL

ordered_products

id	bigint(20)	UNSIGNED NOT NULL
order_id	bigint(20)	UNSIGNED DEFAULT NULL
product_id	bigint(20)	UNSIGNED DEFAULT NULL
quantity	smallint(6)	NOT NULL
price	int(11)	NOT NULL

3.4.4. Kapcsolatok indoklása, ismertetése

A **users** tábla kulcsa az **id** mező „egy a többhöz” kapcsolattal csatlakozik az **orders** tábla idegen kulcsához a **user_id** mezőhöz, mert egy felhasználónak több megrendelése is lehet.

Az **orders** tábla kulcsa az **id** mező „egy a többhöz” kapcsolattal csatlakozik az **ordered_products** tábla idegen kulcsához az **order_id** mezőhöz, mert egy megrendeléshez több megrendelt termék is tartozhat.

Az **products** tábla kulcsa az **id** mező „egy a többhöz” kapcsolattal csatlakozik az **ordered_products** tábla idegen kulcsához az **product_id** mezőhöz, mert egy termék több megrendel termékhez is tartozhat.

3.5. Api

Az API **végpontok**, olyan címek, amelyeken keresztül a különböző kliensek hozzáférést kapnak a szoftver utasításkészletéhez.

Alkalmazásunk a következő főbb végpontokkal rendelkezik:

api/logout: kijelentkezés

api/user: felhasználói adatok

api/signup: regisztráció

api/login: bejelentkezés
api/products: termékek kezelése
api/orders: rendelések kezelése
api/upload: termékhez tartozó fényképek feltöltése

Metódusok:

GET: adatok lekérdezésére szolgál

POST: adatokat tudunk beküldeni a szervernek

APIRESOURCE: az összes CRUD művelet

Api útvonalak: routes/api.php

```
Route::middleware('auth:sanctum')->group(function () {
    Route::post('/logout', [AuthController::class, 'logout']);
    Route::get('/user', function (Request $request) {
        return $request->user();
    });
    Route::apiResource('/users', UserController::class);

    Route::apiResource('/products', ProductController::class);
    Route::get('/products/name/{name}', [ProductController::class, 'searchByName']);
    Route::get('/products/category/{category}', [ProductController::class, 'searchByCategory']);
    Route::get('/products/subCategory/{subCategory}', [ProductController::class, 'searchBySubCategory']);

    Route::apiResource('/orders', OrderController::class);
    Route::get('/openorders', [OrderController::class, 'openOrders']);

    Route::apiResource('/ordered_products', OrderedProductController::class);
    Route::get('/ordered_products/list/{id}', [OrderedProductController::class, 'orderedProductList']);
});

Route::post('/signup', [AuthController::class, 'signup']);
Route::post('/login', [AuthController::class, 'login']);

Route::post('/upload', [PhotoController::class, 'upload']);
```

Middleware alá tartoznak azok az API-k, amelyekhez kell autentikáció.

A legutolsó API útvonal a fotó feltöltés, az útvonal elejéről hiányzik a "/" jel, mert azt a C# kódja tartalmazza.

Az „/openorders” útvonalnak azért nincs „/orders” előtagja, mert paraméter nélküli függvény. A többi egyedi útvonal használható a saját gyökerútvonalával, mert paraméteres függvények. Az egyedi útvonalak meghívásánál az utolsó paraméter jelöli a függvény nevét melyet az útvonalhoz rendelt Controller fájlban kell használnia. Pl: a következőben az “orderedProductList” a függvény neve, amit keres

```
Route::get('/ordered_products/list/{id}', [OrderedProductController::class, 'orderedProductList']);
```

3.6. Backend - Laravel

Models

app/Models, itt található fájlok tartalmazzák az adatbázis táblákhoz tartozó model-eket.

\$table: az adatbázisban tárolt tábla nevét határozza meg

\$fillable: szabályozza azokat a mezőket, amelyek módosíthatók

\$hidden: tartalmazza azokat a mezőket, amelyeket a modell lekérésekor el kell rejtetni.

Kapcsolatok: más táblákkal való kapcsolatokat lehet definiálni, pl: hasOne(), hasMany(), belongsTo

-tábla neve

```
class Order extends Model
{
    use HasFactory, HasApiTokens, Notifiable;
    protected $table = 'orders';
}
```

-változtatható mezők

```
protected $fillable = [
    'user_id',
    'postal_code',
    'city',
    'address',
    'phone_number',
];
```

-más táblákkal való kapcsolat

```
public function user()
{
    return $this->belongsTo(User::class);
}

public function orderToOrdered()
{
    return $this->hasMany(OrderedProduct::class);
}
```

Request

app/Http/Request:

Szabályokat fogalmazzuk meg a beérkező adatoknak: Store, Update beállítások, egyedi a Login és Signup (példa: UpdateOrderProductRequest):

- 'order_id' - kötelező megadni, léteznie kell az "orders" tábla "id" mezőjében
- 'product_id' - kötelező megadni, léteznie kell az "products" tábla "id" mezőjében
- 'quantity' - kötelező megadni, számnak kell lenni, min. értéke 1 lehet
- 'price' - azonos az előzővel

```
return [  
    'order_id' => 'required|exists:orders,id',  
    'product_id' => 'required|exists:products,id',  
    'quantity' => 'required|numeric|min:1',  
    'price' => 'required|numeric|min:1',  
];
```

Controllers

app/Http/Controller/Api:

Ezekben szabályozzuk, hogy a CRUD műveletek esetén mi fusson le.

index() metódus, GET kérés esetén visszaadja a kért adatot

store() metódus POST kérésre létrehoz az adatbázisban új rekordot

show() metódus az egyedi azonosítóra (ID) ad vissza egyetlen rekordot

update() metódus a PUT kérésre módosítja az ID-val azonosított rekordot

destroy() metódus DELETE kérésre eltávolítja az ID-val azonosított rekordot

Itt szűrjük a jogosultságokat is, hogy más-más adat menjen vagy éppen ne menjen ki egy dolgozónak (employee == true), és egy megrendelőnek (employee == false) amit a Controller fájlokba írt **employee()** függvénnyel ellenőrzünk

Egyes esetekben tovább kell szűrniük, pl. egy megrendelő csak a saját ID-jához tartozó rekordokat kaphassa meg:

```

public function orderedProductList($id)
{
    if ($this->employee()) {
        $finalList = OrderedProduct::where('order_id', $id)->get();
    } else {
        $loggedInUserId = Auth::id();
        $list = OrderedProduct::where('order_id', $id)->get();
        $finalList = [];
        foreach ($list as $item) {
            $order = Order::find($item->order_id);
            if ($order && $order->user->id === $loggedInUserId) {
                $finalList[] = $item;
            }
        }
    }
    return OrderedProductResource::collection($finalList);
}

```

A legbonyolultabb függvény az **OrderedProductController store()** függvénye, melynél először ellenőrizzük, hogy alkalmazott indította-e a kérést, majd kiszámoljuk az egy termékre eső árat, ezután ellenőrizzük az árat az adatbázisból vett termék árával és ha egyenlőek (ez az ellenőrzés arra szolgál, hogy ne lehessen eltérő árral rendelést leadni), akkor ellenőrizzük, hogy ha már van ilyen **“product_id”**-val ellátott rekord az **“ordered_products”** táblában akkor a beérkezett kérésből származó mennyiség és ár adatot adja hozzá a már meglévőhöz:

```

public function store(StoreOrderedProductRequest $request)
{
    if ($this->employee()) {
        $validatedData = $request->validated();

        $product = Product::findOrFail($validatedData['product_id']);
        $productPrice = $product->price;

        $orderedProductPrice = $validatedData['price'] / $validatedData['quantity'];

        if (abs($orderedProductPrice - $productPrice) > 0.001) {
            return response()->json(['error' => 'Invalid product price'], 400);
        }

        $existingOrderedProduct = OrderedProduct::where('order_id', $validatedData['order_id'])
            ->where('product_id', $validatedData['product_id'])
            ->first();

        if ($existingOrderedProduct) {
            $existingOrderedProduct->update([
                'quantity' => $existingOrderedProduct->quantity + $validatedData['quantity'],
                'price' => $existingOrderedProduct->price + $validatedData['price']
            ]);
            return response(['data' => new OrderedProductResource($existingOrderedProduct)], 201);
        }

        $ordered = OrderedProduct::create($validatedData);

        return response(['data' => new OrderedProductResource($ordered)], 201);
    }
}

```

A **ProductController**-ben találhatóak a termékek keresését kezelő függvények. A keresési találat és az összes termék lekérését végző funkciók csak azokat a sorokat kéri le melyek “active” értéke ‘1’. Előzetes feltétele a függvények használatának az ‘employee’=‘1’ jogosultság is.

```

public function index()
{
    if ($this->employee()) {
        return ProductResource::collection(Product::where('active', 1)
            ->get());
    }
}

public function employee()
{
    $user = Auth::user();
    return $user->employee;
}

public function searchByName($name)
{
    if ($this->employee()) {
        $products = Product::where('name', 'like', '%' . $name . '%')
            ->where('active', 1)
            ->get();
        return ProductResource::collection($products);
    }
}

public function searchByCategory($category)
{
    if ($this->employee()) {
        $products = Product::where('category', 'like', '%' . $category . '%')
            ->where('active', 1)
            ->get();
        return ProductResource::collection($products);
    }
}

```

A **PhotoController**-hez nem tartoznak más fájlok.

A **PhotoController** elmenti az API-n keresztül beérkezett képeket a 'public/images' mappába.

```

class PhotoController extends Controller
{
    public function upload(Request $request)
    {
        $uploadedFile = $request->file('photo');
        $filename = $uploadedFile->getClientOriginalName();
        $uploadedFile->move(public_path('images'), $filename);
    }
}

```

3.7. Frontend - React

A projektben a react a vite-tal lett létrehozva, a vite egyik leglátványosabb tulajdonsága, hogy ha változtatunk valamit a kódon akkor a weboldalon is azonnal látható a változtatás.

react/.env:

Az env mappában megadunk egy

```
VITE_API_BASE_URL=http://localhost:8000
```

változót ezen a linken keresztül fog kommunikálni a react az Api hívásakor.

Ez az érték az axios-client.js fájlba van átadva /api-val van kiegészítve. Ez a fájl a tokeneket kezeli.

react/package.json:

A package.json fájlban van beállítva hogy a vite a 3000-es porton mutassa be a weboldal tartalmát.

react/index.html:

A weboldal kiindulópontja, itt adjuk meg a betűkészlet forrását és a szükséges script fájlok forrását is. Itt adjuk meg az alapértelmezett <title>-t is.

A legfontosabb a main.jsx és div id=root.

react/src/main.jsx:

Itt importálunk több JS, CSS fájlt, többek közt a saját CSS-t is, továbbá a “router”-t is ide szúrjuk be.

react/src/index.css:

Csoportosítva vannak a beállítások, alul külön szekció a reszponzív beállításoknak.

react/src/axios-client.js:

Itt adjuk meg az .env fájlból az alapértelmezett URL-t, majd utána írjuk, hogy “/api” így API hívásnál már csak az API elérési útvonal végét kell megadni.

```
3  const axiosClient = axios.create({
4      baseURL: `${import.meta.env.VITE_API_BASE_URL}/api`
5  })
```

A továbbiakban a fájl a tokenek kezeléséről rendelkezik.

react/src/router.jsx:

Itt adjuk meg, hogy melyik útvonalon melyik komponens töltődjön be. Minden komponens a StandardView gyermeke → az <Outlet />-ben jelennek meg a gyermekek.

react/src/mainViews/StandardView.jsx:

Három fix eleme van az oldalnak: navigációs sáv, footer és a felugró üzenetek.

react/src/contexts/ContextProvider.jsx:

Célja, hogy a létrehozott három elem (user, token, notification) elérhető legyen az oldal bármely pontján és beállítófüggvényeik is, így az oldal bármely pontjáról lehet állítani ezeket az értékeket.

Alapvetések a JSX fájlokhoz melyek tartalmat jelenítenek meg:

- majdnem minden komponensnél át kell állítani a “title” értékét a tartalomnak megfelelően
- majdnem minden komponensnél kezelni kell a jogosultságot (pl. regisztráció nélkül ne lehessen látható, vagy alkalmazott és megrendelő más-más tartalmat lásson)

react/src/mainViews/Navigation.jsx:

Egyes menüelemek jogosultság vizsgálat alapján jelennek, vagy nem jelennek meg.

Az ‘Adataim’ gomb a saját profil ID-ra mutató linket tartalmaz, az ID-t az ‘useEffect’ függvényben lefutó API kérés eredményéből származtatja.

Az ‘onLogout’ függvény kijelentkezteti a felhasználót.

react/src/mainViews/Footer.jsx:

Az ‘easyQuote’ weboldalán belülre található linkeket tartalmaz, valamint a céges adatokat, elérhetőségeket.

react/src/mainViews/NotFound.jsx:

Ez a komponens jelenik meg ha olyan linket nyitunk meg amely nem szerepel a Router.jsx fájlban beállított útvonalak között.

react/src/subViews/Welcome.jsx:

Ez a komponens jelenik meg a weboldal főoldalán, mely tartalmazza a céget bemutató leírást és két linket, melyek az ‘Ajánlatkérés’ menüpontra mutatnak.

react/src/subViews/HowTo.jsx:

Rövid leírást tartalmaz mely összefoglalja, hogy miként lehet leadni új ajánlatkérést.

react/src/subViews/LoginForm.jsx:

Bejelentkezési űrlapot tartalmaz, illetve az azt beküldő API hívást tartalmazó függvényt, amely kezeli az szerverről érkező válaszokat is.

Létrehoztunk egy függvényt, általában az első felében a logikai részt írjuk, ezek a javascript kódok a visszatérési értékben vannak a html kódok. Majd a végén exportáljuk.

Az oldal meglévő 'token' esetén nem tekinthető meg.

react/src/subViews/SignUp.jsx:

Regisztrációs űrlapot tartalmaz, illetve az azt beküldő API hívást tartalmazó függvényt, amely kezeli a szerverről érkező válaszokat is. Az oldal meglévő 'token' esetén nem tekinthető meg.

Az oldal felhívja a figyelmet a jelszó követelményekre.

react/src/subViews/User.jsx:

Felhasználói profilt módosító űrlapot tartalmaz, illetve az azt beküldő API hívást tartalmazó függvényt, amely kezeli az szerverről érkező válaszokat is. Beküldés előtt ellenőrzi az űrlapra bevitt adatok formai helyességét. Az oldal hiányzó 'token' esetén nem tekinthető meg.

react/src/subViews/NewOrderForm.jsx:

Új ajánlatkérés beküldését lehetővé tevő űrlapot tartalmaz. illetve az azt beküldő API hívást tartalmazó függvényt, amely kezeli az szerverről érkező válaszokat is. Beküldés előtt csatolja a beküldött adatokhoz a felhasználó saját ID értékét is. Az oldal hiányzó 'token' esetén nem tekinthető meg.

react/src/subViews/Orders.jsx:

Alapértelmezett felhasználó esetén ('employee'='0') a saját felhasználói ID-hoz tartozó megrendeléseket jeleníti meg táblázat formájában. Alkalmazott esetén az összes megrendelést megjeleníti amelyeket az API kiküld. Az oldal hiányzó 'token' esetén nem tekinthető meg.

react/src/subViews/ProductsCards.jsx:

Alkalmazott ('employee'='1') számára elérhető, itt tudja kiválasztani azt a rendelést amelyhez hozzá kívánja adni a komponens által generált kártyákon található termékek közül a kívántakat, meghatározva a darabszámot. Háromféle termék keresési funkció érhető el (név, kategória, alkategória) A termékeket listázó kártyákon található 'Részletek' gombra kattintva felugró Modal nyílik meg, mely tartalmazza a termék részletes adatait, és egy hivatkozást amelyen a termék saját adatlapja nyílik meg új lapon. Az oldal hiányzó 'token' esetén nem tekinthető meg.

react/src/subViews/OrderedProductList.jsx:

A komponens egy megrendelést megjelenítő táblázat alsó sorában található hivatkozás megnyitására jelenik meg. Tartalmazza a megrendeléshez tartozó termékek listáját, darabszámát, összegét táblázatot formában. A termék nevét megjelenítő cellában van egy hivatkozás a termék ID értékét magába foglaló linkre, amelyen a termék részletes adatai tekinthetőek meg. Az oldal hiányzó 'token' esetén nem tekinthető meg.

react/src/subViews/ProductDetails.jsx:

A termék részletes adatlapját tartalmazza a komponens, melyre a termék ID-ját véve alapul lehet hivatkozni. A hivatkozások elérhetőek a 'ProductsCards' komponensben található Modal-on, illetve az 'OrderedProductsList' komponensben megjelenő táblázat terméknevet tartalmazó

cella sorában. A termék ára kizárólag alkalmazotti ('employee'='1') jogosultság mellett jelenik meg. Az oldal hiányzó 'token' esetén nem tekinthető meg.

3.8. Asztali alkalmazás - C#

Projekt elemei fejlesztés szempontjából C#

Osztályok:

A projekt értelmezhetőségének szempontjából először a segédosztályokkal kezdjük az osztályok bemutatását. Minden osztály elején, még az osztályon kívül, a clean code elveknek megfelelően az átláthatóság és az értelmezhetőség miatt kommentbe szerepelnek az osztályban definiált változók és függvények. Az alkalmazás későbbi továbbfejlesztésének érdekében az osztályban szereplő változók és függvények előtt angolul kommentek találhatók.

Segédosztályok:

- **DataStorage.cs:** A program több pontján használt stringeket és string tömböket tárolja
- **InputTest.cs:** Ez a segédosztály segít függvényeivel ellenőrizni a beviteli mezők helyességét, kezeli a hibákat
- **LinkOpener.cs:** Lehetővé teszi URL hivatkozások megnyitását az alapértelmezett böngészőben.
- **Md5has.cs:** Ez a segédosztály segít a jelszavak hash-elésében, biztonságos tárolásában.

A **DataStorage** segédosztály egy publikus, statikus osztály, tehát nem példányosítható.

- **databaseConnectionString:** adatbázis kapcsolatot meghatározó string. A program összes adatbáziskapcsolata az itt tárolt karakterláncot használja, segítségével egyszerűen és központilag kezelhető az adatbázis kapcsolat a program különböző részeinél.
- **GetActiveItems():** A program ComboBox-ainak feltöltését szolgáló 'Nem' és 'Igen' stringeket tartalmazó tömb.
- **GetCategoryItems():** Olyan függvény, mely termékkategória string tömböt tartalmaz a legördülőlista feltöltésére. Visszatérési értéke a teljes tömb.
- **GetSubCategoryItems():** Termék-alkategória stringeket tartalmazó tömb, ezzel a függvénnyel könnyedén feltölthető a ComboBox a különböző termék-alkategóriákkal. Visszatérési értéke a teljes tömb.

Az **InputTest** egy publikus, statikus osztály, melynek az összes függvény visszatérési értéke **'true'**, vagy **'false'**. Ennek az osztálynak a célja az alkalmazásban a beviteli mezők ellenőrzése, és **'false'** érték esetén a megadott hibaüzenetet adni.

- **InputStringTest(params string[] inputs):** Bármennyi stringet fogadni képes függvény, mely ellenőrzi, hogy a beküldött stringek üresek, vagy sem. Amennyiben üres, akkor **'false'** értékkel tér vissza.
 - **InputPasswordMinimumLongTest(string password, int minLong):** Ez a függvény az átadott password string paraméter hosszát veti össze az átadott minLong egész számmal. Ha a string rövidebb, akkor **'false'** értékkel tér vissza.
 - **InputPasswordEquals(string password1, string password2):** Ez a függvény ellenőrzi, hogy a két string azonos-e. Ha nem azonos, akkor **'false'** a visszatérési értéke.
 - **InputEmailsValid(string email):** a stringet a MailAddress beépített osztály segítségével ellenőrzi, hogy megfelel-e az általános email cím szabályainak.
1. ábra
- **InputIdForDelete(string id):** a stringet ellenőzi, hogy üres-e, ha üres, akkor **false**.

A **LinkOpener** szintén publikus, statikus osztály. A program weboldal megnyitására használja az osztály függvényét. Jelenleg csak egy függvényt tartalmaz.

- **public static void OpenNavLink():** A NAV online számla weboldalát nyitja meg az alapértelmezett böngészőben. Kezelve a link megnyitása során felléphető hibákat. Mivel ez egy void 'visszatérési értékű' függvény, így nem ad vissza semmit csak végrehajtja a műveletet.

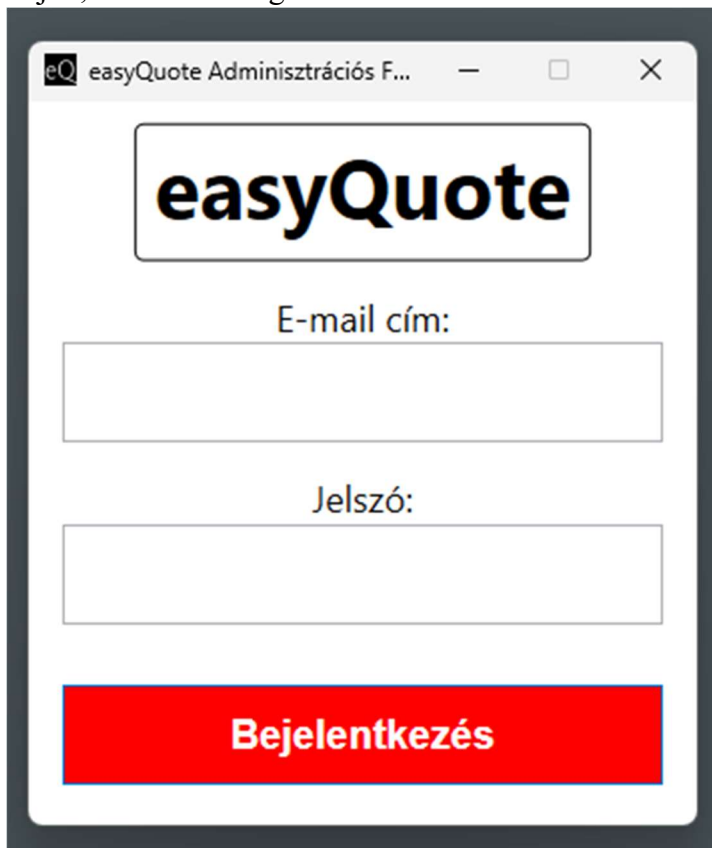
A **HashStringMd5**: publikus, statikus osztály, amelyet a program stringek hash-elésére használ.

- **Md5hash(string text):** a stringet hash-eli MD5 hash függvény által, visszatérési értéke az hash-elt string. Így biztosítja a felhasználók jelszavának biztonságos tárolását.

Ablakok:

MainWindow.xaml és MainWindow.xaml.cs

A **.xaml** fájlban 4 sorra lett tagolva a felület, fix magassággal és szélességgel, a képernyő közepén jelenik meg az ablak. Felül a logót, második sorban az email cím- (TextBox), alatta a jelszó beviteli mezőt (PasswordBox) és végül az utolsó sorban a 'Bejelentkezés' feliratú gombot tartalmazza, mely **Click** eseményéhez van hozzárendelve a '**LoginButton**' függvényben. Az ablak formázásának elemeit az **.xaml** fájlban adjuk meg, ami hasonlít a **css**-re. A clean code elveket követve a beállítási kódokat úgy adtuk meg, hogy első sorban a fontossági sorrendben legyenek, másod sorban a sorok hossza nőjön, az átláthatóság érdekében.



A **MainWindow.xaml.cs** fájl a **LoginButton** függvényt és a **bossOrNot** változót tartalmazza, mely *bool* értéket vehet fel, amit bejelentkezés után kap meg attól függően, hogy a bejelentkezéshez használt jelszó kezdete 'boss@' volt-e vagy sem. A változó azért publikus elérésű, mert a program ennek a változónak az értékét vizsgálja jogosultságkezelés esetén.

A **LoginButton** függvény az adatbázis felé küld egy lekérést, mely lekéri az a sort melynek email mező értéke azonos a beírt email címmel. A sorból a 'password' mező értékét összeveti a beírt jelszó hash-elte értékével. A hash-eléshez az **Md5hash** osztályt használja. Sikeres egyezés esetén megnyitja az **easyquoteDashboard** ablakot. Hiba esetén a megfelelő hibaüzenet jelenik meg.

easyquoteDashboard.xaml/easyquoteDashboard.xaml.cs

Az ablak két sort tartalmaz, felül a menüsor található 25 pixel magassággal, alatt a maradék teret a **MainFrame** nevű Frame tölti ki, ahová a lapok (page) töltődnek be. Az

ablak első betöltésekor automatikusan az '**orders.xaml**' lap tölt be teljes képernyős módban.

Menügombok és klikk események:

- Rendelések - OpenOrdersPage
- NAV Számlázás - OpenNAVLink
- Termékek - OpenProductsPage
- Felhasználók - OpenUsersPage
- Kilépés - CloseAppButton
- Névjegy – OpenAboutWindow

A Rendelések, Termékek, Felhasználók gombok a következő lapokat töltik be a MainFrame-be azonos sorrendben: **orders.xaml**, **products.xaml**, **users.xaml**.

A Névjegy gomb az **AboutWindow** ablakot nyitja meg. A Kilépés gomb bezárja az ablakot.

A NAV Számlázás gomb a **LinkOpener osztály** segítségével nyitja meg a NAV weboldalát.

Orders.xaml/Orders.xaml.cs

Alapértelmezetten bejelentkezés után ezt a lapot tölti be. Két fő oszlopra van felosztva a lap. Mindkettő további sorokra és oszlopokra tagolódik. A bal oldali oszlopban található a keresőmező, a megrendeléseket megjelenítő táblázat és alatta a kiválasztott megrendeléshez tartozó termékeket megjelenítő táblázat melyeket a fenti táblázatban látható megrendeléseket tartalmazó sorok egyikére kattintással való interakció során tölti be az alsó táblázatba. A termékek egységárát összeszorozza a megrendelt mennyiséggel és a '**Összesen**' oszlopban található mezőbe írja be a végső értéket, majd az **Összesen** oszlopban található értékeket összeadja a program és a jobb oldalt található űrlapot tartalmazó oszlop **Száml. össz.** címkétől jobbra található TextBlock-ba írja ki.

The screenshot shows a web application window titled 'anyQuote Adminisztrációs Felület'. It features a search bar at the top left. Below it is a table with columns: ID, UszID, IRSZ, Város, Cím, Tel. szám, Fiz. határidő, Elfogadva, Felmérés, Teljesítve, Létrehozva, and Módosítva. The table contains three rows of order data. Below the table is a second table with columns: ID, Rend. ID, Termék ID, Termék Név, Termék Kat, Mennyiség, Egységár, and Összesen. To the right of the main table is a sidebar with various input fields and buttons. The sidebar includes fields for Rend. ID, IRSZ, Város, Cím, Tel. szám, and dropdown menus for Elfogadva, Felmérés, and Teljesítve. At the bottom of the sidebar, it shows Száml. össz.: 0 Ft and buttons for Frissítés, Kiűrtés, Törölés, and Számlázás.

ID	UszID	IRSZ	Város	Cím	Tel. szám	Fiz. határidő	Elfogadva	Felmérés	Teljesítve	Létrehozva	Módosítva
1	6	1000	Százholdas Pagony	Faház út	06201234567		False	False	False	1/5/2023 12:00:00 AM	1/5/2023 12:00:00 AM
2	7	1500	Százholdas Pagony	Faodu utca	06301234567		False	False	False	1/6/2023 12:00:00 AM	1/6/2023 12:00:00 AM
3	8	2000	Százholdas Pagony	Sátoralja köz	06701234567		False	False	False	1/7/2023 12:00:00 AM	1/7/2023 12:00:00 AM

ID	Rend. ID	Termék ID	Termék Név	Termék Kat	Mennyiség	Egységár	Összesen
----	----------	-----------	------------	------------	-----------	----------	----------

Az **Orders.xaml.cs** fájlban található függvények:

- **LoadGrid():** A felső táblázatba tölti be az 'orders' táblából SQL lekérés által megkapott adatokat.
- **LoadComboboxes():** Az űrlapot tartalmazó oszlopban található ComboBoxokba tölti be az 'ID' értékeket és az 'Igen', 'Nem' értékeket.
- **ClearBoxes():** Kiüríti az űrlap mezőit.
- **ClearButton(object sender, RoutedEventArgs e):** A Kiürítés gomb megnyomásakor meghívja a Clearboxes, LoadGrid, LoadComboboxes függvényeket.
- **DeleteButton(object sender, RoutedEventArgs e):** A 'Törlés' gombra kattintáskor ellenőrizni az 'ID' mező értékét, felhasználói megerősítés után meghívja a DeleteExecute függvényt.
- **DeleteExecute(string orderId):** Az SQL utasítás eltávolítja azokat az rendeléseket az 'orders' táblából, amelyeknek azonosítója megegyezik a felhasználó által kiválasztott ID-vel.
- **UpdateButton(object sender, RoutedEventArgs e):** Az űrlapból beolvasott adatokat átfuttatja az InputStringTest függvényen a kitöltöttség ellenőrzés okán, majd felhasználói megerősítés után az 'Igen' és 'Nem' értékeket 1-re és 0-ra konvertálja, majd átadja az adatokat az UpdateExecute függvénynek.
- **UpdateExecute():** A paraméterként megkapott adatokkal update típusú SQL lekérést hajt végre, hozzáadva az 'updated_at' értékhez az aktuális dátumot és időt, illetve ha a 'completed' értéke '1', akkor hozzátoldja az SQL utasításhoz az 'payment_deadline' beállítást is, melyhez hozzárendeli az aktuális dátum + 15 nap értéket.
- **ShowConfirmationDialog(string message string title):** Beleegyezést kérő ablakot meghatározó függvény.
- **SearchTextChanged(object sender, TextChangedEventArgs e):** A keresési mezőbe beírt minden egyes karakterváltozás hatására fut le és az értéket átadja a SearchResult függvénynek.
- **SearchResult(string searchKeyword):** Az 'orders' tábla 'postal_code', 'city', 'address', 'phone_number', 'created_at' mezőiben keresi az egyezést a függvényhíváskor megkapott értékkel és betölti a felső táblázatba azokat a sorokat ahol találat van.
- **DataGridDoubleClick(object sender, MouseButtonEventArgs e):** Függvény, amely a felső táblázatban található sorra történő dupla kattintás esetén végzi el a műveleteket: a sorban található értékeket az űrlap megfelelő mezőibe helyezi, és szükség esetén átalakítja a '0' és '1' értékeket 'Nem' és 'Igen' értékekké.
- **DataGridSelectionChanged(object sender, SelectionChangedEventArgs e):** Felső táblázat sorára kattintáskor az alsó táblázatba tölti be a megrendeléshez tartozó termékeket sorhoz tartozó ID érték alapján SQL lekérdezés által az 'ordered_products' táblából. Továbbá kiszámolja az összeg értékeket.
- **OpenNAVLink(object sender, RoutedEventArgs e):** Megnyitja az alapértelmezett böngészőben a NAV online számla weboldalát.

Products.xaml/Products.xaml.cs

Két fő oszlopra van felosztva a lap. Mindkettő további sorokra és oszlopokra van felosztva. A bal oldali oszlopban található a keresőmező, a termékeket megjelenítő

táblázat. A jobb oldali oszlopban található a termékek szerkeszthetőségét lehetővé tevő űrlap. A **Frissítés** gomb elmenti az űrlapon található információkat a kijelölt ID értékkel megegyező ID értékű termékhez. A **Kiürítés** gomb kiüríti az űrlap mező értékeit. A **Képfeltöltés** gomb a kijelölt ID-jű termékhez tölt fel új képet és módosítja a termék **img_url** értékét. Az **Új termék** gomb megnyitja az új termék hozzáadása ablakot.

ID	Név	Kategória	Alkategória	Kép URL	Ár Ft	Aktív	Leírás
1	Kezelő - TM50	riasztó	kezelő	partm50w.jpg	83983	True	Multifunkciós érintőképernyő
2	Kezelő - TM70	riasztó	kezelő	tm70w.jpg	107206	True	Paradox TM70 multifunkciós
3	Kezelő - K656 W	riasztó	kezelő	k656w.jpg	62212	True	Érintógombos LCD kezelő LED
4	Kezelő - K32LCD+	riasztó	kezelő	k32lcd.jpg	44432	True	Kétpartíciós 32-zónás LCD ke
5	Kezelő - K641+	riasztó	kezelő	k641cd.jpg	56406	True	Egy vagy több partícióhoz re
6	Kezelő - K32+	riasztó	kezelő	k32.jpg	34272	True	Kétpartíciós 32-zónás LED ke
7	Kezelő - K10 V	riasztó	kezelő	k10v.jpg	22842	True	Kétpartíciós 10-zónás LED ke
8	Kezelő - K10 H	riasztó	kezelő	k10h.jpg	22842	True	Kétpartíciós 10-zónás LED ke
9	Központ - SP4000	riasztó	kozpont	sp4000.jpg	30099	True	2 partíciós 4-32 zónás 32 felh
10	Központ - SP6000	riasztó	kozpont	sp6000.jpg	48786	True	Paradox Spectra SP6000+ köz
11	Központ - EVO192	riasztó	kozpont	evo192.jpg	53140	True	8 partíciós, 8-192 zónás, 999
12	Központ - MG5050+	riasztó	kozpont	mg5050.jpg	62575	True	Paradox Magellan MG5050+ köz
13	Sziréna - VL-640L	riasztó	szirena	szirvl-640.jpg	14984	True	Külséri akkumulátoros sziréna
14	Sziréna - SOLO	riasztó	szirena	szisolo.jpg	15213	True	Külséri akkumulátoros sziréna
15	Sziréna - PS-128-1	riasztó	szirena	szisps128.jpg	23889	True	Külséri akkumulátoros hang-f
16	Mozgásérzékelő - Optex - FLX-S-ST	riasztó	mozgaserzekelo	optex.jpg	7800	True	Optex FLX-S-ST hagyományo
17	Mozgásérzékelő - PMD75	riasztó	mozgaserzekelo	dgt75.jpg	14095	True	Digitális, kétptitkás, csúcsmín
18	Mozgásérzékelő - PMD2P	riasztó	mozgaserzekelo	pmd2.jpg	22842	True	Rádiós digitális, egyoptitkás m
19	Mozgásérzékelő - 476+	riasztó	mozgaserzekelo	pro.jpg	6384	True	Analog mozgásérzékelő relés
20	TELL GSM Compact GSM II	riasztó	riasztokiegészito	gsmcompactgsm-ii.jpg	52007	True	2 bemenet / 2 relé kimenetes
21	Akkumulátor - Seal Lead Acid 4,2 Ah	riasztó	riasztokiegészito	ak42power.jpg	5842	True	S. L. A. szünetmentes riasztó
22	Akkumulátor - Seal Lead Acid 6,5 Ah	riasztó	riasztokiegészito	ak65power.jpg	7064	True	S. L. A. szünetmentes riasztó
23	Transzformátor - 16V AC 30VA transzformátor	riasztó	riasztokiegészito	trafo30.jpg	4166	True	Műgyantával kiöntött transzf
24	Transzformátor - 16V AC 45VA transzformátor	riasztó	riasztokiegészito	trafo45.jpg	4724	True	Műgyantával kiöntött transzf
25	Vezeték	riasztó	riasztokiegészito	utp305belden.jpg	230	True	Bitner CAT5e beltéri, fali UTP
28	Kamera - Techson TCA EB5 E102 IH50 VF 2 Mpx-es Analog HD kamera	kamera	kamera	E1-01.jpg	35905	True	Techson Analog HD kamera, 5
30	Kamera - Techson TCA EB2 E105 IH50 Z4 5 Mpx-es Analog HD kamera	kamera	kamera	tcae2e105ih50z4.jpg	51326	True	Techson Analog HD kamera, 5
31	Kamera - Hikvision DS-2CD2163G2-I(4mm) 6 Mpx-es IP kamera	kamera	kamera	DS-2CD2163G2-I(4mm).PNG	109263	True	Hikvision 6 megapixeles felb

A **Products.xaml.cs** fájlban található függvények:

- **LoadGrid():** A táblázatba tölti be a 'products' táblából SQL lekérés által megkapott adatokat.
- **LoadComboboxes():** Az űrlapot tartalmazó oszlopban található ComboBoxokba tölti be az 'ID' értékeket, az 'Igen', 'Nem' értékeket, a termékek kategóriákat és alkategóriákat.
- **ClearBoxes():** Ez a függvény végzi az űrlap mezőinek kiürítését.
- **ClearButton(object sender, RoutedEventArgs e):** A Kiürítés gomb megnyomására meghívja a Clearboxes, LoadGrid, LoadComboxes függvényeket.
- **UpdateButton(object sender, RoutedEventArgs e):** Az űrlapról beolvasott adatokat átfuttatja az InputStringTest függvényen a kitöltöttség ellenőrzés okán, majd felhasználói megerősítés után az 'Igen' és 'Nem' értékeket 1-re és 0-ra konvertálja, majd átadja az adatokat az UpdateExecute függvénynek.
- **UpdateExecute():** A megkapott adatokkal update típusú SQL lekérést hajt végre.
- **SearchTextChanged(object sender, TextChangedEventArgs e):** A keresési mezőbe beírt minden egyes karakterváltozás hatására fut le és az értéket átadja a SearchResult függvénynek.
- **SearchResult(string searchKeyword):** A products tábla 'name', 'category', 'sub_category', 'description' mezőiben keresi az egyezést a függvény híváskor megkapott értékkel és betölti a táblázatba azokat a sorokat ahol találhat van.
- **DataGridDoubleClick(object sender, MouseButtonEventArgs e):** A felső táblázat sorára duplakattintás esetén meghívódó függvény, a sorban található értékeket berakja az űrlap megfelelő mezőibe, ahol szükséges konvertálja a '0' és '1' értékeket 'Nem' és 'Igen' értékekre.

- **ImageUploadButton(object sender, RoutedEventArgs e):** Ha ki van választva egy termék ID, akkor megnyitja a képkiterjesztésű fájlokra korlátozott képfeltöltési lehetőséget nyújtó ablakot a fájlkezelővel. A kép kiválasztása után a képfájl neve után illeszti az aktuális dátumot és időpontot a fájl ütközések elkerülése végett, majd a fotót elküldi az API számára. Az fájl új nevét és az ID-t átadja a ChangeImageUrl függvénynek.
- **ChangeImageUrl(string id string fileName):** A kapott ID-val megegyező sort megkeresi a **products** táblában és kicseréli a kapott fájl nevével az 'img_url' mező értékét.
- **AddNewProductButton(object sender, RoutedEventArgs e):** Megnyitja az **AddNewProduct** ablakot.

Users.xaml/ Users.xaml.cs

Két fő oszlopra van felosztva a lap. Mindkettő további sorokra és oszlopokra van felosztva. A bal oldali oszlopban található a keresőmező és a felhasználókat megjelenítő táblázat. A jobb oldali oszlopban található a felhasználók szerkeszthetőségét lehetővé tevő űrlap. A Frissítés gomb elmenti az űrlapon található információkat a kijelölt ID értékkel megegyező ID értékű felhasználóhoz. A Kiürítés gomb kiüríti az űrlap mező értékeit. A Törlés gomb törli az ID által kijelölt felhasználót. Az Új felhasználó gomb megnyitja az új irodai felhasználó hozzáadása ablakot.

ID	Név	E-mail	Dolgozó	Létrehozva	Módosítva
1	boss	boss@easyquote.hu	True	1/1/2023 12:00:00 AM	1/1/2023 12:00:00 AM
2	iroda1	iroda1@easyquote.hu	True	1/1/2023 12:00:00 AM	1/1/2023 12:00:00 AM
3	iroda2	iroda2@easyquote.hu	True	1/2/2023 12:00:00 AM	1/2/2023 12:00:00 AM
4	iroda3	iroda3@easyquote.hu	True	1/3/2023 12:00:00 AM	1/3/2023 12:00:00 AM
5	felmero	felmero@easyquote.hu	True	1/4/2023 12:00:00 AM	1/4/2023 12:00:00 AM
6	Micimackó	minimacko@thepooh.hu	False	1/5/2023 12:00:00 AM	1/5/2023 12:00:00 AM
7	Malacka	malacka@theshy.hu	False	1/6/2023 12:00:00 AM	1/6/2023 12:00:00 AM
8	Füles	fules@thesad.hu	False	1/7/2023 12:00:00 AM	1/7/2023 12:00:00 AM

Keresés:

Felhaszn. ID: 4

Név:

E-mail:

Dolgozó:

Új jelszó:

Jelszó ism.:

Frissítés

Kiürítés

Törlés

Új felhasználó

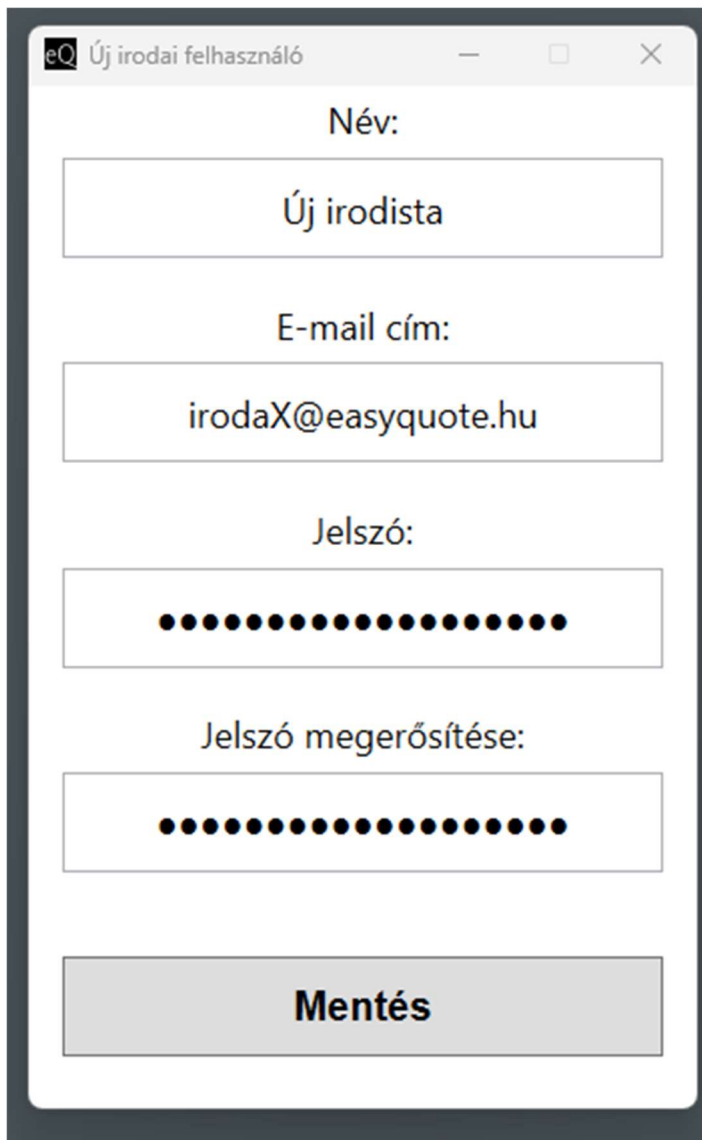
A **Users.xaml.cs** fájlban található függvények:

- **BossOrNot():** Ellenőrzi a **MainWindow.xaml.cs** fájlban található bossOrNot bool értékét, ha az érték hamis, akkor a releváns mezőkre nem lehet kattintani és módosítani, illetve ezeket a mezőket és gombokat elrejti.
- **LoadGrid():** SQL lekérés által lekéri a **users** táblából az adatokat és betölti a táblázatba.
- **LoadComboboxes():** Az űrlapon található comboBox-ba betölti az 'ID' értékéhez tartozó 'Igen', 'Nem' értékeket.
- **ClearBoxes():** Törli az űrlap mezőit és meghívja a **BossOrNot** függvényt.

- **ClearButton(object sender, RoutedEventArgs e):** Kiürítés gomb megnyomására meghívja a Clearboxes, LoadGrid, LoadComboxes függvényeket.
- **DeleteButton(object sender, RoutedEventArgs e):** Törlés gomb megnyomására ellenőrzi az 'ID' értékét, majd felhasználói megerősítés után meghívja a **DeletExecute** függvényt.
- **DeleteExecute(string userId):** SQL utasítás által törli a felhasználó által kiválasztott ID-val rendelkező felhasználót a **users** táblából.
- **ShowConformationDialog(string message, string title):** belleegyzést kérő ablak függvénye.
- **UpdateButton(object sender, RoutedEventArgs e):** Az űrlapról beolvasott adatokat átfuttatja az **InputStringTest** függvényen, ha a jelszómező ki van töltve akkor azokat is átfuttatja az **InputTest** osztály jelszóellenőrző függvényein, majd felhasználói megerősítés után az 'Igen' és 'Nem' értékeket 1-re és 0-ra konvertálja, majd átadja az adatokat az **UpdateExecute** függvénynek.
- **UpdateExecute():** A megkapott adatokkal update típusú SQL parancsot hajt végre hozzáadva az "updated_at" értékhez az aktuális dátumot és időt, illetve ha a jelszó ki van töltve, akkor hozzátoldja az SQL utasításhoz a jelszó értékét miután átfuttatta az **Md5hash** függvényen.
- **SearchTextChanged(object sender, TextChangedEventArgs e):** A keresési mezőbe beírt minden egyes karakterváltozás hatására fut le és értéket átadja a **SearchResult** függvénynek.
- **SearchResult(string searchKeyword):** A **users** tábla 'name', 'email' mezőiben keresi az egyezést a függvényhíváskor megkapott értékkel és betölti a táblázatba azokat a sorokat ahol találat van.
- **DataGridDoubleClick(object sender, MouseButtonEventArgs e):** A felső táblázat sorára duplakattintás esetén meghívódó függvény, mely a sorban található értékeket berakja az űrlap megfelelő mezőibe, ahol szükséges konvertálja a '0' és az '1' értékeket 'Nem' és 'Igen' értékekre. Továbbá ellenőrizni a bossOrNot bool értékét és ha igaz, akkor további szabályozást alkalmaz, hogy a 'boss@' kezdetű felhasználó a különböző email cím előtagokkal rendelkező felhasználóknál milyen adatokat állíthasson át.
- **AddNewUserButton(object sender, RoutedEventArgs e):** Megnyitja a **NewUser** ablakot.

NewUser.xaml/NewUser.xaml.cs

Az ablak 4 sorra van felosztva mérete fix 550*350 pixel, a képernyő közepén nyílik meg. Felül a Név, alatt az E-mail cím, majd Jelszó, Jelszó megerősítése és végül a Mentés gomb.

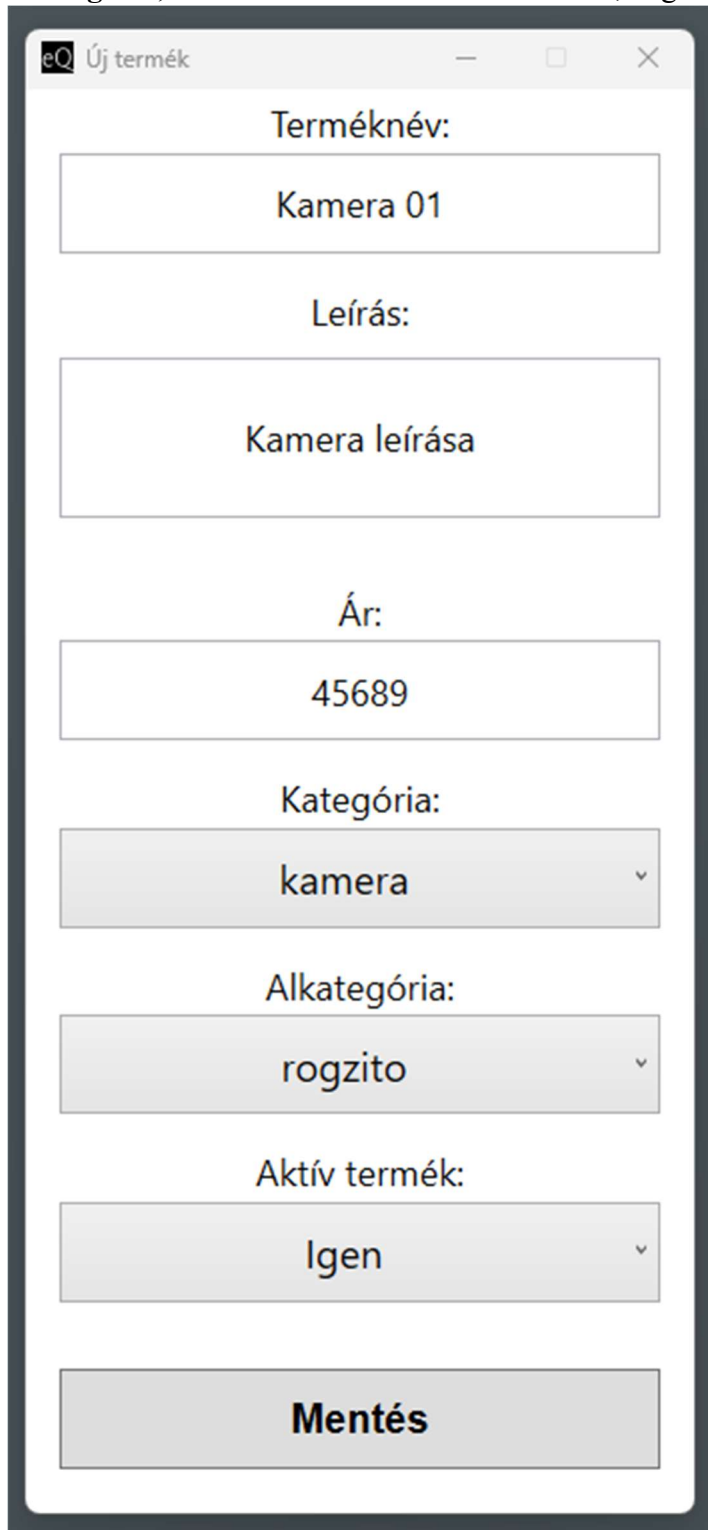


A `NewUser.xaml.cs` fájlban található függvények:

- **SaveNewUserButton(object sender, RoutedEventArgs e):** A Mentés gomb megnyomására fut le a beviteli mezőből kiolvassa az adatokat, majd az **InputTest** függvényen átfutattja azokat. Ha az ellenőrzésen minden adat hiba nélkül átment, akkor egy insert into SQL utasítással létrehozza az új felhasználó bejegyzését, melyhez a password mezőbe írt értéket az **Md5hash** függvénnyel hash-eli, majd az SQL utasításhoz hozzáadja az aktuális dátumot és időpontot az 'updated_at' és 'created_at' mezőkbe.
- **InputTesting():** A regisztrációs űrlapon megadott adatokat veti alá ellenőrzésnek. Az elágazások különböző típusú ellenőrzéseket hajtanak végre, melyek szinte mindegyike az **InputTest** osztály függvényeit használja, kivéve egyet, mely azt ellenőrizni, hogy az email az 'iroda' előtaggal kezdődik vagy sem, mert a regisztráció kizárólag 'iroda' előtaggal rendelkező email címekkel történhet meg, mivel ez a regisztráció csakis az irodai dolgozók részére készült C#-os programhoz ad belépési jogosultságot.

AddNewProduct.xaml/AddNewProduct.xaml.cs

Az ablak 7 sorra van felosztva mérete fix 750*350 pixel, a képernyő közepén nyílik meg. Fentről lefelé haladva: a **Terméknév**, a **Leírás**, és az **Ár** TextBoxok, míg a **Kategória**, az **Alkategória**, és az **Aktív termék** ComboBoxok, végül a **Mentés** gomb.



The screenshot shows a mobile application window titled "Új termék" (New product). The window contains the following elements from top to bottom:

- Terméknév:** A text box containing "Kamera 01".
- Leírás:** A text box containing "Kamera leírása".
- Ár:** A text box containing "45689".
- Kategória:** A dropdown menu with "kamera" selected.
- Alkategória:** A dropdown menu with "rogzito" selected.
- Aktív termék:** A dropdown menu with "Igen" selected.
- Mentés** button at the bottom.

Az **AddNewProduct.xaml.cs** fájlban található függvények:

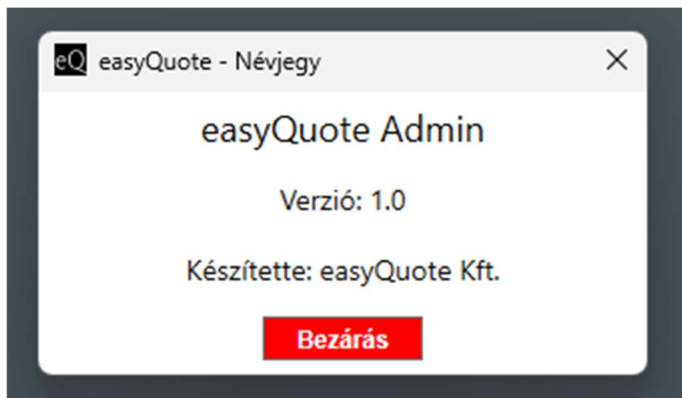
- **LoadComboboxes():** Betölti a ComboBoxokba a **DataStorage** osztályban található függvények által az 'Igen' és 'Nem' értékeket, illetve a termék

kategóriákat és alkategóriákat. Létrehoztunk egy `selectedFilePath` nevű változót amely képfeltöltés esetén a fájl új nevét fogja tárolni.

- **SaveNewProductButton(object sender, RoutedEventArgs e):** A Mentés gomb megnyomására az űrlapról kiolvasott értékeket átfuttatja az `InputTest` osztály stringeket ellenőrző függvényén az értékeket, majd átalakítja a **active** értéket **0**-ra, vagy **1**-re. Ezután felugrik a termékfotót kiválasztását lehetővé tevő fájlkezelő ablak amelyben lekorlátozzuk a kiválasztható fájlok típusát kép típusú fájlokra. Ha nem választunk ki képet akkor rögtön átadja az árlap értékeit **SaveNewProduct** függvénynek. Ha választunk ki képet akkor átadja az **ImageUpload** függvénynek, mely a fájl neve után hozzátoldja az aktuális dátumot és időpontot, majd feltölti a fájlt az API-ra és a **selectedFilePath** változónak a fájl új nevét adja mint érték. Ezután ezt az új értékkel átadott változóval hívjuk meg a **selectedFilePath** függvényt.

AboutWindow.xaml/AboutWindow.xaml.cs

Fix 180*320 pixel méretű 4 sorból álló ablak. Fent a program neve, alatta az aktuális verziószáma, harmadik sorban a készítő cég neve, végül az utolsó sorban a Bezárás gomb található.



Az **AboutWindow.xaml.cs** fájlban a **CloseAboutButton(object sender, RoutedEventArgs e)** függvény található, mely a Bezárás gomb megnyomásával hívódik meg és bezárja az ablakot.

4. Teszt dokumentáció

A tesztelést kézi teszteléssel valósítottuk meg, ahol megtapasztaltuk, hogy a funkciók helyesen működnek-e.

Az API-k működését Postman alkalmazással is teszteltük.

A kézi tesztelés eredményei:

Megrendelői regisztráció:

1. Helytelen jelszó megadása

Felhasználónév: megrendelo1

E-mail cím: megrendelo1@gmail.com

Jelszó: megrendelo1

Jelszó megerősítése: megrendelo1

Megjegyzés: jelszó: min. 8 karakter, min. egy speciális karakter

Funkció működik: igen, A mezők kitöltése helytelen! kiírást kaptunk mivel nem volt speciális karakter a jelszóban

2. Megfelelő adatok megadása

Felhasználónév: megrendelo1

E-mail cím: megrendelo1@gmail.com

Jelszó: megrendelo1@

Jelszó megerősítése: megrendelo1@

Megjegyzés: jelszó: min. 8 karakter, min. egy speciális karakter

Funkció működik: igen, az adatbázisba bekerültek a megrendelo1 adatai

Új ajánlat igénylése - megrendelő

1. Kitöltési mezők kihagyásával

Irányítószám: 1032

Város: Budapest

Cím:

Telefonszám: 06201234567

Megjegyzés: A csillaggal jelölt mezők kitöltése kötelező!

Funkció működik: igen, A mezők kitöltése helytelen! kiírást kaptunk mivel a cím nem lett kitöltve

2. Hibás telefonszám formátum

Irányítószám: 1032

Város: Budapest

Cím: Riasztó utca 5.

Telefonszám: +36201234567

Megjegyzés: kötelező telefonszám formátum: 06201234567

Funkció működik: igen, A mezők kitöltése helytelen! kiírást kaptunk mivel hibás a telefonszám formátum

3. Megfelelő adatok megadásával

Irányítószám: 1032

Város: Budapest

Cím: Riasztó utca 5.

Telefonszám: 06201234567

Megjegyzés: A csillaggal jelölt mezők kitöltése kötelező!

Funkció működik: igen, az adatbázisba bekerültek a megrendelés adatai és a rendelések menüpont alatt is feljöttek a kitöltött adatok.

Kolléga regisztráció

Felhasználónév: kollegal

E-mail cím: kollegal@gmail.com

Jelszó: kollegal@

Jelszó megerősítése: kollegal@

Megjegyzés: jelszó: min. 8 karakter, min. egy speciális karakter

Funkció működik: igen, az adatbázisba bekerültek a kollegal adatai és a C#-ban kolléga megjelölést követően megjelenik a termékek menüpont.

Keresés funkció

Rendelés kiválasztása (legördülőből): megrendelo1

Keresés kategória szerint: riasztó

Funkció működik: igen, csak a riasztós alkatrészek láthatóak

Keresés alkategória szerint: kezelő

Funkció működik: igen, csak a kezelők láthatóak

Keresés név szerint: TM

Funkció működik: igen, csak a TM50 és a TM70 eszközök láthatóak

Rendelés összeállítása

Kiválasztott termékek:

Kezelő - TM50 2 db

Központ - EVO192 1 db

Sziréna - VL-640L 1 db

Funkció működik: igen, a kiválasztott termékek hozzáadását követően a termék sikeresen hozzáadva felugró ablak megjelent és a rendelések menüpontban a megrendelt termékeknel elkészült az ár kalkuláció. Továbbá az adatok bekerültek az adatbázisba az ordered_products táblába.

5. Továbbfejlesztési lehetőségek

- a web- és az asztali alkalmazásnál email-es értesítések küldésére szükség van a megfelelő tájékoztatásra
- az asztali alkalmazásnál későbbiekben egy számlázórendszer integrálása teheti teljessé a folyamatot
- az árak és termékek frissítése, felvétele, megrendelése beszállítói API-n keresztül
- szükség esetén a webalkalmazásban benne rejlik a webáruházzá való fejlesztés lehetősége
- mivel a megrendelőnk kifejezett kérése volt bizonyos funkciók beleégetése az alkalmazásba, mint pl. a 'boss@' e-mailcímmel való belépés használata/felismerése ezt a funkciót a későbbiekben inkább az adatbázis módosításával a users táblában egy új mező beiktatásával pl. 'role', mint szerep lenne szabályozható, melyik felhasználó milyen jogosultságokkal rendelkezzen
- jelenleg az alkalmazás csak fix árakkal tud dolgozni, szükséges az egyedi árak kezelése is, ilyen például a rugalmas munkadíj, a telepítéshez szükséges anyagok, kiszállási díj kilométer alapon

6. Összegzés

A projektfeladat egy valós probléma megoldását valósította meg. A vállalkozások részére komoly kihívást jelent az árajánlat elkészítése. Minél egyszerűbben és gyorsabban készül el egy árajánlat annál hatékonyabban tud működni a cég. Célunk az volt, hogy a megrendelő regisztrációját követően a helyszínre érkezve a munka felmérését végző kolléga a helyszínen néhány perc alatt el tudja készíteni az árajánlatot. A webshopszerű alkalmazással ez megvalósítható, mivel a munka felmérését követően az eszközök kosárba helyezésével el is készül az ár kalkuláció.

A projekt megismerése érdekében jelen dokumentációban bemutatjuk a web- és az asztali alkalmazás működését, a telepítési információkat, az adatbázis, API, Backend, Frontend, C# és teszt dokumentációkat.

Összességében úgy érezzük, hogy a megszerzett tudást nagyon hasznosan fel tudtuk használni egy valós probléma megoldására.

7. Források

Aurora kiadó: Szabványkövető statikus honlapok szerkesztése / HTML5+CSS3+SVG2

<https://axios-http.com/docs/intro>

<https://learn.pvga.hu/category/laravel/>

<https://laravel.com/>

https://hu.wikipedia.org/wiki/Google_Chrome

https://hu.wikipedia.org/wiki/Mozilla_Firefox

https://en.wikipedia.org/wiki/Microsoft_Edge

[https://hu.wikipedia.org/wiki/Opera_\(webb%C3%B6ng%C3%A9sz%C5%91\)](https://hu.wikipedia.org/wiki/Opera_(webb%C3%B6ng%C3%A9sz%C5%91))

[https://hu.wikipedia.org/wiki/Safari_\(webb%C3%B6ng%C3%A9sz%C5%91\)](https://hu.wikipedia.org/wiki/Safari_(webb%C3%B6ng%C3%A9sz%C5%91))

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

https://hu.wikipedia.org/wiki/Microsoft_Visual_Studio

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

<https://www.youtube.com/watch?v=qJq9ZMB2Was>

<https://www.youtube.com/watch?v=i3J8gyARN-Y>

<https://www.w3schools.com/>