



**Eötvös Loránd Tudományegyetem**

**Informatikai Kar**

**Programozáselmélet és Szoftvertechnológia Tanszék**

# Szoftverfejlesztő cég CRM rendszere Laravel alapokon

**Témavezető:**

Dr. Gludovátz Attila

adjunktus, Ph.D.

**Szerző:**

Pergel Dániel

Programtervező informatikus BSc.

**Szombathely, 2021**

## Szoftverfejlesztő cég CRM rendszere Laravel alapokon

Egy 21. századi cég, vállalkozás életében - amelyik különféle termékek, szolgáltatások értékesítésével foglalkozik -, már elképzelhetetlenné vált, hogy valamilyen módszer segítségével ne tartsa nyilván az ügyfeleit. Az olyan cégek, amelyek nem fordítanak kellő figyelmet az ügyfélkapcsolat-kezelésre, potenciális vevői nagy részét elveszíti. Erre az egyik legjobb módszer egy CRM rendszer. Egy CRM rendszer rengeteg különféle modulból tevődik össze, az ügyfelekkel való együttműködéstől, a kiszolgáláson keresztül, a potenciális vásárlók felkutatásáig, szinte minden területet lefed. Egy ilyen rendszert készíték el egy szoftverfejlesztő cégnek.

A web-es alkalmazás célja, hogy támogassa a vállalkozás mindennapi munkáját, segítsen kapcsolatot tartani a meglévő ügyfelekkel, a vásárlási folyamatot gördülékenyen levezesse és felkutassa a lehetséges ügyfeleket. Egy ilyen rendszer segítségével a szoftverfejlesztő vállalat, amellett, hogy a mindennapi munkáját könnyebbé teszi, még az ügyfelei számát is növelni tudja.

A rendszer főbb funkciói:

- Meglévő és leendő ügyfelek adatainak kezelése.
- Digitális árajánlat készítése automatikusan.
- Értékesítési üzlet lehetőségének kezelése és támogatása.
- Kampányok készítése és nyomon követése.

Az alkalmazás web-es környezetben jön létre, így különféle eszközökön (asztali számítógép, táblagép, mobiltelefon stb.) is hozzáférést biztosítana a cég felhasználóinak. A program maga, az egyik legnépszerűbb PHP keretrendszerben valósul meg. A Laravel egy ingyenes, nyílt forráskódú PHP keretrendszer, amely az MVC, azaz Model-View-Controller tervezési minta alapján épül fel. Rengeteg hasznos csomag található a Laravel-hez, az egyik ilyen az úgynevezett Voyager, amely egy adminisztrációs felületet biztosít. A programomat tehát ebben a keretrendszer készíttem el, használva a Voyager csomagját is.

# Tartalomjegyzék

<b>1. Bevezetés.....</b>	<b>1</b>
1.1. A CRM rendszerek .....	2
1.2. A dolgozat tartalma .....	2
<b>2. Felhasználói dokumentáció .....</b>	<b>3</b>
2.1. Megoldott probléma leírása .....	3
2.2. Felhasznált módszerek rövid leírása.....	3
2.3. Program használatához szükséges összes információ.....	3
2.3.1. A program telepítése, elindítása .....	3
2.3.2. Bejelentkezés.....	8
2.3.3. Kezdőképernyő.....	9
2.3.4. Menü .....	10
2.3.5. Ügyfelek.....	11
2.3.6. Kapcsolattartók .....	12
2.3.7. Csoportok .....	13
2.3.8. Projektek .....	14
2.3.8.1. Árajánlat fázis.....	17
2.3.8.2. Értékesítési fázis.....	19
2.3.8.3. Lezárt projektek .....	20
2.3.9. Feladatok.....	20
2.3.9.1. Kampányok.....	21
2.3.10. Valuták.....	24
2.3.11. Alkalmazottak .....	24
2.3.12. Jogosultságok.....	24
2.4. Az ügyfélkapcsolat kezelés folyamatának bemutatása .....	24
2.4.1. Meglévő aktuális és múltbéli ügyfelek menedzselése .....	25
2.4.2. Potenciális ügyfelek felkeresése .....	27
<b>3. Fejlesztői dokumentáció.....</b>	<b>29</b>
3.1. A probléma részletes specifikációja .....	29
3.2. Felhasznált módszerek részletes leírása, a használt fogalmak definíciója.....	29
3.2.1. PHP .....	29
3.2.2. Laravel .....	30
3.2.3. MVC.....	30
3.2.4. Voyager.....	30
3.2.5. MySQL .....	30
3.3. Program logikai és fizikai szerkezetének leírása.....	30
3.3.1. Adatkezelési műveletek, BREAD-ek .....	30

3.3.1.1.	BREAD létrehozása .....	31
3.3.1.2.	BREAD használata .....	31
3.3.2.	MVC tervezési minta megvalósítása .....	32
3.3.3.	Modell réteg .....	33
3.3.3.1.	Campaign modell .....	33
3.3.3.2.	Contact modell .....	33
3.3.3.3.	ContactCustomer modell .....	34
3.3.3.4.	Customer modell .....	34
3.3.3.5.	Project modell .....	35
3.3.3.6.	Task modell .....	35
3.3.4.	Nézet réteg .....	35
3.3.5.	Vezérlő réteg .....	36
3.3.5.1.	CampaignController vezérlő .....	36
3.3.5.2.	CompleteController vezérlő .....	36
3.3.5.3.	PrintController vezérlő .....	37
3.3.5.4.	SaleController vezérlő .....	37
3.3.5.5.	StornoController vezérlő .....	37
3.3.5.6.	SuccessfulController vezérlő .....	38
3.3.5.7.	UnsuccessfulController vezérlő .....	38
3.3.6.	Adatszerkezet, adatbázis .....	38
3.3.6.1.	Egyed típusok és attribútumok .....	39
3.3.7.	Programcsomagok, modulok .....	47
3.3.7.1.	Árajánlat nyomtatása .....	47
3.3.7.2.	E-mail küldése .....	48
3.3.7.3.	Voyager gombok létrehozása .....	49
3.4.	Tesztelési terv és az eredményei .....	50
<b>4.</b>	<b>Összefoglalás és további fejlesztési lehetőségek .....</b>	<b>53</b>
4.1.	Összefoglalás .....	53
4.2.	Továbbfejlesztési lehetőségek .....	54
<b>5.</b>	<b>Táblázatjegyzék .....</b>	<b>55</b>
<b>6.</b>	<b>Ábrajegyzék .....</b>	<b>56</b>
<b>7.</b>	<b>Irodalomjegyzék .....</b>	<b>57</b>

## 1. Bevezetés

Azoknak a vállalatoknak, amelyek különféle termékek, szolgáltatások értékesítésével foglalkoznak, és napi szinten kerülnek kapcsolatba új és a már meglévő vevőikkel, elengedhetetlen, hogy valamilyen módon, nyilvántartsák az egyes ügyfeleiket, hozzájuk tartozó projekteket. Ahogy a vállalatok egyre inkább növekednek, erre a célra, már nem elegendő, hogy papír alapon vagy Microsoft Office Excel táblázatban vezessék az ügyfelek listáját. Nehézkessé, körülményessé válik az ügyfelekkel való kapcsolattartás, az ügyfelek nyilvántartása és az ügyeik intézése is. Ezekre a nehézségekre nyújtanak kiváló megoldást a CRM<sup>1</sup> rendszerek.

Célom egy webes környezetben működő CRM rendszer elkészítése volt, specifikusan egy szoftver fejlesztő cégnek. Ezen rendszer segítségével a vállalat alkalmazottai könnyedén tudják kezelni az ügyfeleiket, azok aktuális projektjeit, akár az irodából, akár otthoni környezetből is. Láthatják, hogy milyen munkafolyamatok vannak hátra az egyes projektekkal kapcsolatban. A rendszer segítségével egyszerűen elő tudnak állítani digitális árajánlatot, valamint különböző, személyre szóló kampányokat tudnak készíteni az ügyfeleknek, különböző ügyfélcsoportok alapján.

A munkám során törekedtem arra, hogy minden fontosabb modult megvalósítsak, azokat a funkciókat, amiket egy modern, mai CRM rendszer használ. Emellett igyekeztem a szoftverfejlesztő cég igényeire is összpontosítani, hogy számukra a lehető legmegfelelőbb rendszert tudjam megalkotni. Az elkészült program 4 fő modulból tevődik össze:

1. Ügyfelek adatainak kezelése
2. Digitális árajánlat készítése
3. Értékesítési üzlet lehetőségének kezelése, támogatása és a projektek nyomon követése
4. Kampányok készítése

---

<sup>1</sup> Customer Relationship Management – Ügyfélkapcsolat kezelő rendszer

## 1.1. A CRM rendszerek

A CRM az angol Customer Relationship Management rövidítése, magyarul ügyfélkapcsolat kezelő rendszernek szokás nevezni. A CRM rendszerek rengeteg különféle modulból tevődnek össze, az ügyfelekkel való együttműködéstől, a kiszolgáláson keresztül, a potenciális vásárlók felkutatásáig, szinte minden területet lefednek [1].

A CRM rendszerek rendkívül sokrétűek, használatukkal nemcsak az éppen aktuális ügyfeleinket, vevőinket tudjuk nyomon követni, hanem az egykori ügyfeleinket is, akikkel korábban álltunk kapcsolatban, illetve a potenciális partnereinket is. Láthatjuk a legfontosabb adataikat, hogy mikor és milyen ügyben keresték fel a cégünket, mik voltak azok a feladatok, projektek, amiket megvalósítottunk számukra. Új ügyfelek esetén megkönnyíti a szerződéskötés folyamatát, a meglévőknek pedig, az újraértékesítés folyamatában nyújt segítséget.

A CRM rendszerek főbb funkciói közé sorolhatók a vevők nyilvántartása, a folyamatok kezelése, feladatok követése és a kampánymenedzsment.

A CRM rendszerek tehát rendkívül összetettek, manapság már rengeteg cég használ valamilyen rendszert az ügyfelek kezelésére, menedzselésére. CRM rendszer nélkül a nagyobb, de már a közepes vállalatok is nehezebben boldogulnak. A rendszerek egyértelműen megkönnyítik a napi adminisztrációs feladatokat.

## 1.2. A dolgozat tartalma

A szakdolgozatomban be fogom mutatni, az elkészült program használatát, működését. A program telepítésétől kezdve, a funkciók használatáig minden területre kitérünk. A működés után a CRM rendszerek gyakorlati hasznát fogom bemutatni. Ezek után a fejlesztői dokumentáció fejezetben, a program megalkotásához felhasznált technológiákat ismertetem, majd magának a program szerkezetének, az adatbázis tervnek a felépítését mutatom be. Ezek után a program tesztelésére kerül sor, melyeket a különböző teszteseteken keresztül ismertetek. Végezetül pedig összegzem a programomat, a szakdolgozatomban leírtakat és megemlítem a továbbfejlesztési lehetőségeket is.

## 2. Felhasználói dokumentáció

Az alábbi fejezetben a program alapvető használatát mutatjuk be, a telepítéstől kezdve, az ügyfelek felvitelén keresztül a kampányok küldéséig, minden fontosabb elemet ismertetünk. A fejezet végén pedig egy példán keresztül szemléltetjük, a program sokrétűségét.

### 2.1. Megoldott probléma leírása

A program egy szoftverfejlesztő cég mindennapi munkáját hivatott megkönnyíteni, azon belül is az ügyfelekkel, potenciális vevőkkel való kapcsolattartást, a hozzájuk tartozó munkák, gyors és egyszerű kezelését, áttekintését. Új felkeresés esetén árajánlat készítését, illetve a régebbi ügyfelek újbóli felkeresését.

### 2.2. Felhasznált módszerek rövid leírása

A program egyik fontos kritériuma volt, hogy modern környezetben készüljön el, ne legyen lekorlátozva egyetlen számítógépre. Hordozható legyen, a felhasználók bárhol, akár távmunka során is hozzáférjenek a programhoz. Ezen szempontok alapján webes környezetben valósult meg a szoftver.

A mai modern webapplikációk, weboldalak már nem csak számítógépeken, hanem szinte bármilyen okos eszközön, így akár a telefonunkon, táblagépünkön is tökéletesen használhatók, a reszponzivitásnak köszönhetően. Nagy erőforrást sem igényelnek ezek a rendszerek, csupán internet hozzáférésre és egy böngészőre van szükségünk, hogy használhassuk a programot.

### 2.3. Program használatához szükséges összes információ

Ebben az alfejezetben a webes alkalmazás használatához szükséges összes elérhető információt leírom.

#### 2.3.1. A program telepítése, elindítása

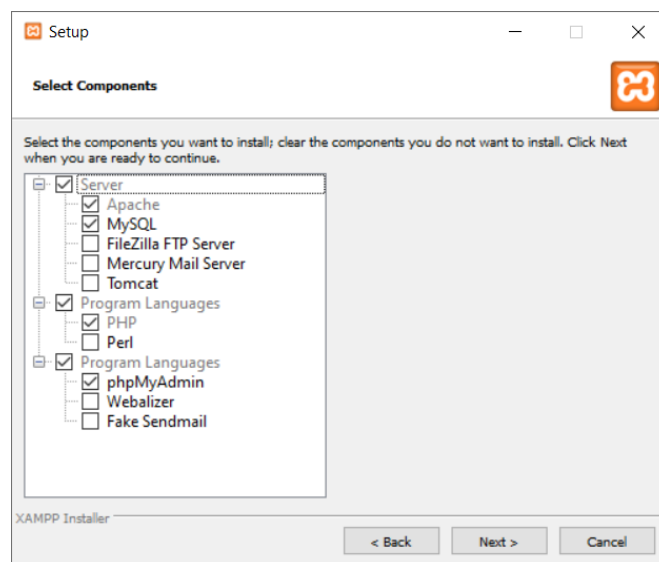
A program használatához le kell töltenünk és telepítenünk kell a XAMPP szervercsomagot, a Composer-t és a Node.js-t. Mindegyik program ingyenesen letölthető. A XAMPP szervercsomagot a <https://www.apachefriends.org/hu/index.html>

weboldalon tudjuk letölteni (1. ábra), itt, ki kell választanunk az operációs rendszerünknek megfelelő változatot, majd rákattintva elindul a letöltés.



1. ábra: XAMPP szervercsomag letöltése a hivatalos weboldalról

Letöltés után a telepítő fájlt futtatva haladjunk végig a telepítő varázsló lépésein. A program használatához elengedhetetlen az Apache webszerver, a MySQL adatbáziskezelő szerver és a PHP modulok telepítése (2. ábra). Ajánlott még a phpMyAdmin modul telepítése is, ezáltal könnyen tudjuk kezelni az adatbázisunkat.

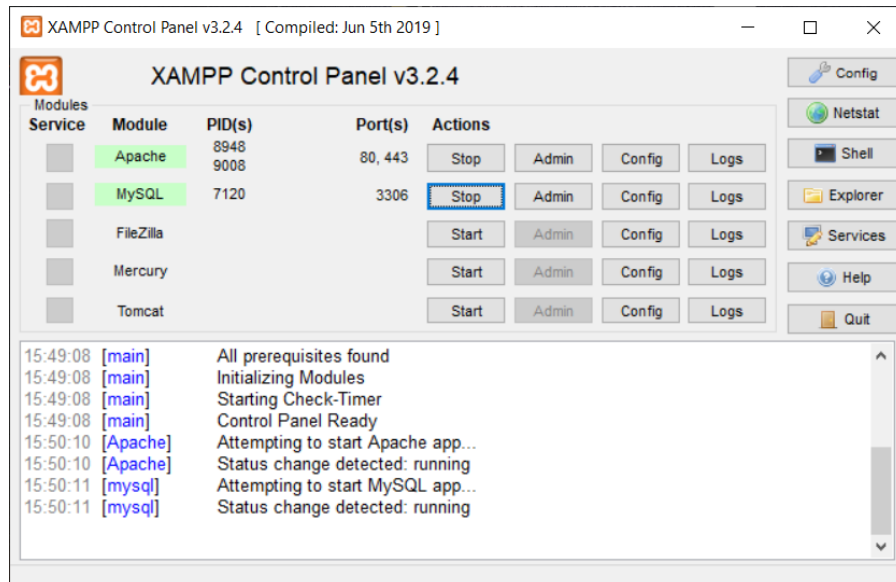


2. ábra: A XAMPP web szerver telepítő varázslója

A „Next” gombra kattintva a varázsló alapértelmezetten Windows operációs rendszer esetén a `C:\xampp` mappában telepíti a programot, amit akár meg is változtathatunk. Ajánlott azonban az alapértelmezett helyre telepíteni a XAMPP szerver, mert más helyre telepítve fájlhozzáférési problémák léphetnek fel.



A telepítést követően lehetőségünk van elindítani a XAMPP vezérlőpultját, melyet a kiválasztott mappából (alapértelmezett *C:\xampp*) is el tudunk érni a *xampp-control.exe* fájl elindításával. A vezérlőpulton kapcsoljuk be az Apache és MySQL komponenseket a „Start” gombok segítségével. Amennyiben a telepítés és az elindítás is sikeres volt, abban az esetben a modulok nevei zöld kiemelést kapnak (3. ábra).



3. ábra: A XAMPP web szerver vezérlőpanelja

Következő lépésként a Composer-t töltjük le és telepítjük fel. A Composer egy függőség- és verziókezelő a PHP nyelvhez, mely lehetővé teszi, a szükséges könyvtárak projektenkénti telepítését és frissen tartását. A Composer-t, a XAMPP-hoz hasonlóan egyszerűen le tudjuk tölteni és telepíteni. Letölteni a <https://getcomposer.org/download/> oldalon tudjuk (4. ábra). Az operációs rendszerünktől függően különböző módon tudjuk telepíteni (itt most a Windows-ra történő telepítés kerül bemutatásra). A Windows Installer részénél lévő *Composer-Setup.exe* linkre kattintva, letöltődik a telepítő.

## Download Composer Latest: v2.0.12

### Windows Installer

The installer - which requires that you have PHP already installed - will download Composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

Download and run [Composer-Setup.exe](#) - it will install the latest composer version whenever it is executed.

### Command-line installation

To quickly install Composer in the current directory, run the following script in your terminal. To automate the installation, use [the guide on installing Composer programmatically](#).

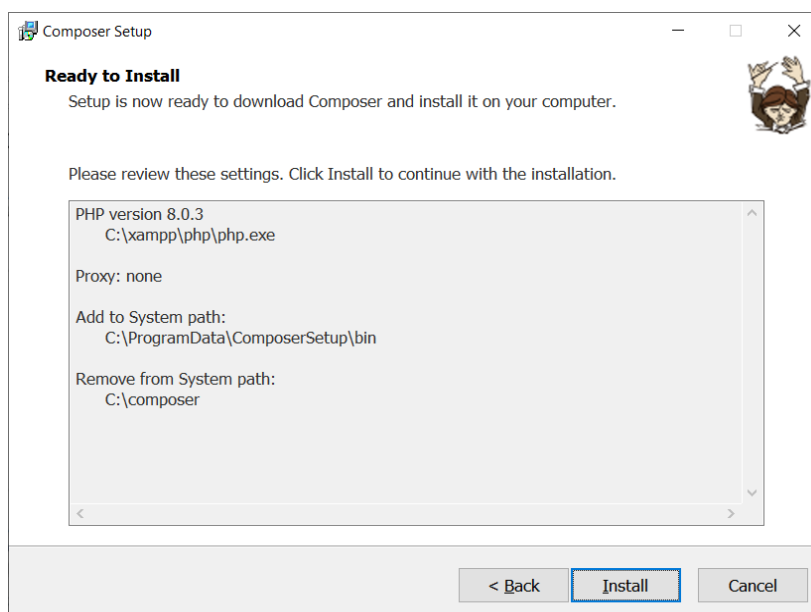
```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') === '756890a4488ce9024fc62c56153228907f1545c228516cbf63f8
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

This installer script will simply check some `php.ini` settings, warn you if they are set incorrectly, and then download the latest `composer.phar` in the current directory. The 4 lines above will, in order:

- Download the installer to the current directory
- Verify the installer SHA-384, which you can also [cross-check here](#)
- Run the installer
- Remove the installer

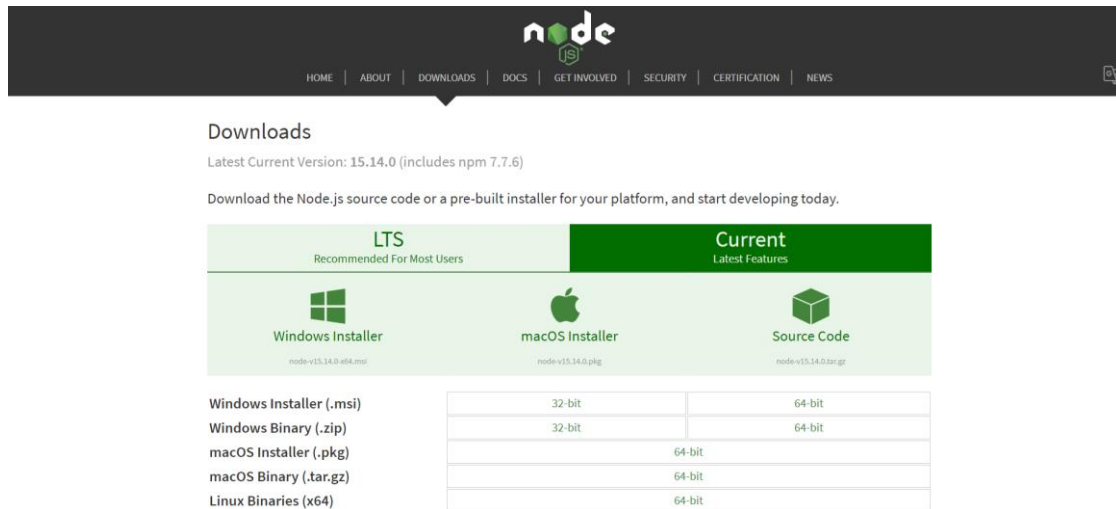
4. ábra: A Composer letöltése a hivatalos weboldalról

A telepítőt elindítva, a varázsló segítségével tudjuk telepíteni a Composer-t. A telepítés során az első párbeszéd ablakban válasszuk ki, hogy a „Developer mode”-ot telepítse. A következő lépésben a ki kell választanunk a használni kívánt parancssori PHP-t. Alapértelmezetten a `C:\xampp\php\php.exe` fájlt kell kiválasztanunk, amennyiben a XAMPP szervert a `C:\xampp` mappába telepítettük. Természetesen, hogyha a szerverünket más helyre telepítettük, úgy ebben az esetben, a telepítés helyén kell megkeresnünk, és betallóznunk a szükséges `php.exe` fájlt. A „Next” gombokra, majd végül az „Install” gombra kattintva elkezdődik a telepítés (5. ábra). A program néhány másodperc alatt feltelepül, a telepítő varázslóból pedig a „Finish” gombbal tudunk kilépni.



5. ábra: A Composer telepítő varázslója

Utolsóként pedig a Node.js-t kell letöltenünk és telepítenünk. A telepítőt letölteni a <https://nodejs.org/en/download/> oldalról tudjuk (6. ábra). Itt az operációs rendszerünknek és az általunk használt architektúrának megfelelő verziót töltjük le. Érdeemes átváltani a „Current” fülre, amellyel a legfrissebb verziót tudjuk letölteni és telepíteni.



6. ábra: A Node.js letöltése a hivatalos weboldaltól

A telepítőt elindítva a varázsló lépésein végig haladva egyszerűen tudjuk végrehajtani a telepítést.

Miután minden szükséges alkalmazást feltelepítettünk, a program használatához még, magának a CRM rendszerünknek a telepítését és beüzemelését kell elvégeznünk. A program összes állományát a XAMPP webservert könyvtárában lévő *htdocs* mappába kell másolnunk, a könyvtárszerkezet megtartásával.

Másolás után az adatbázis szerkezetet kell kialakítanunk. Ezt a XAMPP által telepített phpMyAdmin segítségével tudjuk megtenni. Menjünk a <http://localhost/phpmyadmin> oldalra. Itt az importálás menüpontra kattintva megtudjuk nyitni a *crmsystem\_db.sql* fájlt, melynek futtatásával létrehozuk a *crmsystem\_db* adatbázist a megfelelő táblákkal. Alapértelmezetten a XAMPP jelszó nélkül hozza létre a *root* felhasználót. Amennyiben ezt megváltoztatjuk – biztonsági okok miatt célszerű – úgy, az imént átmásolt mappában található *.env* fájlban is rögzítenünk kell a változást.

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

7. ábra: A Laravel .env fájlban az adatbázis kapcsolat beállításai

Fontos, hogy a *DB\_DATABASE* értéke megegyezzen az adatbázisunk nevével, a *DB\_USERNAME* és a *DB\_PASSWORD* mezőkben pedig a felhasználónév, jelszó párost kell rögzítenünk (7. ábra).

Végezetül, miután minden szükséges programot telepítettünk, a programunkat bemásoltuk a megfelelő helyre, és az adatbázist is konfiguráltuk, elindíthatjuk a programunkat. Ehhez el kell külön indítanunk a Laravel szerver szolgáltatását. Parancssori ablakban lépünk be a CRM rendszer projekt mappájába. Itt a következő parancsot írjuk be: *php artisan serve*. Ezzel tudjuk elindítani a Laravel szerverét. Ezt követően a <http://127.0.0.1:8000/admin> címen tudjuk ténylegesen elindítani a programunkat.

### 2.3.2. Bejelentkezés

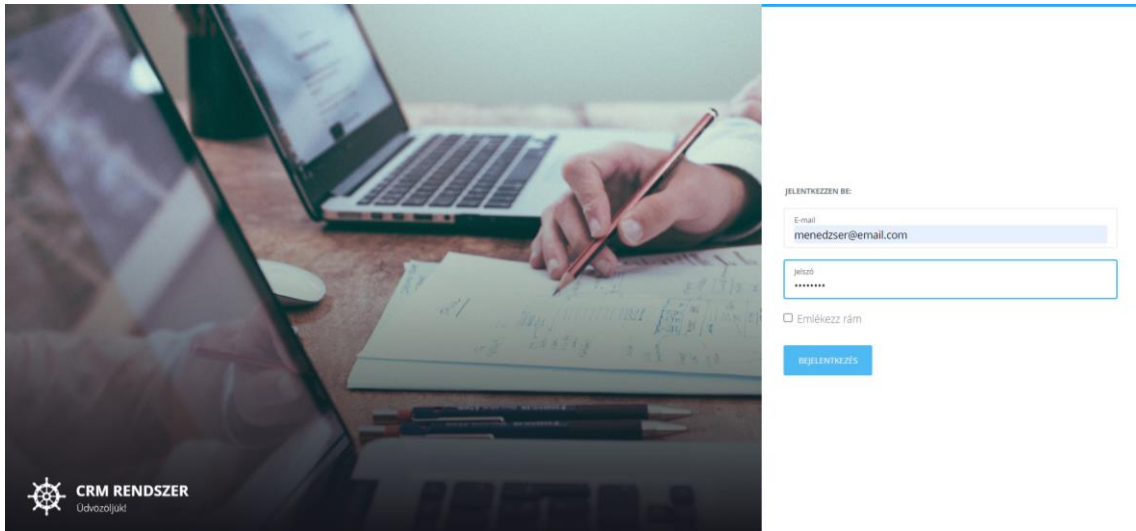
A program elindítását követően a CRM rendszer bejelentkező felülete fogad minket (8. ábra). Itt a megfelelő e-mail cím, jelszó párossal tudunk bejelentkezni. A programot csakis a bejelentkezett felhasználók tudják használni. Több felhasználóval is be tudunk jelentkezni és a szerepköröktől függően, más-más jogok állnak a rendelkezésünkre.

Bejelentkezéshez szükséges e-mail cím - jelszó párosok (az első oszlopban a szerepkör neve látható):

1. táblázat: Előre beállított szerepkörök, felhasználónevek és jelszavaik

Menedzser Felhasználó	<i>menedzser@email.com</i>	<i>password</i>
Normál Felhasználó	<i>felhasznalo@email.com</i>	<i>password</i>
Normál Alkalmazott	<i>alkalmazott@email.com</i>	<i>password</i>
Normál Munkatárs	<i>munkatars@email.com</i>	<i>password</i>

A programot innentől kezdve menedzser módban fogom bemutatni. Miután beírtuk az e-mail címet és jelszót, kattintsunk a „Bejelentkezés” gombra. Sikertelen bejelentkezés esetén a program egy figyelmeztetést fog küldeni, hogy az e-mail cím, jelszó páros nem megfelelő. Sikeresen bejelentkezés esetén, a program kezdőképernyője fogad minket.



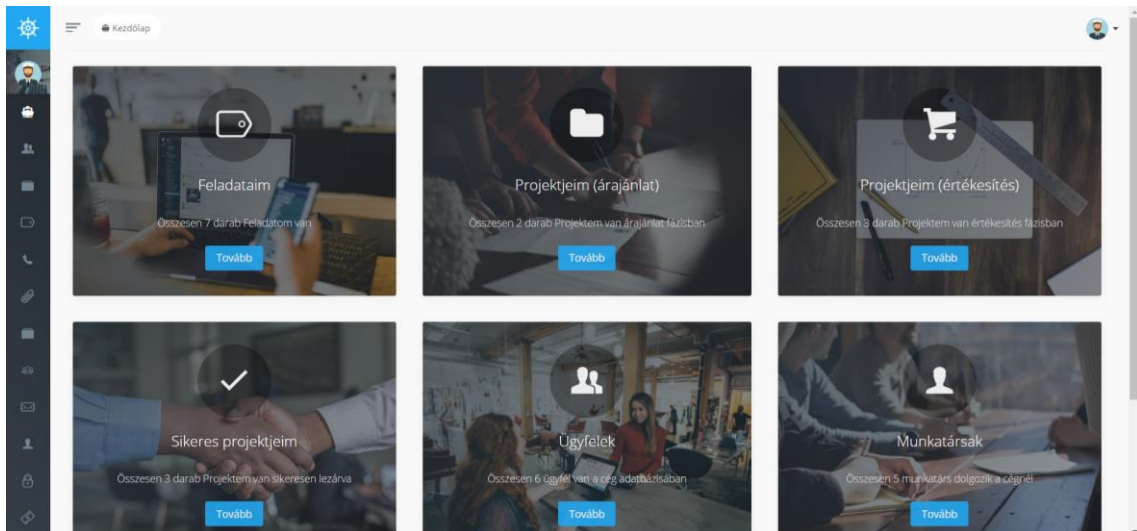
8. ábra: A CRM rendszer bejelentkező felülete

### 2.3.3. Kezdőképernyő

Az alkalmazás elindításakor a kezdőképernyő fogad minket (9. ábra). Itt összesen hat darab widget<sup>2</sup> található. A widget-ek segítségével mindegyik felhasználó személyre szabottan láthatja, hogy mennyi darab feladata van még hátra, mennyi darab projektje található a különböző – árajánlat, értékesítési, sikeresen lezárt – fázisokban, valamint, hogy mennyi ügyfél van felvéve a cég rendszerébe, és hogy mennyi alkalmazott dolgozik a vállalatnál. A widget-ek nem csak egyfajta statisztikát adnak nekünk, hanem segítségükkel, gyorsabban el tudjuk érni az adott menüpontokat. Mindegyik widget-en található egy „Tovább” gomb, amelyre kattintva bejön az adott oldal. Például a „Feladataim” widget-nél lévő „Tovább” gombra kattintva, a „Feladataim” menüpontot érhetjük el. A widget-ek tehát segítenek gyorsabban eligazodni az egyes menüpontok között, és számszerűsítik az aktuális feladatainkat, projektjeinket.

---

<sup>2</sup> Grafikus alkotóelemek, melyek elérhetővé tesznek valamilyen tartalmat.

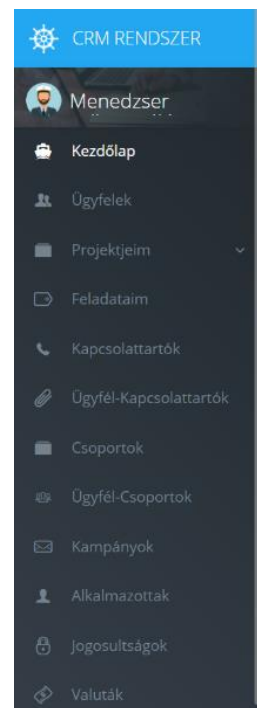


9. ábra: A program kezdőlapja, a widget-ekkel

#### 2.3.4. Menü

Az alkalmazásban nem csak a widget-ek segítségével tudunk navigálni. Bal oldalon található egy „slide” menü (10. ábra), amelynek segítségével a rendszer minden menüpontját el tudjuk érni. A program teljes egészében reszponzív felépítésű, így akár asztali számítógépen, akár mobiltelefonon használjuk, mindig tökéletes lesz a megjelenése.

A főmenü segítségével tehát az egyes menüpontok érhetőek el. Ezen kívül, hogyha a jobb felső sarokban található profilképünkre kattintunk, úgy egy legördülő menü segítségével, elérhetővé válik a profil oldalunk, és itt tudunk kijelentkezni a rendszerből. A „Profil” menüpontra kattintva tudjuk szerkeszteni a saját profilunkat. Meg tudjuk változtatni a nevünket, e-mail címünket, jelszavunkat, illetve a profilképünket is.



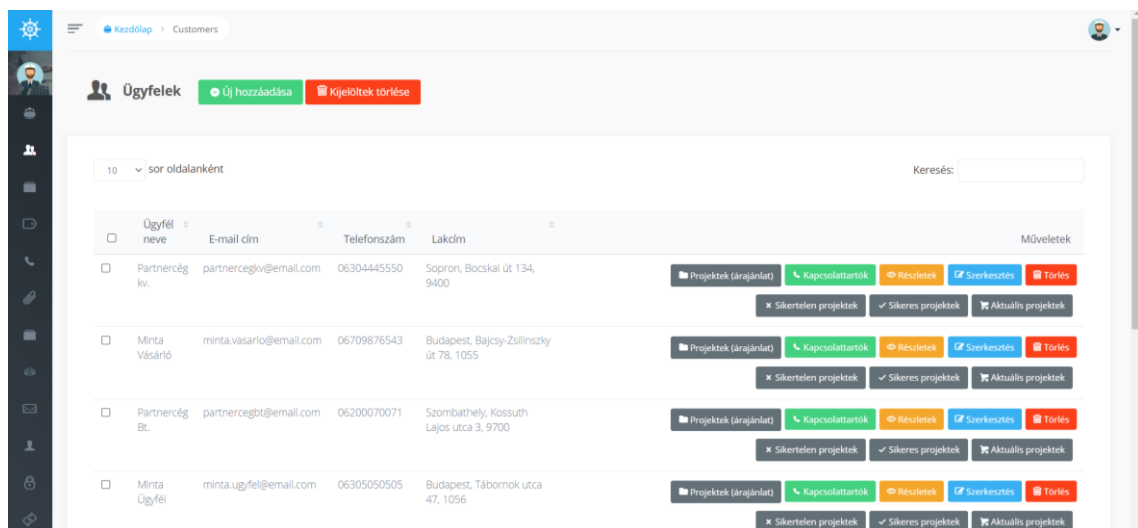
10. ábra: A menürendszer

A programban minden nézet felépítése rendkívül hasonló. Középen, táblázatos nézetben láthatjuk az egyes rekordjainkat, a lap tetején tudunk mindig új adatokat felvenni, míg a lap alján tudunk lapozni az egyes oldalak között. Be tudjuk állítani, hogy mennyi adat jelenjen meg egyszerre a képernyőn, illetve minden menüpont rendelkezik egy multifunkcionális keresővel. Itt bármilyen adat alapján tudunk keresni. Például az ügyfeleink listájában, ugyanazzal a kereső segítségével, akár telefonszámra, akár névre is kereshetünk. Ez nagyban megkönnyíti a program használatát.

### 2.3.5. Ügyfelek

A CRM rendszer középpontjában az ügyfeleink találhatók. Őket az „Ügyfelek” menüpont alatt találjuk (11. ábra). Itt táblázatos formában láthatjuk az összes ügyfelünket, tudunk új ügyfelet létrehozni, a meglévőket módosítani, törölni és az adataikat részletesebben böngészni. Alapértelmezetten az egyes ügyfeleknek csak a nevük, e-mail címük, telefonszámuk és a címük jelenik meg, viszont a részletek gombra kattintva elérhető az összes hozzájuk tartozó adat, mint például a felvétel dátuma vagy az ügyfél típusa. A szerkesztés gombra kattintva értelemszerűen az adott partnernek az adatait tudjuk szerkeszteni.

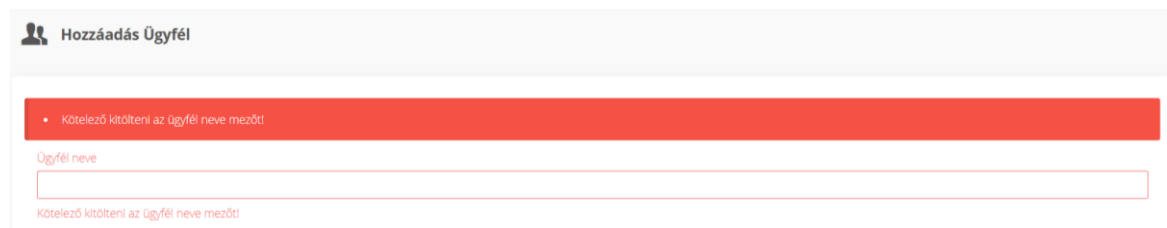
Ezen kívül még öt gomb tartozik a vevőinkhez. Ezeknek a segítségével ki tudjuk listázni az egyes ügyfeleinkhez tartozó árajánlat, értékesítési, sikeresen lezárt és sikertelenül lezárt fázisban lévő projektjeiket, valamint a vevőhöz tartozó kapcsolattartókat. Ekkor nem a teljes adatsor jelenik meg, csupán csak egy szűrt listát kapunk. Ha például rákattintunk bármely ügyfelünknél az „Aktuális projektek” gombra, abban az esetben, a hozzá tartozó, értékesítési fázisban lévő projektek jelennek meg. Hasonló módon működik a többi gomb is.



11. ábra: Az „Ügyfelek” oldal

Új ügyfelet, a lap tetején lévő „Új hozzáadása” gombbal tudunk felvenni. Ekkor egy új beviteli űrlap fogad minket, ahol a partnerünk minden fontosabb adatát ki kell töltenünk, mint például a nevét, elérhetőségeit – e-mail cím, telefonszám, cím –, illetve találunk itt egy megjegyzés rovatot is, ahol különböző információkat tudunk rögzíteni. Például mivel foglalkozik az ügyfél, honnan hallott a cégünkről, és egyéb hasonló

információkat, melyekből akár pár év elteltével is betudjuk azonosítani az ügyfelet. Ezen kívül még ki kell választanunk egy legördülő menüpont segítségével, hogy az adott partnerünk céges ügyfél-e vagy magánszemély. Végezetül a „Mentés” gombra kattintva új partnerünk máris rögzül a rendszerünkben és megkezdhetjük vele a közös munkát. Adatbevitel során amennyiben bármelyik adatot rosszul, vagy hiányosan töltöttük ki, a rendszer hibaüzenetet küld (12. ábra), és nem engedi elmenteni az ügyfelet, egészen addig, ameddig ezeket a hibákat nem javítottuk ki.



12. ábra: Hibásan vagy üresen kitöltött adatnál lévő figyelmeztetés

### 2.3.6. Kapcsolattartók

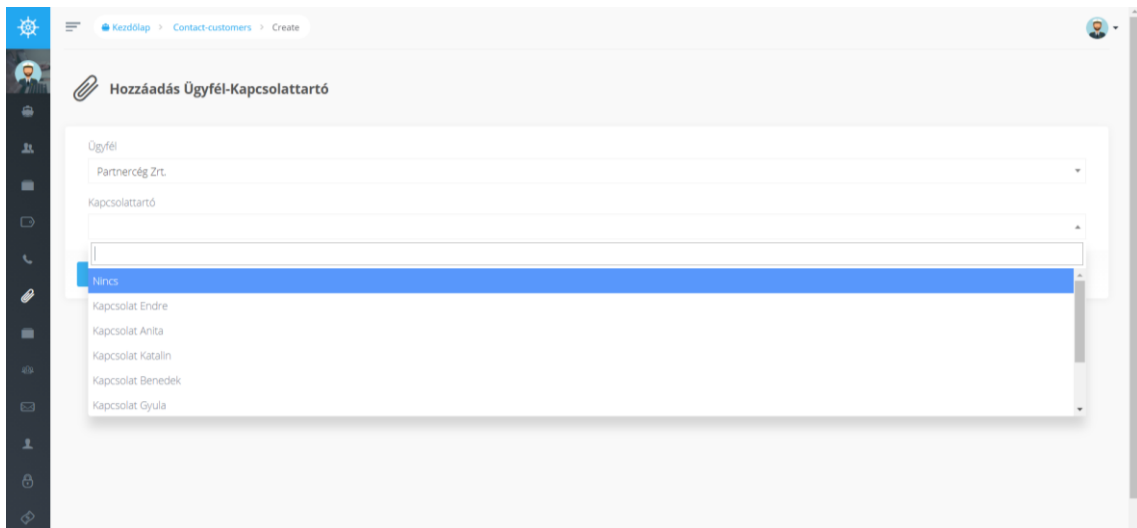
Minden ügyfélhez tartoznak kapcsolattartók. Ők azok a személyek, akik képviselik a vevőt, kommunikálnak a cégünkkel. Őket keressük, ha további egyeztetések szükségesek, vagy ha kérdéseink merültek fel az adott projekttel vagy a céggel kapcsolatban. Ezért kulcsfontosságú a kapcsolattartók rögzítése. Egy ügyfélhez, legyen szó akár egy cégről vagy magánszemélyről, több kapcsolattartó is tartozhat. Őket a „Kapcsolattartók” menüpont alatt érhetjük el. Ilyenkor a teljes listát, az adatbázisban található összes személyt láthatjuk. Amennyiben a fentebb már említett, „Ügyfelek” nézetről navigálunk a kapcsolattartókhoz, akkor az adott ügyfélhez tartozó szűrt listát láthatjuk. Csak úgy, mint az ügyfeleinknél, a kapcsolattartóknál is láthatjuk a részletesebb adataikat, tudjuk őket szerkeszteni vagy törölni. Ehhez, akár a teljes, akár a szűrt listán belül a „Részletek”, „Szerkesztés” és „Törlés” gombokra kell kattintanunk.

Új kapcsolattartó létrehozása két lépésből tevődik össze. Elsőként rögzítenünk kell magát a kapcsolattartót. Ezt az „Új hozzáadása” gomb által tudjuk megtenni. Minden fontosabb adatát, így például a nevét, e-mail címét, telefonszámát, lakcímét is fel kell vinnünk a rendszerbe. Ezekon kívül még itt is található egy megjegyzés rovat, ami csakúgy mint az ügyfeleknél, arra szolgál, hogy minden fontosabb információt fel tudjunk vinni az adott kapcsolattartóval kapcsolatban, mint például, hogy mikor



dolgozik, mikor elérhető, milyen pozícióban van az adott cégnél, és egyéb további hasznos információkat, melyek megkönnyítik a kommunikációt.

Miután felvittünk a rendszerünkbe a kapcsolattartót, még nem végeztünk, ugyanis külön, manuálisan is össze kell még kapcsolnunk a megfelelő ügyféllel. Ezt a lépést az „Ügyfél-kapcsolattartók” menüpont alatt tudjuk megtenni. Itt táblázatos formában láthatjuk, az eddig összekapcsolt ügyfeleket és kapcsolattartókat. Az „Új hozzáadása” gombra kattintva két legördülő lista jelenik meg (13. ábra). Az elsőből a megfelelő ügyfelet, a másodikból a hozzá tartozó kapcsolattartót kell kiválasztanunk. Erre a műveletre azért van külön szükség, hogy a későbbiekben, a projektek létrehozásánál, egy lépésben tudjuk az adott projekthez az ügyfelet és a kapcsolattartót is hozzácsatolni, ezzel könnyítve meg a későbbi munkát.



13. ábra: Az ügyfél és a kapcsolattartó összekapcsolása

### 2.3.7. Csoportok

A csoportok az ügyfelekkel történő kapcsolattartás egyik legfontosabb része, ugyanis ezek alkotják a kampányok alapjait. Különböző ügyfélcsoportokat tudunk létrehozni, mindezt a „Csoportok” menüpont alatt tudjuk megtenni. Egy csoportnak két adata van, a csoport megnevezése és egy rövidebb leírás róla, amiben kifejezhetjük, hogy pontosan milyen ügyfelek kerüljenek a csoportba, milyen részleteket kell tudni az adott ügyfélcsoportról.

Természetesen a „Részletek”, „Szerkesztés” és „Törlés” gombok itt is megtalálhatóak, ezeken kívül pedig, minden csoporthoz további két művelet tartozik. Az „Ügyfelek” és a

„Kampányok” gomb. Mindkettő egy-egy szűrt listát ad vissza. Értelemszerűen az „Ügyfelek” gombra kattintva, azoknak a vevőknek a listáját láthatjuk, akik az adott csoporthoz tartoznak. Míg a „Kampányok” gomb megnyomásával, azokat a kampányokat láthatjuk, amelyek az adott csoporthoz lettek létrehozva.

Ügyfeleket különböző csoportokhoz rendelni az „Ügyfél-csoportok” menüpont alatt tudunk. Ennek működése nagyon hasonló, mint az „Ügyfél-kapcsolattartók” menüpontnak a működése. Elsőként táblázatos nézetben láthatjuk az eddig összekapcsolt ügyfeleket és csoportokat, majd az „Új hozzáadása” gombra kattintva két legördülő listát találunk. Az elsőből a megfelelő ügyfelet kell kiválasztanunk, a másodikból pedig azt a csoportot, amelyikbe szeretnénk, hogy a partnerünk belekerüljön. Egy ügyfelet, akár több csoportba is belerakhatunk, attól függően, hogy milyen megkereséssel fordult hozzánk az adott vevő, és hogy a későbbiekben milyen kampányok kapcsán akarjuk majd megkeresni.

Különösen fontos, hogy jól definiáljuk a csoportjainkat és a megfelelő ügyfeleket adjuk hozzájuk. Hiszen, ahogy már fentebb említésre került, ez az alapja majd a későbbi kapcsolattartásnak. Például azt az ügyfelet, aki egy online webáruházzal kapcsolatban keres fel minket, nem célszerű a „kutatás és fejlesztés” csoportba rakni, ugyanakkor a „webfejlesztés”, vagy „webshop” csoporthoz mindenképpen érdemes hozzárendelni.

#### 2.3.8. Projektek

Az alkalmazásban kiemelt szerepe van a projekteknek. Az ügyfelek ugyanis különböző feladatokkal, kérésekkel, munkákkal fordulnak hozzánk, amiket mi projektekként rögzítünk a rendszerben, amelyeket a „Projektjeim” menüpont alatt érhetünk el. Ekkor egy összesített listában, azok a projektek jelennek meg, amelyeknek az éppen bejelentkezett felhasználó a projekt gazdája, így mindenki a saját projektjeit tudja itt kilistázni. Van lehetőség ügyfélre szűrt listát is megjeleníteni, ehhez az „Ügyfelek” menüpontról kell az egyes ügyfelekhez tartozó projektek gombot megnyomni. Ekkor csakis az adott vevőhöz kapcsolódó projektek fognak megjelenni, ebben a listában viszont már nem csak a sajátjaink, hanem a cég többi alkalmazottja által menedzselte projekt is megjelenik.

A projektek négy különböző nézetben jelennek meg. **(1) Árajánlat fázis, (2) Értékesítési fázis, (3) Sikeresen lezárt, illetve (4) Sikertelenül lezárt projektek.** Ezekre külön-külön tudunk rámenni a „Projektjeim” menüpontnál, vagy pedig az „Ügyfelek” menüpontnál, az egyes ügyfelekhez tartozó gombokra kattintva. Mindegyik fázisban különböző műveleteket tudunk végrehajtani a projektjeinkkel.

Amikor létrehozunk egy új projektet az automatikusan árajánlat fázisba fog kerülni. Projektet létrehozni az „Új hozzáadása” gomb segítségével tudunk, bármely projekt nézetükben (14. ábra). Ekkor számos adatot meg kell adni a programnak, ezek a következők:

- **Projekt neve:** Értelemszerűen ide az adott projekt megnevezése kerül. Érdemes rövid, lényegre törő és egyértelmű nevet adnunk, hogy mindig tudjuk miről is szól az adott megbízatás.
- **Határidő:** Minden munkánál van egy határidő, amire az ügyfél várja a projekt elkészültét. Ezt a határidőt kell itt rögzíteni. Fontos eleme a projekteknek, hiszen ez alapján állandóan nyomon tudjuk követni, hogy melyik munkára mennyi időnk van még, hogy időben elkészítsük. Ha bármelyik munkánál túlszaladunk ezen a határidőn, az az ügyfél elvesztését is jelentheti. Kitöltése roppant egyszerű, egy dátum mezőből kell kiválasztani a határidő pontos napját. Tudunk lépkedni az évek és a hónapok között, majd a nap lenyomásával rögzül a dátum.
- **Munkaórák ára:** A munkaórák ára elsődlegesen a projekt teljes munkaköltsége szempontjából számítanak. Szoros összefüggésben vannak a valutával és a munkaórák számával, ugyanis a program a digitális árajánlatban ezek alapján számítja ki a projekt összköltségét.
- **Valuta:** A valuta mezőnél egy legördülő listából kell kiválasztani, azt a valutát, amelyben a vevő szeretné rendezni a projekt költségeit. Ezeket a menedzser kezelni is tudja, a „Valuták” menüpont alatt. Ennek köszönhetően tehát, a projektünket, és emiatt az árajánlatot nem csak forintban tudjuk elkészíteni. Ez megkönnyíti az esetleges külföldi vevőkkel való munkafolyamatot, hiszen ők az árajánlatban eleve a saját országuknak megfelelő valutát láthatják.

- **Munkaórák száma:** A munkaórák ára és a valuta mellett a munkaórák száma határozza meg a projekt teljes költségét. Ide kell azt a számot beírni, amennyi órát mi előzetesen szeretnénk ráfordítani az adott munkára.
- **Megjegyzés:** Ide kerülhet bármilyen információ az adott munkával kapcsolatban. Ezek az apró megjegyzések szolgálnak arra, hogy a projekt szempontjából minden lényeges információt lejegyezzünk ide. Bármilyen feljegyzést leírhatunk, amivel kapcsolatban úgy érezzük, hogy segítheti a munkánkat. Az ügyfél által kért változtatásokat, új információkat is mindenképpen érdemes itt rögzíteni.
- **Ügyfél-kapcsolattartó:** Egy legördülő listából itt tudjuk kiválasztani a projekthez tartozó ügyfél-kapcsolattartó párost. Itt az „Ügyfél-kapcsolattartók” menüpontban felvett adatokat láthatjuk. Tehát egy lépésben tudjuk a megfelelő pároshoz hozzákapcsolni a projektet. Nem kell külön keresgélni az ügyfelet, majd a hozzátartozó kapcsolattartót, egyszerűen csak az adott párost kell kiválasztani. Ezzel pedig már rögzült is az adott ügyfélhez a frissen létrehozott projektünk.
- **Csoport:** Itt a már korábban létrehozott csoportok közül tudunk egyet hozzárendelni az adott projekthez. Tehát a csoportokat nem csak az ügyfelekkel tudjuk összekötni, hanem az egyes projektjeinkkel is. Ez abban segít minket, hogy a munkáinkat tudjuk kategorizálni, beazonosítani, egyfajta címkével ellátni. Visszatekintve majd a korábbi projektjeinkre láthatjuk, hogy melyiket hoztuk létre többek között webfejlesztés, tanácsadás, adatelemzés céljából, így ha egy későbbi munka esetében szeretnénk ötletet, információt meríteni egy korábbiából, akkor ezáltal könnyebben ki tudjuk választani, hogy melyik projektre is van szükségünk.
- **Projekt menedzser:** A rendszer automatikusan az éppen bejelentkezett felhasználót jelöli ki az adott projekt menedzserének. Ezt egy legördülő menü segítségével meg tudjuk változtatni, a cég bármely alkalmazottját hozzá tudjuk rendelni egy projekthez. A projektmenedzsernek többlet joga van az egyes projektekkel kapcsolatban, illetve az ő felelőssége, hogy az adott munka, a megfelelő időben és minőségben készüljön el.

Végezetül a „Mentés” gombra kattintva a projekt bekerül a rendszerünkben. Az alkalmazás itt is figyelni természetesen, hogy minden adatot helyesen töltöttünk-e ki,

amennyiben valahol hiányosságot észlel, úgy figyelmeztet minket erre. Amennyiben viszont, a létrehozás során minden rendben történt, úgy a projektünk automatikusan árajánlat fázisba kerül.

14. ábra: Új projekt létrehozása

### 2.3.8.1. Árajánlat fázis

Az árajánlat fázisban lévő munkáinkat a „Projektjeim” menüponton belül az „Árajánlat Fázis” menü alatt érhetjük el (15. ábra). Ebben az esetben, a bejelentkezett felhasználóhoz tartozó projektek jelennek csak meg. Fontos ugyanakkor, hogy valamilyen módon láthassuk azokat a projekteket is, melyek nem hozzánk tartoznak, de valamilyen oknál fogva szeretnénk őket megnézni. Ebben az esetben a másik módon kell rákeresnünk a projektekre, az „Ügyfelek” menüpontról. Itt már, azok a munkák is megjelennek, amelyeknek nem mi vagyunk a projektmenedzserei.

Projekt név	Projekt határideje	Ügyfél-Kapcsolattartó	Csoport	Projekt menedzser	Műveletek
Minta projekt 12	2021-07-21	Partnercég kv. - Kapcsolat András	Csoport02	Menedzser Felhasználó	<input type="checkbox"/> Feladat hozzáadása <input type="checkbox"/> Feladatok <input type="checkbox"/> Árajánlat nyomtatása <input type="checkbox"/> Részletek <input type="checkbox"/> Szerkesztés <input type="checkbox"/> Törölés <input type="checkbox"/> Lezárás (sikertelen) <input type="checkbox"/> Értesítés
Minta projekt 07	2021-07-22	Minta Vásárló - Kapcsolat Mónika	Csoport01	Menedzser Felhasználó	<input type="checkbox"/> Feladat hozzáadása <input type="checkbox"/> Feladatok <input type="checkbox"/> Árajánlat nyomtatása <input type="checkbox"/> Részletek <input type="checkbox"/> Szerkesztés <input type="checkbox"/> Törölés <input type="checkbox"/> Lezárás (sikertelen) <input type="checkbox"/> Értesítés

15. ábra: A „Projektjeim - Árajánlat fázis” oldal

A projektmenedzsernek az adott projekttel kapcsolatban mindig többlet joga van. Míg mások projektjeinél csak a részleteket, a feladatokat és az árajánlatot tudjuk megtekinteni, addig a saját projektjeinknél tudunk feladatot is hozzáadni, illetve mozgatni tudjuk a projektet a különböző fázisok között. Az árajánlat fázisban lévő projektekkel tehát ezeket a műveleteket tudjuk elvégezni:

- *„Feladatok”*: Erre a gombra kattintva tudjuk szűrt listában megjeleníteni az adott projekthez tartozó feladatokat. Ilyenkor a hozzá tartozó összes feladatot láthatjuk, meg tudjuk nézni, hogy mik azok, amik még hátravannak ahhoz, hogy a projektünket véglegesnek tekinthessük.
- *„Feladatok hozzáadása”*: Az adott projekthez új feladatot, csakis a projektmenedzser tud felvenni. A *„Feladat hozzáadása”* gombra kattintva specifikusan, az adott megbízatáshoz kapcsolódó feladatokat tudunk létrehozni. Így tehát a létrehozásnál, nem kell külön még a projektet is kiválasztani egy legördülő listából, az ugyanis már előre ki van töltve a program által.
- *„Árajánlat nyomtatása”*: A program segítségével rendkívül gyorsan, egy gombnyomásra tudunk digitális árajánlatot készíteni (16. ábra). Miután felvettük az ügyfelünket, bevittük a rendszerbe az általa kért projektet, valamint hozzáadtuk az előzetesen tervezett feladatok listáját, utána van lehetőség elkészíteni az árajánlatot. Az árajánlatba automatikusan bekerül az ügyfél neve, a projekthez tartozó kapcsolattartó neve, e-mail címe, de bekerül többek között a projekt neve, határideje és a hozzá tartozó feladatok is. Ezekon kívül a már korábban megadott munkaórák ára, a valuta és a munkaórák száma segítségével, láthatjuk a teljes projekt összköltségét is az árajánlaton. Valamint a program automatikusan ráteszi még a cégünk logóját, nevét, az aláírás helyét, a projekt menedzser nevét és elérhetőségeit, valamint az árajánlat lejárat dátumát is. Az *„Árajánlat nyomtatása”* gomb lenyomásával, PDF formátumban megkapjuk az árajánlatot. Ezzel egy üzletkötés során rengeteg időt tudunk spórolni, hiszen nem kell külön program segítségével árajánlatot készíteni, azt megformázni, ugyanis a program automatikusan elvégzi, csupán egyetlen gombnyomás segítségével. Az elkészült fájl neve a projekt neve lesz, mögé rakva egy alsó vonással az árajánlat szót. Például *mintaprojekt\_árajánlat.pdf*

- „Értékesítés”: Amennyiben az ügyfél elfogadta az árajánlatot, abban az esetben megkezdődhet vele a közös munka. Így tehát ez a projektünk már nem árajánlat fázisba van, hanem ennek a gombnak a lenyomásával áttehetjük értékesítési fázisba. Így tehát a projektünk már, az „Értékesítési Fázis” menüpontból érhető el.
- „Lezárás (sikertelen)”: Viszont, ha a vevő nem fogadja el az árajánlatot, abban az esetben a projektünket le kell zárunk és sikertelenné kell nyilvánítanunk. Ezt ennek a gombnak a megnyomásával tudjuk megtenni.

**CRM 2021**

## Árajánlat

	Címzett		Feladó
<b>Név:</b>	Minta Vásárló	<b>Név:</b>	Menedzser Felhasználó
<b>Kapcsolattartó:</b>	Kapcsolat Mónika	<b>E-mail:</b>	menedzser@email.com
<b>E-mail:</b>	kapcsolatmonika@email.com		

**Projekt megnevezése:** Minta projekt 07

**Célok és feladatok leírása:**

1. Feladat\_alfa
2. Feladat\_beta
3. Feladat\_epsilon
4. Feladat\_dzéta
5. Feladat\_théta
6. Feladat\_kappa

**A projekt határideje:** 2021-07-22.  
**A projekt munkaideje:** 20 óra  
**A projekt ára:** 4400 HUF / munkaóra  
**A projekt teljes ára:** 88000 HUF

Szombathely, 2021-04-26. Menedzser Felhasználó

Az árajánlat érvényességi ideje: 2021-05-26.  
Mintavállalat - 2021

16. ábra: A PDF formátumban kinyomtatott digitális árajánlat

### 2.3.8.2. Értékesítési fázis

Miután árajánlat fázisból átléptettük a projektünket értékesítési fázisba, az „Értékesítési Fázis” menüpont alól tudjuk elérni. Ebben a fázisban is különböző műveleteket tudunk

végrehajtani az egyes munkákkal kapcsolatban. A szokásos részletek, szerkesztés, törlés műveleteken kívül, itt is meg tudjuk nézni a projekthez kapcsolódó feladatok listáját, valamint tudunk új feladatot hozzárendelni, hasonló módon, mint az árajánlat fázisnál.

A projektmenedzser tudja az aktuális munkát tovább léptetni a következő fázisba. Egy értékesítési fázisban lévő projektet kétféleképpen lehet lezárni. Sikeres együttműködés esetén a „Lezárás (sikeres)” gombra kell kattintani. Amennyiben viszont a projekt valamilyen oknál fogva mégsem valósul meg, akkor a „Lezárás (sikertelen)” gombra kattintva, a rendszer sikertelennek minősíti az adott munkát.

#### 2.3.8.3. Lezárt projektek

Minden munkának kétféle befejezése lehet: vagy sikeres volt az adott együttműködés, vagy pedig sikertelen. A sikeresen lezárt projekteket a „Lezárt (Sikeres)” menüpont alatt érhetjük el, míg a sikertelen munkáinkat a „Lezárt (Sikertelen)” menüpontból. Mindkét lezárási módnál van lehetőségünk „stornózni” az adott projektet, amennyiben az ügyfél még módosítani szeretne rajta, vagy valamivel még nem elégedett. A „Stornó” gomb megnyomásával, a projektünk visszakerül értékesítési fázisba.

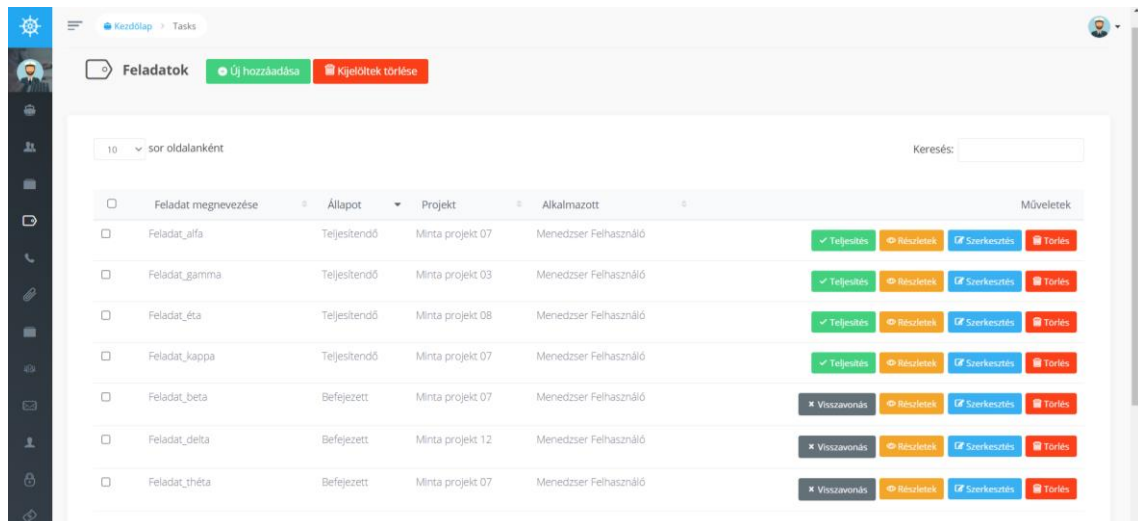
#### 2.3.9. Feladatok

A feladatok a projektek megvalósításának főbb lépései. Ezek határozzák meg azokat a pontokat, amelyeket végre kell hajtani ahhoz, hogy a projekt elkészüljön. A feladatokat két helyről tudjuk elérni. Vagy a „Feladataim” menüpont alatt, ekkor az összes hozzánk, azaz a bejelentkezett felhasználóhoz tartozó feladatot láthatjuk (17. ábra). A másik mód pedig, bármely projektjeim menüpont alatt, az egyes sorokban a „Feladatok” gombra kattintva. Ebben az esetben egy szűrt listát látunk, amely csak az adott munkához tartozó feladatokat listázza ki, viszont ekkor már, mindegyik felhasználóhoz tartozó feladatot látjuk.

A feladatoknál, a szokásos műveleteken – új, részletek, szerkesztés, törlés – kívül, van még egy művelet, amely az adott feladat teljesítése. Létrehozáskor mindegyik feladat alapról teljesítendő besorolást kap, így tehát ezeket a feladatokat még teljesítenünk kell. Amint valamelyik feladattal elvégeztünk, a „Teljesítés” gombra kattintva, a feladathoz tartozó státusz mezőben lévő szöveg megváltozik, és felveszi a befejezett értéket. Ekkor a gombunk szövege és színe is megváltozik. A gomb újbóli megnyomása



esetén, újra aktívvá tudjuk tenni a feladatunkat. Így folyamatosan felügyelni tudjuk, hogy az adott projektből, mennyi feladatunk van még hátra. Az adott feladat státuszát, csakis a feladathoz rendelt alkalmazott tudja megváltoztatni.



17. ábra: A „Feladatok” oldal

Csakúgy, mint a feladatok megtekintését, a feladatok létrehozását is két helyről tudjuk elérni. Elsőként a „Feladataim” menüpontnál az „Új hozzáadása” gombra kattintva. Ekkor a feladat neve mellett, egy legördülő listából ki kell választanunk a megfelelő projektet, amihez a feladatot létre akarjuk hozni. A másik opció, hogy az árajánlat fázisban és az értékesítési fázisban lévő projekteknel, az adott projekt sorában lévő „Feladat hozzáadása” gombra kattintunk. Ekkor már csakis a feladat nevét kell kitöltenünk, ugyanis a hozzá tartozó projekt alapból ki lesz választva a legördülő listában. Ezekon kívül pedig, meg kell határoznunk a feladat végrehajtóját. Ez alapból a bejelentkezett felhasználó, viszont bármely munkatársunkhoz tudjuk delegálni, hozzárendelni az adott feladatot.

#### 2.3.9.1. Kampányok

Az ügyféllel való kapcsolattartás legcélravezetőbb formája a kampányok. Különböző kampányok segítségével, akár a meglévő, akár a potenciális ügyfeleink számára tudunk hírleveleket küldeni. Az e-mail az egyik legköltséghatékonyabb, ám emellett az egyik legjobb módja annak, hogy elérjük az ügyfeleinket, különböző ajánlatokkal keressük fel őket. A kampányokat két helyről tudjuk elérni a programban. Elsőként a „Kampányok” menüpont alatt, ekkor az összes kampányunkat láthatjuk (18. ábra), másrészt pedig a

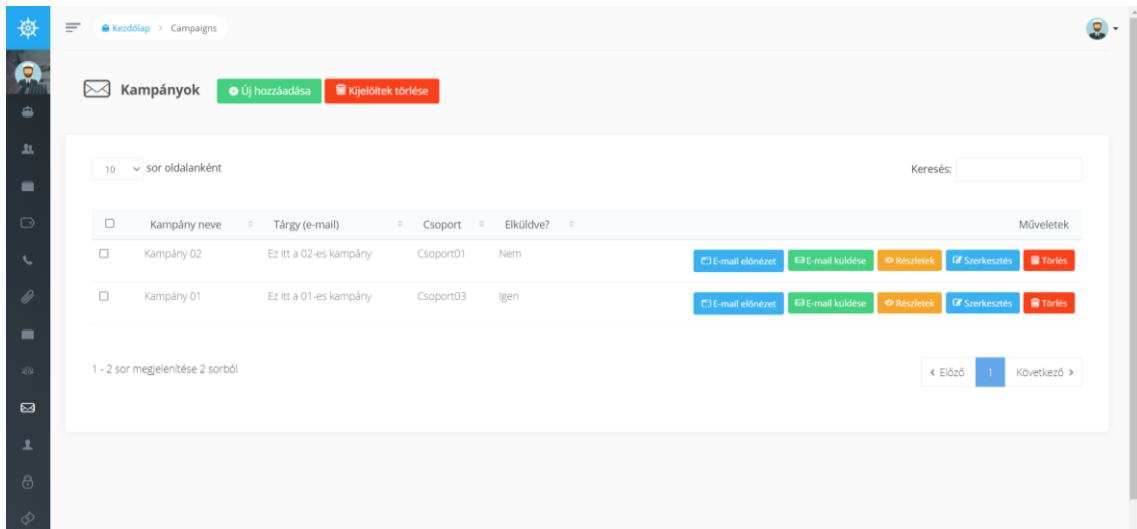
„Csoportok” menüpontból, a „Kampányok” gombra kattintva. Ekkor egy szűrt listát láthatunk, amely az adott ügyfélcsoportoz tartozó kampányokat listázza ki.

A programban a kampányok készítése két lépésből áll. Először is, meg kell határoznunk, azokat az ügyfeleket, akiknek szeretnénk kampányt készíteni. Erre szolgál a korábban már bemutatott ügyfélcsoportok. Tehát hírlevelet, ügyfélcsoportoknak tudunk küldeni. Ehhez vagy felhasználjuk a már korábbiakban létrehozott csoportjaink egyikét, vagy ha új célból akarunk e-mailt küldeni, akkor hozhatunk létre új csoportokat is.

Második lépésként, ha már megvan a kész ügyfélcsoportunk, akkor a kampányok oldalon az „Új hozzáadása” gombra kattintva hozhatjuk létre az adott kampányt. Ekkor meg kell adnunk a **kampány nevét**. Itt érdemes valami beszédes, hangzatos nevet adnunk. Ezt a nevet csakis mi fogjuk látni, ez nem fog kikerülni az ügyfelekhez. Így mindenképpen olyan nevet válasszunk, ami alapján egyértelműen be tudjuk azonosítani, hogy jelen esetben melyik kampányról is van szó. A következő adat, amit meg kell adnunk, az a kiküldendő e-mail-nek a **tárgya**, majd magának az e-mailnek a **szövegét** adhatjuk meg. Ezt egy markdown editor<sup>3</sup> segítségével tudjuk megtenni. Ennek köszönhetően nem csak egyszerű, hanem formázott szöveget tudunk küldeni, felsorolásokkal, címsorokkal, szövegkiemelésekkel vagy akár képekkel is. Minden e-mailben a fogadó fél, automatikusan meglesz szólítva, így ezzel nekünk nem kell törődnünk külön, ezt a program elvégzi helyettünk, az ügyfél neve alapján. Végezetül az utolsó rész, a megfelelő **csoport kiválasztása**. Ezt egy legördülő menü segítségével tudjuk megtenni, ahol láthatjuk az ügyfél csoportjainkat.

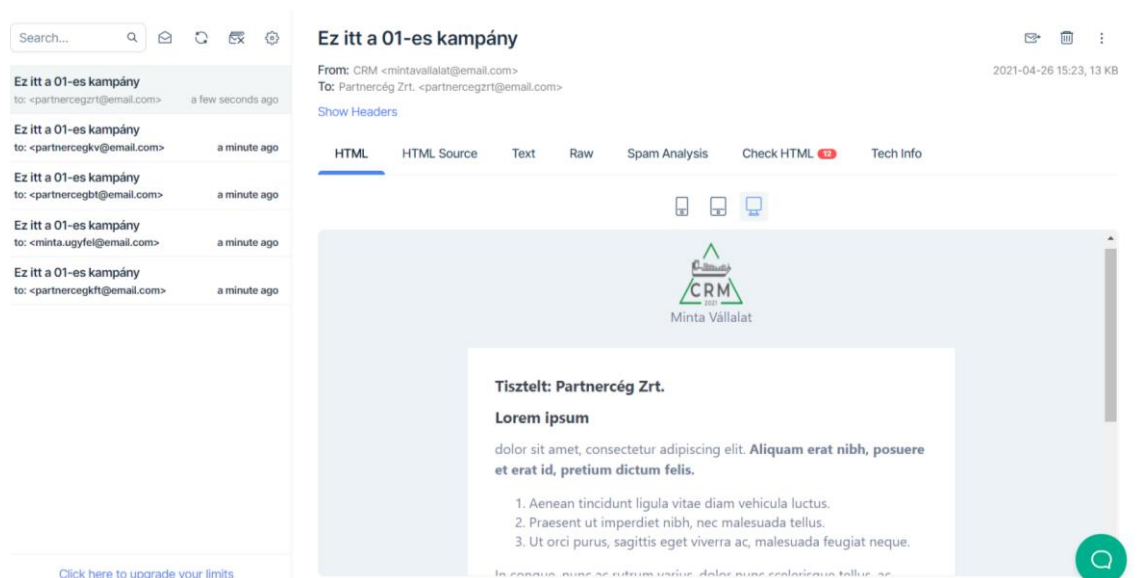
---

<sup>3</sup> Egy WYSIWYG (What You See Is What You Get – amit látsz, azt kapod) szerkesztő.



18. ábra: A „Kampányok” oldal

Létrehozás után megjelenik a kampányunk a listában. Az „Elküldve?” oszlopban létrehozás után alapértelmezetten a „Nem” szócska jelenik meg. Ez minden újonnan létrehozott kampány esetén ezt az értéket fogja felvenni. Küldés előtt, meg tudjuk nézni az e-mailünket. Ehhez az „E-mail előnézet” gombra kell kattintanunk. Le tudjuk ezáltal ellenőrizni, hogy nem vétettünk-e valamilyen hibát az e-mail létrehozása során. Majd ezt követően, ha el akarjuk küldeni a hírlevelet, akkor az adott kampánynál lévő „E-mail küldése” gombra kell kattintanunk. Ekkor az összes ügyfél e-mail címére ki lesz küldve az adott e-mail, akik tagjai a kijelölt ügyfél csoportunknak (19. ábra). A küldést követően az „Elküldve?” oszlopban, már az „Igen” szó jelenik meg, így figyelmeztetve minket, hogy ezt az e-mail-t már egyszer elküldtük.



19. ábra: A kiküldött e-mail-ek listája

#### 2.3.10. Valuták

A „*Valuták*” menüpont, csak a menedzser jogosultsággal rendelkező felhasználóknak érhető el. A projektek létrehozása során kell egy legördülő lista segítségével kell kiválasztanunk a munkához rendelt valutát. Ezeket itt tudjuk megadni. Az egyes valutákat tudjuk szerkeszteni és törölni, az „*Új hozzáadása*” gombra kattintva pedig, új valutát tudunk felvenni. Ekkor a teljes nevét és a 3 betűs rövidítését<sup>4</sup> kell megadnunk.

#### 2.3.11. Alkalmazottak

Ők a szoftverfejlesztő cég alkalmazottjai. Az „*Alkalmazottak*” menüpontból érhetőek el. Táblázatos nézetben látható a cég összes alkalmazottja, a nevük, e-mail címük, profilképük, illetve a szerepkörük. A normál felhasználók láthatják a munkatársaik részletesebb adatait, valamint tudják szerkeszteni a saját profiljukat. Míg a menedzserek szerkeszteni és törölni is tudják az egyes felhasználókat. Valamint, csak a menedzserek tudnak új alkalmazottat felvenni a rendszerbe. Az „*Új hozzáadása*” gombra kattintva, meg kell adni az alkalmazott nevét, e-mail címét, jelszavát – amit az első bejelentkezés után érdemes a felhasználónak megváltoztatnia – szerepkörét, illetve tetszőlegesen egy profilképet is lehet az egyes alkalmazottakhoz hozzárendelni.

#### 2.3.12. Jogosultságok

A „*Jogosultságok*” menüpont csak a menedzserek számára érhető el. Itt tudják a meglévő felhasználói szerepköröket módosítani, törölni, valamint új szerepkört létrehozni. Új létrehozásánál meg kell adni a nevét, illetve kiválasztani, azokat a lehetőségeket, amikkel az új felhasználó rendelkezhet.

### 2.4. Az ügyfélkapcsolat kezelés folyamatának bemutatása

A program teljeskörűen lefedi az ügyfélkapcsolat kezelés főbb feladatait. Az ügyfélkapcsolat kezelés egy többlépcsős, bonyolult folyamat, amelynek célja a cég sikeressége, a profit növelése, az ügyfelek megtartása, új, potenciális ügyfelek felkutatása, illetve az alkalmazottak munkájának, az adminisztrációs folyamatoknak a megkönnyítése.

---

<sup>4</sup> Az ISO 4217-es nemzetközi szabvány tartalmazza a valuták jelölésének 3-karakteres kódját.

Középpontban az ügyfelek állnak. Kétféle ügyfelet különíthetünk el: a meglévőket, azokat a partnereinket, akik éppen kapcsolatban állnak a céggel, van aktuális projektjük, vagy egy volt partnerünk, aki valamikor már és felkereste a cégünket. A másik csoport pedig a potenciális vevők. Ők azok, akiket valamilyen másik kommunikációs csatornán, például egy blogon, közösségi médiában találtunk meg és kampányok, hírlevelek formájában keressük őket.

#### 2.4.1. Meglévő aktuális és múltbéli ügyfelek menedzselése

Amikor egy ügyfél felkeresi a cégünket, egy konkrét ötlettel, megvalósítandó projekttel kapcsolatban, őket a meglévő ügyfelek kategóriába tudjuk sorolni. Ide tartoznak azok a vevőink is, akikkel korábban már volt kapcsolatunk, dolgoztunk velük, és újra felkeresik a cégünket, egy újabb sikeres munka reményében. Minden vevőnk az „Ügyfelek” menüpont alatt található, itt a kereső mezőben, akár név, e-mail cím vagy telefonszám alapján tudunk rájuk keresni. Ha a program találatot jelezz, akkor tudhatjuk, hogy valamilyen okból már kapcsolatban álltak a cégünkkel. Láthatjuk részletesebben az adataikat, megnézhetjük, hogy mik voltak a korábbi közös munkák, mit kell tudni az adott partnerről. Ha nincs találat, akkor egy teljesen új ügyfélről van szó, így őt rögzítenünk kell a rendszerbe.

Második lépésként mindenképpen érdemes a kapcsolattartó személyét rögzíteni. Ő az, akivel a munka folyamán kommunikálni fogunk, őt fogjuk különböző esetekben keresni. Fontos, hogy ne csak a céget ismerjük, hanem a kapcsolattartót is. A bizalom építés egyik fontos lépése, hogy ismerjük azt, akivel napi, heti rendszerességgel kommunikálunk. Bejövő hívás, vagy e-mail esetén egyből tudjuk, hogy ki keres minket, hiszen a programból azonnal megtudhatjuk, így név szerint tudjuk megszólítani az illetőt, ami mindenképpen egy pozitív dolog az ügyfél szemében.

Ezek után az ügyfél felvázolja, hogy pontosan miért is keresett fel minket, miért fordult a cégünkhöz. Ezeket a különféle kéréseit, feladatait, projektek formájában tudjuk rögzíteni a programban. Célszerű már az elején minden fontosabb dolgot lejegyzetelni, hogy minél tisztább képet kaphassunk az ügyfél elképzeléseiről. A megbízatás rögzítése után, érdemes hozzá egy-egy feladatot is felvinni, amiket tudunk, hogy egész biztosan el kell majd végeznünk. Általában egy munkánál (projektnél) már előre tudhatunk néhány

feladatot, például a követelmények leírása, a feladat részletes specifikálása, az adatbázis kialakítása, a rendszer fejlesztése, tesztelése, a design kialakítása, beüzemelés, implementálás, fenntartás stb. Természetesen, ahogy a munka halad előre, jönnek majd az újabb és újabb feladatok, de ezekből az előzetesen rögzített lépésekből az ügyfél is tisztább képet kaphat arról, hogy pontosan mit is fogunk mi végrehajtani.

Minden kezdetben létrehozott projekt árajánlat fázisba kerül. Miután a munkának minden fontosabb elemét rögzítettük, felvittük a feladatokat, meghatároztuk a projekt árát, határidejét, abban az esetben minden adott ahhoz, hogy az árajánlatot elküldjük a vevőnek. A program digitális árajánlattal könnyíti meg az alkalmazottak munkáját, hiszen nem kell külön alkalmazást használni, egyszerűen a program PDF formátumban elkészíti az árajánlatot. Így nem is kell feltétlenül kinyomtatni, egyszerűen online formában át tudjuk adni az ügyfelünknek, ezzel amellet, hogy időt és pénzt is spórolunk, még cégünk ökológiai lábnyomát is tudjuk csökkenteni. Az ügyfél itt dönthet, hogy vagy elfogadja az árajánlatot és ezzel megkezdődhet a közös munka, vagy elutasítja azt, és ezzel a jelenlegi munkakapcsolat lezárul. Ettől függően tudjuk a projektet értékesítési fázisba, vagy sikertelenül lezárt fázisba rakni. Fontos, hogy a sikertelen projekteket is rögzítsük mindig, hiszen ez is egyfajta visszajelzést nyújt nekünk.

A munka sikeres folytatása esetén, újabb feladatokat tudunk hozzáadni a projekthez, azokat folyamatosan teljesíteni kell majd. A feladatokat ki tudjuk rendelni a különböző munkatársaknak, ők azonnal látják ezt a saját oldalukon, így mindig tudják, hogy mikor, milyen feladatok kell még elvégezniük. Miután mindegyik projekthez tartozó feladatot teljesítettük és a projekt elkészült, át tudjuk mozgatni sikeresen lezárt fázisba. Ez jelenti a közös munka (remélhetőleg időleges) végét, sikerrel abszolváltunk egy újabb feladatot. Lehetséges természetesen értékesítési fázisból sikertelen projektet is kreálni, illetve van lehetőség egy lezárt projektet újra megnyitni, amennyiben az ügyfél valamivel még nem elégedett, vagy módosítást, kiegészítést akar még.

Minden vállalat célja a sikeres együttműködés a partnerekkel. A folyamat végigvezetése segít nekünk, hogy mindig képben legyünk, hogy az adott projektnél hol tartunk, az ügyfélnek mindig be tudjuk mutatni, hogy most éppen melyik folyamatnál vagyunk.

Mindezek segítenek abban, hogy a meglévő ügyfeleink a lehető legegységesebbek legyenek, és a későbbiekben is, újabb munka esetén a cégünket válasszák.

#### 2.4.2. Potenciális ügyfelek felkeresése

A másik nagy ügyfélcsoport a potenciális ügyfelek. Ők azok, akik nem állnak és nem álltak semmilyen kapcsolatban a cégünkkel, viszont a tevékenységükből, vagy érdeklődési körükből adódóan relevánsak lehetnek a számunkra. Velük kapcsolatban, hosszú távon mindenképpen azt szeretnénk elérni, hogy ha valamikor szükségük lesz valamilyen informatikai programra, megoldásra, akkor mindenképpen a mi cégünket válasszák. Ezért kulcsfontosságú, hogy folyamatosan építsük ki velük a kapcsolatot. Ismerjék meg a cégünket, tudjanak rólunk, domborítsuk ki, hogy miben vagyunk jók, miért minket válasszanak más cégek helyett.

A potenciális ügyfelek felkutatásának több lehetősége is van. Célszerű például azokban a témákban, melyekkel foglalkozik a cégünk, vagy akár mi magunk – webes technológiák, webáruházak üzemeltetése, webdesign – különböző weboldalakat, blogokat üzemeltetni, ahol heti, havi rendszerességgel írunk különböző bejegyzéseket. A weboldalakon különböző hírlevelekre is fel lehetne iratkozni, melyekkel amellet, hogy magáról a bejegyzésekről is értesítést kap az adott személy, egyúttal a saját cégünk promóciós hirdetéseivel is meg tudjuk őt keresni. Természetesen a GDPR<sup>5</sup> szabályok betartása mellett. Idővel pedig, hogy újabb és újabb témákról blogolunk, egyre többen iratkozhatnak fel a különböző hírlevelekre, hozzászólásokat írhatnak a bejegyzések alá, ezáltal is növelve a potenciális ügyfelek listáját. Őket folyamatosan rögzítjük egyben a rendszerben is, különböző ügyfélcsoportokhoz rendeljük őket, hogy átláthassuk, hogy az adott vevőt melyik témák érdekelhetik.

Természetesen ezekben az ügyfélcsoportokba, nem csak a potenciális vevőink kerülhetnek, hanem a már meglévő partnereink is. Fontos, hogy a későbbiekben őket se hanyagoljuk el, hiszen nekik már van egy képük a cégünkkel kapcsolatban, így – amennyiben az előző munkakapcsolat sikeres volt – ők nagyobb valószínűséggel fognak minket választani egy újabb munka esetén.

---

<sup>5</sup> GDPR (General Data Protection Regulation) – általános adatvédelmi rendelet, az Európai Parlament és a Tanács (EU) 2016/679 rendelete.

Majd amikor a cégünk például webes technológiák témakörben kitalál valamilyen új, innovatív megoldást, akkor a webes technológiák ügyfélcsoportba tartozó ügyfeleinket meg tudjuk keresni, a programba implementált kampányok modul segítségével. Itt meg tudjuk adni a kiküldendő e-mailnek a tárgyát, szövegét. A tárgy, talán az egyik legfontosabb rész, ugyanis a küldő neve mellett, ami a cégünk neve, a meglévő és potenciális vevők, az e-mail tárgy részével találkoznak először. Ez alapján döntenek el, hogy rákattintanak-e az e-mailre, hogy elolvassák-e a teljes szöveget, vagyis érdemben foglalkoznak-e azzal a levéllel, amivel megkerestük őket. Tehát kulcsfontosságú, hogy megfelelő, rövid, tömör, de ugyanakkor figyelemfelkeltő, érdekes szöveget válasszunk tárgynak. Az e-mail szövege is rendkívül fontos. Ne írjunk regény hosszúságú szöveget, de azért tartalmasan magyarázzuk el, hogy miért is kerestük meg az ügyfelünket.

Így folyamatosan időről időre tudunk a vevőinknek – legyen szó meglévő, vagy potenciális vevőről – különféle ajánlatokat küldeni. Ezáltal biztosítva azt, hogy folyamatosan lesznek érdeklődők, akik felkeresik a cégünket, megbíznak minket munkákkal. Hiszen egy vállalatnak nem csak a jelenre kell koncentrálnia, hanem mindig gondolnia kell a jövőre is, erre pedig a potenciális ügyfelek felkutatása az egyik legjobb módszer.



### 3. Fejlesztői dokumentáció

Az alábbi fejezetben bemutatom a program elkészítésének főbb lépéseit. Bemutatásra kerülnek azok a webes technológiák, felhasznált módszerek, amelyek segítségével megalkottam a rendszert. Ismertetem a programom fejlesztői felületét, adatbázistervét, és a nagyobb modulok elkészülését. A fejezetben részletezem továbbá, hogy hogyan lehet egyszerűen tovább bővíteni a program funkcionalitásait, amely segítségével teljesen a felhasználó igényeire lehet szabni az alkalmazást.

#### 3.1. A probléma részletes specifikációja

Az alkalmazás egy szoftverfejlesztő cég mindennapi munkáját hivatott megkönnyíteni. Elsősorban az ügyfelekkel való kapcsolattartást, az ügyfelekhez tartozó feladatok, projektek napi menedzselését, nyomon követését, és a későbbiekben a kapcsolattartást segíti. A program egyik fő szempontja a mobilitás, hogy szinte bárhol és bármikor elérhető legyen, ne korlátozódjon le a használata egy számítógépre vagy egy irodára. Többek között ezen okoknál fogva a program, egy web alkalmazás keretében készült el.

#### 3.2. Felhasznált módszerek részletes leírása, a használt fogalmak definíciója

A rendszer PHP nyelven íródott, Laravel keretrendszer segítségével, azon belül is a Voyager kiegészítővel, MySQL adatbázist használva. A Voyager-nek köszönhetően a frontend<sup>6</sup>, nagy részben adott volt, ehhez külön designt nem terveztem, felhasználtam az előre definiált sablon oldalakat. Az alkalmazás követi az MVC (model-view-controller), azaz a modell-nézet-vezérlés tervezési mintát.

##### 3.2.1. PHP

A PHP egy szerveroldali szkriptnyelv. Dinamikus weboldalak elkészítésére használjuk. Széles körben elterjedt, alapvetően weboldalak létrehozására használják. A PHP a Personal Home Page (személyes honlap) kifejezésből ered, de hivatalos jelentése a következő: Hypertext Preprocessor [2].

---

<sup>6</sup> A felhasználóval kapcsolatot tartó réteg neve, feladata az adatok megjelenítése, a bejövő adatok befogadása

### 3.2.2. Laravel

A Laravel az egyik légnépszerűbb PHP keretrendszer. Ingyenes, nyílt forráskódú rendszer, amely az MVC, azaz a modell-nézet-vezérlő programtervezési mintát követi. A Laravel jellemzői között van, hogy rendkívül könnyen kezelhető, bővíthető, fejleszthető és gyorsan lehet vele dolgozni. Megkönnyíti az általános feladatok elvégzését [3].

### 3.2.3. MVC

Az MVC (model-view-controller), modell-nézet-vezérlő egy programtervezési minta. Az MVC minden alkalmazást három egységre bont. A megjelenésért, az adatok kezeléséért és a felhasználó utasításaiért is külön egység felel [4].

- Model (modell): Ez a réteg kezeli az adatokat, felelős azok tárolásáért. A modellnek a része az adatbázis is.
- View (nézet): Ez a réteg biztosítja a felhasználói felületet. Az adatok megfelelő formában történő megjelenése is itt megy végbe.
- Controller (vezérlő): Ez a réteg hívja meg a modell egyes függvényeit, majd az onnan kinyert adatokat, visszajuttatja a nézetnek.

### 3.2.4. Voyager

A Voyager egy admin csomag a Laravel-hez, amely tartalmazza a BREAD – browse, read, edit, add, delete – műveleteket, médiakezelőt, menüépítőt. Segítségével könnyen tudunk az alkalmazásunkhoz adatokat adni, szerkeszteni, törölni [5].

### 3.2.5. MySQL

SQL-alapú többszálú, relációs adatbázis-kezelő szerver. Teljesen nyílt forráskódú LAMP (Linux-Apache-MySQL-PHP) összeállítás része. Grafikus felületű, nyílt forrású kezelőfelülete a phpMyAdmin program [6].

## 3.3. Program logikai és fizikai szerkezetének leírása

### 3.3.1. Adatkezelési műveletek, BREAD-ek

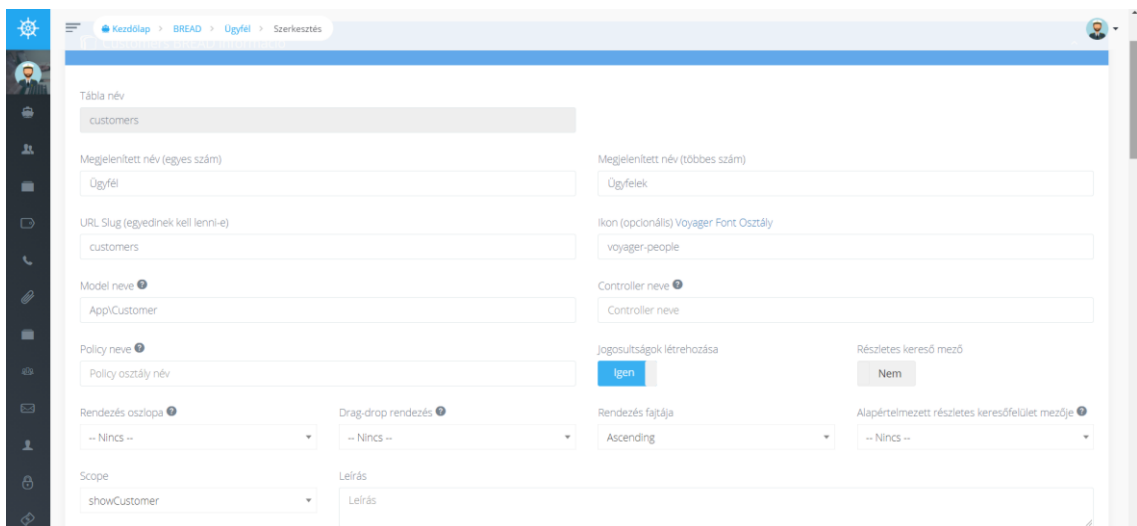
A Voyagerben az alapvető adatkezelési műveletek az úgynevezett BREAD-ek. A BREAD egy mozaikszó, mely az angol *browse*, *read*, *edit*, *add* és *delete* szavakból áll össze, vagyis a böngészés, olvasás, módosítás, hozzáadás és törlés műveleteket tartalmazza.

### 3.3.1.1. BREAD létrehozása

A Voyagerben BREAD-et, csakis adatbázis táblához tudunk létrehozni. Táblát rendkívül egyszerűen lehet létrehozni, ugyanis biztosítja nekünk a kiegészítő csomag a grafikus felületet az adatbázis menedzseléséhez. Ehhez a programba adminisztrátor módban kell belépnünk, az *admin@admin.com* email címmel és a *password* jelszóval. Ezek után az „Eszközök” menüponton belül az „Adatbázis” almenübe kell belépnünk. Itt az „Új tábla létrehozása” gomb segítségével tudunk új táblát felvenni. Fontos, hogy kapcsoljuk be azt az opciót, hogy a program automatikusan hozzon létre modellt a táblához. Itt meg tudjuk adni a tábla nevét, az egyes oszlopok nevét, típusát, hosszúságát. Miután létrehoztuk a táblát, lépünk át a „BREAD” menübe és vegyünk fel egy új BREAD-et a táblánkhöz. Miután hozzáadtuk, a Voyager automatikusan elkészíti a táblához tartozó sablon nézeteket, alapvető műveleteket, így könnyítve meg a munkánkat.

### 3.3.1.2. BREAD használata

A Voyager nem csak abban segít nekünk, hogy automatikusan elkészíti a megfelelő nézetünket, és a hozzá tartozó alpműveleteket – böngészés, olvasás, szerkesztés, hozzáadás, törlés –, hanem számtalan egyéb lehetőséget is kínál. A BREAD szerkesztése vagy hozzáadása részénél, egy komplett űrlapot kapunk, melyben rengeteg mindent be tudunk állítani az adott táblára, nézetre vonatkozóan (20. ábra).

The screenshot shows the Voyager Admin interface for creating a BREAD. The breadcrumb navigation at the top reads "Kezdőlap > BREAD > Ügyfél > Szerkesztés". The form contains the following fields and options:

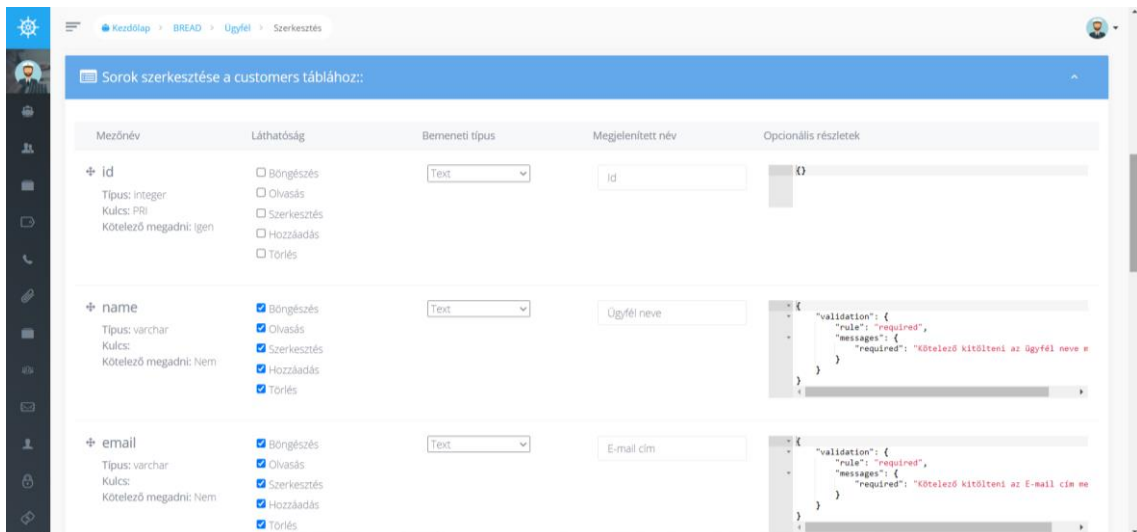
- Tábla név:** Text input with "customers" entered.
- Megjelenített név (egyes szám):** Text input with "Ügyfél" entered.
- Megjelenített név (többes szám):** Text input with "Ügyfelek" entered.
- URL Slug (egyedinek kell lenni-e):** Text input with "customers" entered.
- Ikon (opcionális) Voyager Font Osztály:** Text input with "voyager-people" entered.
- Model neve:** Text input with "App\Customer" entered.
- Controller neve:** Text input with "Controller neve" entered.
- Policy neve:** Text input with "Policy osztály név" entered.
- Jogosultságok létrehozása:** Two buttons, "Igen" (highlighted in blue) and "Nem".
- Rendezés oszlopa:** Dropdown menu with "-- Nincs --" selected.
- Drag-drop rendezés:** Dropdown menu with "-- Nincs --" selected.
- Rendezés fajtája:** Dropdown menu with "Ascending" selected.
- Részletes kereső mező:** Dropdown menu with "-- Nincs --" selected.
- Scope:** Dropdown menu with "showCustomer" selected.
- Leírás:** Text area with "Leírás" entered.

20. ábra: Az Adminisztrátor felhasználóknak elérhető BREAD létrehozása oldal

Megtudjuk adni az adatbázis tábla megjelenített nevét, hogy milyen néven láthassuk az egyes nézetekben, ki tudjuk választani az ikonját, azt a modellt, vezérlőt, amit használjon a nézet. Ezekon kívül különböző kereső mezőket tudunk megadni. Be lehet állítani

részletes kereső mezőt, amelynek segítségével *drag and drop*<sup>7</sup> rendezést tudunk létrehozni. Rendezéseket tudunk beállítani, hogy például melyik rekord szerint legyen rendezve a nézet, illetve szűrést tudunk még beállítani. Szűrést a scope segítségével tudunk létrehozni, amelyek a BREAD böngészési eredményeit szűrik le. Ehhez elsőként a BREAD-nél megadott modellben kell létrehozni egy scope-ot, majd a lenyíló mezőből kell kiválasztani a megfelelő szűrést tartalmazó metódust.

Ezekon kívül pedig az egyes sorokra tudunk még részletes beállításokat elvégezni (21. ábra). Meg tudjuk adni, hogy az öt BREAD műveletből, melyiket lehessen végrehajtani az egyes mezőkön. Ki lehet választani a bementi típust, a megjelenített nevet, illetve az „*Opcionális részletek*” részénél, validációt és alapértelmezett szöveget is be tudunk állítani.

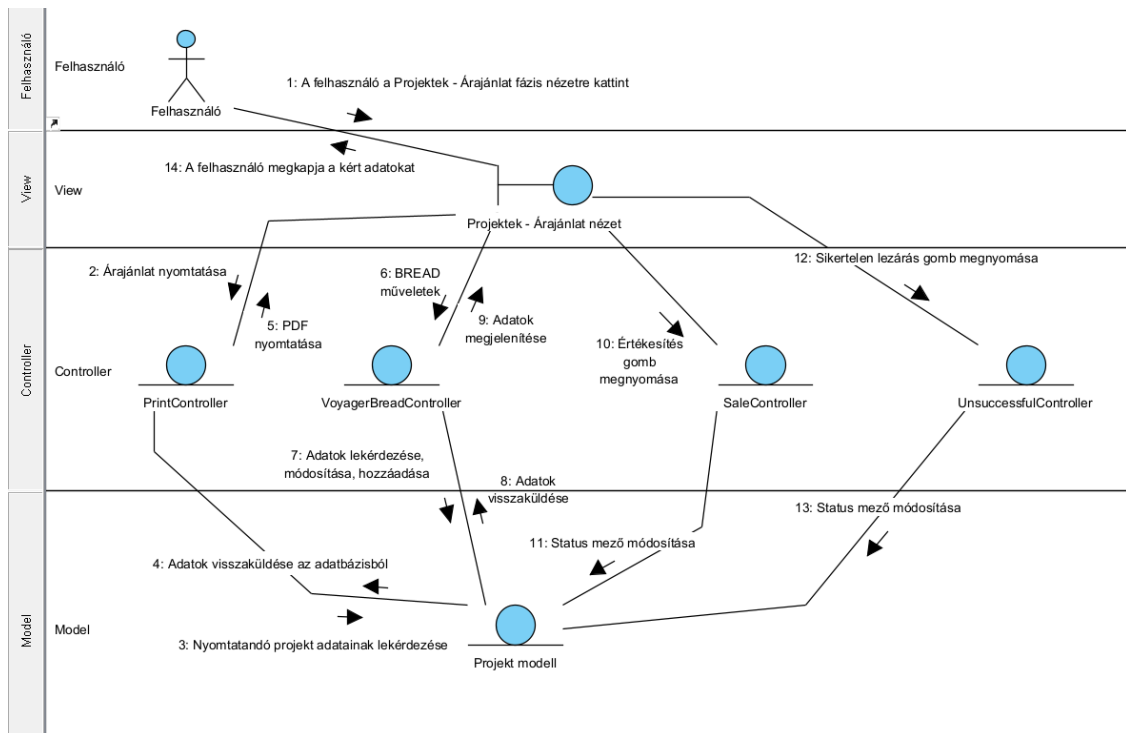


21. ábra: Az Adminisrátor felhasználóknak elérhető BREAD létrehozása oldalon az egyes mezők beállítása

### 3.3.2. MVC tervezési minta megvalósítása

Az MVC tervezési mintának köszönhetően, az alkalmazás három fő rétegre bontható, modellekre, nézetekre és vezérlőkre (22. ábra). Maga a Laravel és a Voyager is megkönnyíti ennek a három rétegnek a kezelését.

<sup>7</sup> Fogd és vidd



22. ábra: Az MVC tervezési minta bemutatása a Projekt modellel és a hozzá kapcsolódó vezérlőkkel, nézettel

### 3.3.3. Modell réteg

A modellek az *app/* mappában találhatóak. Célszerű a programmal automatikusan modellt létrehozni, amikor adatbázis táblát hozunk létre, de ezt manuálisan is meg tudjuk tenni. Mindegyik modell egy-egy adatbázis táblához kapcsolódik. A modell réteg kezeli az adatokat, felelős azok tárolásáért, ez a réteg tartalmazza az üzleti logikát.

#### 3.3.3.1. Campaign modell

A *Campaign* modellben található a *scopeShowCampaign()* nevű függvény. A függvény hozza létre azt a lekérdezést, amelynek segítségével a „Csoportok” menüpontban lévő „Kampányok” gombra kattintva, le tudjuk szűrni az adott csoporthoz tartozó kampányokat. A gomb lenyomásának hatására a függvény megkapja az adott csoport egyedi azonosítóját, amit elment egy változóban. Majd az erre vonatkozó lekérdezést elmenti egy változóban és visszaadja. Amennyiben nem érkezett át egyedi azonosító, abban az esetben az illető, a „Kampányok” menüpontról érte el a kampányok oldalt, így ő nem egy szűrt listát szeretne visszakapni, hanem az összes kampányt megtekinteni.

#### 3.3.3.2. Contact modell

A *Contact* modellben a *scopeShowContact()* nevű függvény van. A függvény megvalósítása, szinte teljesen ugyanaz, mint a *Campaign* modellben található

`scopeShowCampaign()` függvény. Itt viszont, az adott ügyfélhez tartozó kapcsolattartókat fogja nekünk leszűrni a rendszer. Az „Ügyfelek” oldalon található „Kapcsolattartók” gomb hatására, most az adott ügyfél azonosítója kerül átadásra. A metódus itt is megnézi, hogy üres-e ez a változó, amennyiben igen úgy a teljes listára vonatkozó lekérdezést adja vissza. Viszont, ha a változó értéke nem egyenlő a *null* értékkel, akkor az adott ügyfélre való szűrést adja vissza a függvény.

#### 3.3.3.3. *ContactCustomer* modell

Ebben a modellben található függvény, a `getNameAttribute()` függvény. A függvény segítségével tudjuk azt megtenni, hogy amikor új projektet hozunk létre, és kiválasztjuk a projekthez tartozó ügyfelet és kapcsolattartót, akkor azt egyben tudjuk kiválasztani, és egymás után láthatjuk az adott ügyfél és kapcsolattartó nevét.

A függvény a *Contacts* és a *Customers* táblákból kéri le a kapcsolattartó és az ügyfél nevét. *Join* segítségével összekapcsolja a táblákat a *Contact\_customers* táblával és egy *where* feltétellel lekérdezi a *Contact\_customers* táblából az adott egyéni azonosítóval rendelkező sort. Majd az ehhez tartozó ügyfél és kapcsolattartó nevét is. Ezeket végül egy *return* segítségével, egy kötőjellel összekötve adja vissza. A függvény eredménye, a projekt létrehozásánál, az ügyfél-kapcsolattartó kiválasztásánál, a legördülő listában látszik.

#### 3.3.3.4. *Customer* modell

A `scopeShowCustomer()` függvény célja, hogy a „Csoportok” oldalon lévő, „Ügyfelek” gombra kattintva, szűrt listában, az adott csoporthoz tartozó ügyfelek jelenjenek meg. Gombnyomásra itt is átadódik a csoport azonosítója. Amennyiben ez az egyedi azonosító vett fel értéket, szóval a program használója szűrt listát akar visszakapni, abban az esetben a függvény, a *Customer\_groups* táblában keresi meg azokat a sorokat, amelyek az adott csoport azonosítóval rendelkeznek. A `customergroups()` függvény, a két tábla közötti – *Customers* és *Customer\_groups* – kapcsolatot írja le manuálisan. Azért nem elegendő csak a Voyager-ben összekapcsolni ezt a két táblát, mert a függvény visszatérési értékét használja fel a `scopeShowCustomer()` függvény. Ez teszi lehetővé, hogy a visszaadott szűrés majd az ügyfeleket jelenítse meg a *Customers* táblából.

#### 3.3.3.5. *Project modell*

A *Project* modellben négy darab *scope* függvény található.

*scopeProjectBid()*: A függvény teszi lehetővé, hogy az árajánlatban lévő projektek jelenjenek meg. Az „Ügyfelek” oldalon a „Projektek (árajánlat)” gombra kattintva, átadja a függvénynek a megfelelő ügyfél azonosítót. Amennyiben nem érkezik azonosító, úgy a „Projektjeim – árajánlat fázis” oldalon használjuk a szűrést, vagyis csak azokat az árajánlat fázisban lévő projekteket listázza ki a program, melyeknek a projekt menedzsere, a bejelentkezett felhasználó. Amennyiben viszont ez az ügyfél azonosító vesz fel értéket, úgy az adott ügyfélhez minden, árajánlat fázisban lévő projekt ki fog listázódni, függetlenül a projektmenedzser személyétől.

A *scopeProjectSale()* az értékesítési fázisban, a *scopeProjectSuccessful()* függvény a sikeresen lezárt, a *scopeProjectUnsuccessful()* függvény pedig a sikertelenül lezárt projekteket jeleníti meg. Mind a három függvény felépítése, és működése is azonos a *scopeProjectBid()* függvényével, a különbség csupán a szűrési feltételben van.

#### 3.3.3.6. *Task modell*

A *Task* modellben a *scopeShowTask()* függvény található, amely szintén egy szűrési függvény. Ez az adott projekthez tartozó feladatokat szűri le. Itt is számít, hogy van-e átadott projekt azonosító vagy pedig nincs. Amennyiben nincs, úgy az összes feladatot leszűri, az éppen bejelentkezett felhasználóra vonatkozóan. Átadott azonosítónál pedig az adott projekthez tartozó feladatokat szűri le a program.

#### 3.3.4. Nézet réteg

A nézetek feladata a modellek tartalmának a megjelenítése. Bármelyik modellnek a tartalmát meg tudjuk jeleníteni, csupán egy-egy nézetet, view-ot kell készítenünk. Az MVC egyik óriási előnye, hogy egy modellhez, bármennyi nézetet tudunk generálni. A nézeteink a *resources/views* mappában találhatóak. A Voyager egyik legnagyobb előnye viszont az, hogy automatikusan létrehozza a modellhez tartozó nézeteinket. Nekünk csupán a megfelelő adattáblához egy BREAD-et kell létrehoznunk, a program pedig automatikusan generál hozzá egy *browse*, *read*, *edit*, *add* és *delete* nézetet is. Ezeknek a segítségével, az adataink mindig a megfelelő módon tudjuk menedzselni.

### 3.3.5. Vezérlő réteg

A Controller-ek, vagyis a vezérlők feladata, hogy összekössék a modelleket a nézetekkel. Egy esemény bekövetkezésére, például egy gombra való kattintásra, a vezérlő az eseményt lefordítja a modellnek, majd megjeleníti hozzá a megfelelő nézetet. A vezérlőink az *app/Http/Controllers* mappában találhatóak. A Voyager, az alapvető BREAD műveletekre a saját vezérlőjét, a *VoyagerBreadController*-t használja. Ezenkívül viszont, ha valamilyen plusz eseményt szeretnénk létrehozni a programban, annak a vezérlőjét már a megfelelő mappában kell definiálni.

#### 3.3.5.1. *CampaignController* vezérlő

A *CampaignController* vezérlő felelős a hírlevelek megfelelő összeállításáért és küldéséért. Három függvény található benne, egy *send()*, egy *view()* és egy *sendEmailToCustomer()* függvény.

A *send()* függvény gyűjti össze az e-mail adatait, a tartalmát és a fogadó felek neveit és e-mail címeit. A „Kampányok” oldalon lévő „Email küldése” gombra kattintva, a függvény megkapja a kiválasztott kampány azonosítóját. Ezek alapján lekérdezi az adatait – tárgy, tartalom rész –, és a kampányhoz tartozó ügyfélcsoportban lévő ügyfelek azonosítóit és neveit. Ezeket az adatokat különböző változóknak tárolja el a függvény. Ezek után csoportosítja az egyes ügyfeleket, így amennyiben véletlenül az egyik partnerünket ugyanabba az ügyfélcsoportba kétszer is felvettük, a hírlevél hozzá is csak egyszer fog megérkezni. Majd a függvény a *sendEmailToCustomer()* függvényt hívja meg, amellyel megtörténik a tényleges küldés. Végezetül pedig, az elküldve mezőt igen értékűre állítja.

*sendEmailToCustomer()* függvény felelős a tényleges küldésért.

*view()* függvény felépítése rendkívül hasonló a *send()* függvényhez. A függvény az e-mail előnézetéért felelős, lekér minden fontosabb adatot, a végén viszont nem hívja meg a *sendEmailToCustomer()* függvényt, hanem megjeleníti a böngészőben az adott e-mailt.

#### 3.3.5.2. *CompleteController* vezérlő

Ebben a vezérlőben található a *complete()* függvény. A függvény feladata, hogy amikor valamilyen felhasználó teljesíti az egyik feladatát, akkor annak a feladatnak a státusza



megváltozzon. A „*Feladataim*” oldalon a „*Teljesítés*” gombra kattintva, a program átadja az adott feladat azonosítóját, ez alapján a függvény lekérdezi az azonosítóhoz tartozó feladatot és elmenti egy változóban. Majd megvizsgálja, hogy ennek a feladatnak a „*status*” mezőjének mi az értéke. Amennyiben „*UNCOMPLETED*”, vagyis még nem teljesített, abban az esetben az új érték „*COMPLETED*” lesz, tehát az adott feladat elvégzésre került. Ugyanez igaz fordítva is, azaz, ha a feladat már teljesített, akkor gombnyomás hatására újra elvégzendő lesz. Végezetül a függvény elvégzi az adatok mentését is, így a változás rögzül az adatbázisban is.

#### 3.3.5.3. *PrintController* vezérlő

A *PrintController* vezérlőben található *downloadPDF()* függvény végzi el, a digitális árajánlat nyomtatását. Amikor a „*Projektek – Árajánlat fázis*” oldalon rákattintunk az „*Árajánlat nyomtatása*” gombra, akkor a függvénynek átadásra kerül a megfelelő projekt azonosító. A rendszer lekérdezések segítségével, lekérdezi a projekthez tartozó minden adatot, illetve a hozzá tartozó ügyfél, kapcsolattartó adatait, a feladatokat, illetve a projektmenedzser adatait is. Ezek az adatok továbbításra kerülnek a megfelelő nézetnek, az árajánlat sablonjának, végezetül pedig a tényleges nyomtatás is végrehajtásra kerül.

#### 3.3.5.4. *SaleController* vezérlő

A *SaleController()* vezérlőben lévő *sale()* függvény felelős azért, hogy a projektjeink árajánlat fázisból, értékesítési fázisba kerüljenek át. A függvény megkapja a projekt azonosítóját, az alapján lekérdezi az adatait, majd a „*status*” mezőjét „*BID*” értékről, „*SALE*” értékre állítja át. Ezek után menti a változásokat az adatbázisba, illetve a függvény átirányít minket a „*Projektek – Értékesítési fázis*” oldalra.

#### 3.3.5.5. *StornoController* vezérlő

A vezérlőben lévő *storno()* függvény, a sikeresen vagy sikertelenül lezárt projekteket teszi vissza értékesítési fázisba. A függvény megkapja a projekt azonosítóját, annak segítségével, lekérdezi az adott projektet, majd a „*status*” mezőt „*SALE*” értékre állítja, vagyis értékesítési fázisba teszi a projektünket. Az adatok mentése után, a függvény a „*Projektek – Értékesítési fázis*” oldalra visz minket.

#### 3.3.5.6. *SuccessfulController* vezérlő

A *SuccessfulController* vezérlőben található *successful()* függvény a projekteket értékesítési fázisból sikeresen lezárt fázisba teszi. Ez a függvény is megkapja a projekt azonosítóját, a „*Projektek – Értékesítési fázis*” oldalon, a „*Lezárás (sikeres)*” gombra kattintva. Majd az adott projektnek a „*status*” mezőjét változtatja meg „*SUCCESSFUL*” értékre. Végül a „*Projektek – Lezárt (sikeres)*” oldalra irányít minket vissza függvény.

#### 3.3.5.7. *UnsuccessfulController* vezérlő

Az *UnsuccessfulController* vezérlőben található *unsuccessful()* függvény teljesen megegyezik a *SuccessfulController* vezérlőben lévő *successful()* függvénnyel, csupán a „*status*” mezőt most, „*UNSUCCESSFUL*” értékre változtatja meg, majd a „*Projektek – Lezárt (sikertelen)*” oldalra irányít vissza.

#### 3.3.6. Adatszerkezet, adatbázis

Az adatbázis kialakításánál a fő cél az volt, hogy minden szükséges adatot megfelelően tudjunk tárolni, miközben elkerüljük az adatismétlést, redundanciát, így nem kreálunk felesleges adatot.

A program MySQL adatbázist használ, ugyanis alapvető kitétel volt, hogy egyszerre több ember legyen képes, egy időben használni a programot. Erre pedig a MySQL adatbázis bizonyult a legjobbnak.

A táblák létrehozásánál külön hangsúlyt kapott a kapcsolatok megfelelő kialakítása. Így például könnyedén letudjuk kérdezni, az egyes ügyfelekhez tartozó projekteket, az alkalmazottakhoz tartozó feladatokat, anélkül, hogy felesleges mezőket hoztunk volna létre.

### 3.3.6.1. Egyed típusok és attribútumok

Egyed típus: *Customers*

Ez a tábla tartalmazza az ügyfelek adatait. Amelyik ügyfelet felvesszük a rendszerünkbe, ő ebben a táblában lesz eltárolva. Mindegyik ügyfél kap egy ügyféltípust a *Customertypes* táblából. A *Customers* és a *Customertypes* táblák között egy a többhöz kapcsolat van.

2. táblázat: *Customers* tábla definíciója

Attribútumok	
Név és típus	Leírás
Id: integer(10)	egyedi azonosító
name: varchar(255)	az ügyfél neve
email: varchar(255)	az ügyfél e-mail címe
phone: varchar(255)	az ügyfél telefonszáma
address: text	az ügyfél címe
description: text	megjegyzés
created_at: date	felvétel dátuma
updated_at: date	módosítás dátuma
Idegen kulcsok	
Név	Leírás
customertype_id	a <i>Customertypes</i> tábla egy rekordjára hivatkozik

Egyed típus: *Customertypes*

Ebben a táblában van eltárolva az ügyfél típusa. A *Customers* tábla hivatkozik rá.

3. táblázat: *Customertypes* tábla definíciója

Attribútumok	
Név és típus	Leírás
Id: integer(10)	egyedi azonosító
name: varchar(255)	a típus neve, lehet „céges” vagy „magánszemély”

Egyedtípus: *Contacts*

Ez a tábla tartalmazza a kapcsolattartók adatait.

4. táblázat: *Contacts* tábla definíciója

Attribútumok	
Név és típus	Leírás
Id: integer(10)	egyedi azonosító
name: varchar(255)	a kapcsolattartó neve
email: varchar(255)	a kapcsolattartó e-mail címe
phone: varchar(255)	a kapcsolattartó telefonszáma
address: text	a kapcsolattartó címe
description: text	megjegyzés
created_at: date	felvétel dátuma
updated_at: date	módosítás dátuma

Egyedtípus: *Contact\_customers*

A tábla segítségével kapcsoljuk össze az ügyfeleket és a kapcsolattartókat. Egy a többhöz kapcsolat van a *Contact\_customers* tábla és a *Customers*, valamint a *Contacts* táblák között.

5. táblázat: *Contact\_customers* kapcsolótábla definíciója

Attribútumok	
Név és típus	Leírás
Id: integer(10)	egyedi azonosító
created_at: date	összekapcsolás dátuma
updated_at: date	módosítás dátuma
Idegen kulcsok	
Név	Leírás
customer_id	a <i>Customers</i> tábla egy rekordjára hivatkozik
contact_id	a <i>Contacts</i> tábla egy rekordjára hivatkozik

Egyedtípus: *Groups*

Ebben a táblában vannak eltárolva az egyes ügyfélcsoportok.

6. táblázat: *Groups* tábla definíciója

Attribútumok	
Név és típus	Leírás
Id: integer(10)	egyedi azonosító
name: varchar(255)	a csoport neve
description: text	a csoport leírása
created_at: date	létrehozás dátuma
updated_at: date	módosítás dátuma

Egyedtípus: *Customer\_groups*

A tábla segítségével kapcsoljuk össze az ügyfeleket és a csoportokat. Egy a többhöz kapcsolat van a *Customer\_groups* tábla és a *Customers* valamint a *Groups* táblák között.

7. táblázat: *Customer\_groups* kapcsolótábla definíciója

Attribútumok	
Név és típus	Leírás
Id: integer(10)	egyedi azonosító
created_at: date	összekapcsolás dátuma
updated_at: date	módosítás dátuma
Idegen kulcsok	
Név	Leírás
customer_id	a <i>Customers</i> tábla adott rekordjára hivatkozik
group_id	a <i>Groups</i> tábla adott rekordjára hivatkozik

Egyedtípus: *Campaigns*

A táblában a különféle kampányok vannak eltárolva. A *Campaigns* és a *Groups* tábla között egy a többhöz kapcsolat van.

8. táblázat: *Campaigns* tábla definíciója

Attribútumok	
Név és típus	Leírás
Id: integer(10)	egyedi azonosító
name: varchar(255)	a kampány neve
subject: varchar(255)	a hírlevél tárgya
body: text	a hírlevél szövege
sended: varchar(255)	az e-mail elküldésének státusza
created_at: date	létrehozás dátuma
updated_at: date	módosítás dátuma
Idegen kulcsok	
Név	Leírás
group_id	a <i>Groups</i> tábla adott rekordjára hivatkozik

Egyedtípus: *Currencies*

A táblában tároljuk a különböző valutákat, amelyeket a projekteknél használunk.

9. táblázat: *Currencies* tábla definíciója

Attribútumok	
Név és típus	Leírás
Id: integer(10)	egyedi azonosító
name: varchar(255)	a valuta teljes neve
currency: varchar(3)	a valuta három betűs rövidítése
created_at: date	létrehozás dátuma
updated_at: date	módosítás dátuma

Egyedtípus: *Projects*

A projektek szerepelnek ebben a táblában. A *Projects* tábla négy táblával is össze van kötve, mindegyikkel egy a többhöz kapcsolattal. Kapcsolat van a *Projects* és a *Contact\_customers* tábla, a *Projects* és a *Currencies* tábla, a *Projects* és a *Groups* tábla, valamint a *Projects* és a *Users* tábla között is.

10. táblázat: *Projects* tábla definíciója

Attribútumok	
Név és típus	Leírás
Id: integer(10)	egyedi azonosító
name: varchar(255)	a projekt megnevezése
deadline: date	a projekt határideje
price: double(10)	munkaóra ára
working_hour: integer(10)	munkaórák száma
status: varchar(255)	a projekt státusza (árajánlat fázis, értékesítési fázis, sikeresen lezárt, sikertelenül lezárt)
description: text	megjegyzés
created_at: date	létrehozás dátuma
updated_at: date	módosítás dátuma
Idegen kulcsok	
Név	Leírás
contactcustomer_id	a <i>Contact_customers</i> tábla egy adott rekordjára hivatkozik
currency_id	a <i>Currencies</i> tábla egy adott rekordjára hivatkozik
group_id	a <i>Groups</i> tábla egy adott rekordjára hivatkozik
user_id	a <i>Users</i> tábla egy adott rekordjára hivatkozik

Egyedtípus: *Tasks*

A táblában a projektekhez tartozó feladatok kerülnek eltárolásra. Egy a többhöz kapcsolat van a *Tasks* és a *Projects*, valamint a *Tasks* és a *Users* táblák között.

11. táblázat: *Tasks* tábla definíciója

Attribútumok	
Név és típus	Leírás
Id: integer(10)	egyedi azonosító
name: varchar(255)	a feladat megnevezése
status: varchar(255)	a feladat státusza (teljesítendő, befejezett)
created_at: date	létrehozás dátuma
updated_at: date	módosítás dátuma
Idegen kulcsok	
Név	Leírás
project_id	a <i>Projects</i> tábla egy rekordjára hivatkozik
user_id	a <i>Users</i> tábla egy rekordjára hivatkozik

Egyedtípus: *Users*

Ez a tábla tartalmazza a vállalat alkalmazottjait. Egy a többhöz kapcsolattal van összekapcsolva a *Users* és a *Roles* tábla

12. táblázat: *Users* tábla definíciója

Attribútumok	
Név és típus	Leírás
Id: integer(10)	egyedi azonosító
name: varchar(255)	az alkalmazott neve
email: varchar(255)	az alkalmazott email címe
avatar: varchar(255)	az alkalmazott profilképe
created_at: date	felvétel dátuma
updated_at: date	módosítás dátuma
Idegen kulcsok	
Név	Leírás
role_id	a <i>Roles</i> tábla egy adott rekordjára hivatkozik

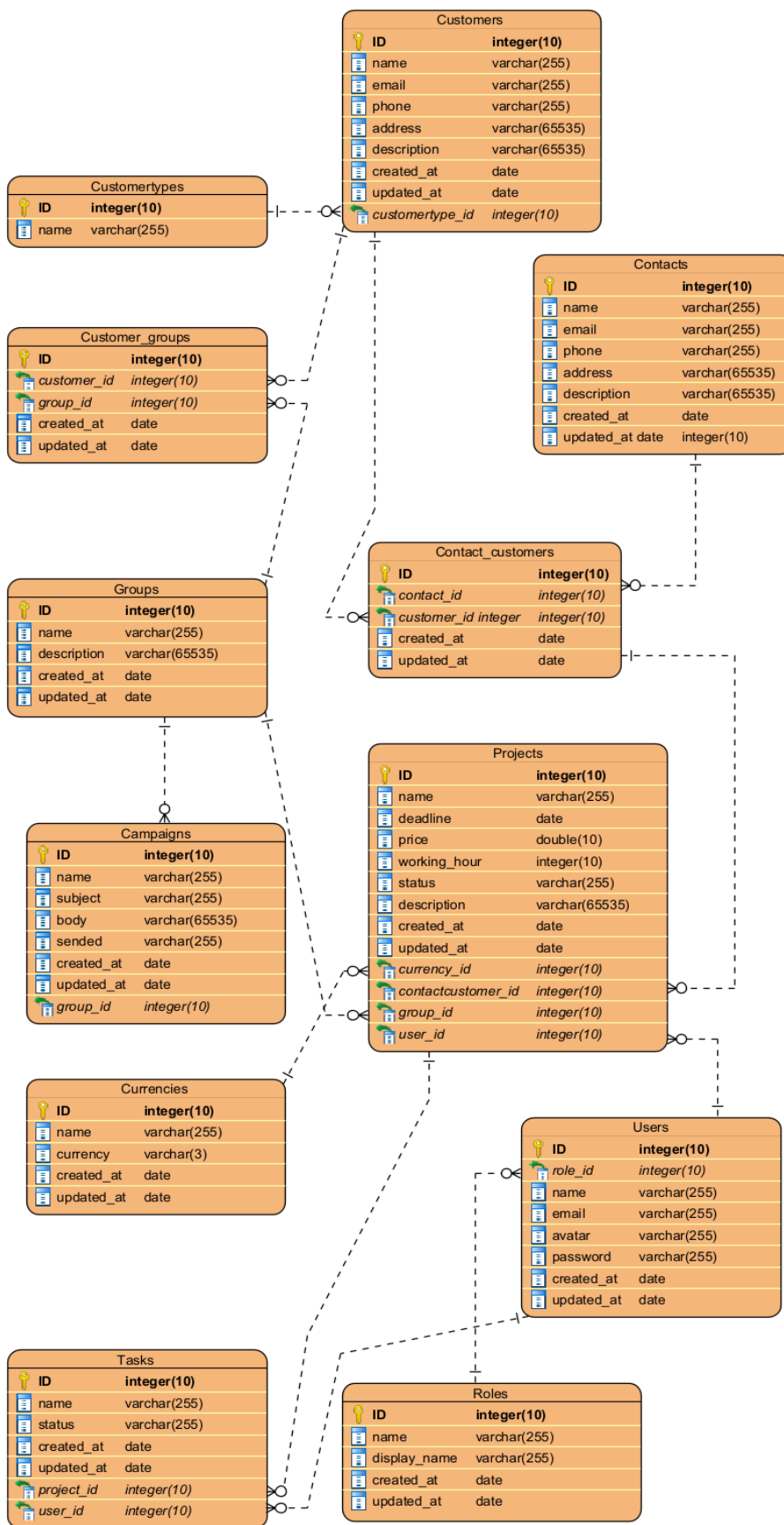


Egyedtípus: *Roles*

Ez a tábla tartalmazza az egyes alkalmazotti jogosultságokat.

13. táblázat: *Roles* tábla definíciója

<b>Attribútumok</b>	
<b>Név és típus</b>	<b>Leírás</b>
Id: integer(10)	egyedi azonosító
name: varchar(255)	a jogosultság neve
display_name: varchar(255)	megjelenített név
created_at: date	létrehozás dátuma
updated_at: date	módosítás dátuma



23. ábra: A program adatbázis tervezete

### 3.3.7. Programcsomagok, modulok

#### 3.3.7.1. Árajánlat nyomtatása

A Laravel egyik nagy előnye, hogy egyszerűen tudunk újabb csomagokat feltelepíteni, amelyeket használni szeretnénk a programunkban. Alapból a Laravel nem képes PDF nyomtatásra, viszont számtalan olyan kiegészítő csomagot lehet hozzá telepíteni, amelyekkel már könnyedén tudunk PDF fájlokat létrehozni. A program elkészítése során, a *Dompdf* programcsomagot használtam, hogy tudjak PDF fájlokat készíteni. A *Dompdf* lényegében egy HTML fájlból hoz létre egy PDF fájlt. Telepítése rendkívül egyszerű. Composer segítségével, parancssorban pár másodperc alatt fel tudjuk telepíteni. A „*composer require barryvdh/laravel-dompdf*” parancsot kell hozzá futtatni. Ezt követően még, be kell jegyeznünk a csomagot a Laravel alkalmazásunkba, és már használhatjuk is.

A programban a digitális árajánlat nyomtatása történik PDF fájlba. A *PrintController* vezérlőben lévő *downloadPDF()* függvény felelős a nyomtatásért. Amikor a „*Projektek – Árajánlat fázis*” oldalon rákattintunk az „*Árajánlat nyomtatása*” gombra, akkor hívódik meg ez a függvény. Ahhoz, hogy a nyomtatás valóban létrejöjjön, még két dolgot kellett előtte megcsinálni.

Elsőként létre kellett hozni egy útvonalat. Ez a *routes\web.php* fájlban található. Amikor rákattintunk a gombra, akkor meghívódik ez az útvonal, és a gombnyomásra kapott adatokat – ami jelen esetben az adott projekt elsődleges azonosítója –, az útvonal átküldi a *PrintController* vezérlő *downloadPDF()* függvényének.

Második lépésként pedig meg kellett határozni azt a nézetet, amit szeretnénk kinyomtatni. Ez a *resources\views* mappában található *pdf.blade.php* fájl. Ennek a HTML oldalnak küldi át a *downloadPDF()* függvény a megfelelő adatokat. Habár az oldal kinézete állandó, a benne lévő adatok – mint például az ügyfél neve, a projekt ára, határideje – dinamikusan töltődnek fel. A függvény betölti a nézetet, és végül elvégzi a nyomtatást.

### 3.3.7.2. E-mail küldése

E-mail küldése a Laravel alkalmazásokban rendkívül egyszerű. A Laravel ezt egy e-mail API<sup>8</sup>-n keresztül valósítja meg, a SwiftMailer<sup>9</sup> könyvtár segítségével. Ennek a megvalósítását tartalmazza a Laravel, így külön csomagot most nem kellett telepíteni. A program ezt egy előre elkészített sablon segítségével oldja meg, úgynevezett Markdown-nal. Ez teszi lehetővé, hogy HTML sablonokat képes a Laravel megjeleníteni, így modern, letisztult e-maileket tudunk küldeni. Minden ilyen markdown két fájlból áll, egy nézetből és egy osztályból.

A nézet fájl a *resources\views\emails* mappán belül található. Itt tudjuk megadni, a küldendő e-mailnek a felépítését, hogy melyik szöveg hova kerüljön, milyen komponensekből álljon az e-mail. A Laravel lehetővé teszi számtalan komponens használatát. Az e-mail pontos szövege természetesen dinamikusan töltődik majd fel, itt csupán egy sablont tudunk létrehozni.

Az osztályunk az *app\Mail* mappában található *CampaignMail.php* fájl. Itt tudjuk megadni, hogy melyik nézetet használja az e-mail, mi legyen a tárgya, a szövege.

A tényleges küldésért a *CampaignController* vezérlőben található *sendEmailToCustomer()* függvény felelős. A függvény megkap három bemenő paramétert, a fogadó ügyfél azonosítóját, az e-mail tartalmát és tárgyát. A metódus először a kapott ügyfél azonosítót használva lekérdezi az adott ügyfelet, akinek kell küldeni az e-mailt, ezt elmenti egy *customer* változóban. A levél tartalma *contents* változóban van eltárolva, míg a tárgya, a *subject* változóban.

```
Mail::to($customer)->send(new CampaignMail($contents, $subject));
```

Az iménti sor felelős az e-mail tényleges küldéséért. A programsorban a *to* rész mondja meg, hogy kinek kell küldeni, a *send* rész pedig létrehoz egy új *CampaignMail* osztályt, amelynek átadja a *contents* és *subject* változókat, vagyis az e-mail szövegét és tárgyát. Ezzel pedig megtörténik az e-mail küldése.

---

<sup>8</sup> API – Application Programming Interface (alkalmazásprogramozási felület)

<sup>9</sup> Egy komponens alapú könyvtár e-mailek küldéséhez a PHP alkalmazásokban.

### 3.3.7.3. Voyager gombok létrehozása

A Voyager több beépített gombbal is rendelkezik. Ilyenek például a „Részletek”, „Szerkesztés”, „Törlés” vagy az „Új hozzáadása” gombok. Ezekon kívül viszont rendkívül egyszerűen tudunk saját gombokat létrehozni. Ehhez az `app\Actions` mappában kell egy új Action osztályt létrehozni (például `PrintAction.php`).

Ebben az action osztályban több dolgot is meg tudunk adni a gombbal kapcsolatban.

- A `getTitle()` függvénnyel tudjuk megadni a gomb feliratát.
- A `getIcon()` függvénnyel lehet kiválasztani a gomb ikonját. Az ikonok listája előre van definiálva a Voyagerben, nekünk onnan kell kiválasztani a megfelelő ikont.
- A `getAttributes()` függvénnyel lehet a gomb stílusát megadni.
- A `shouldActionDisplayOnDataType()` függvénnyel azt az URL-t, vagyis oldalt kell kiválasztani, ahol szeretnénk a gombot használni.
- A `getDefaultRoute()` függvénnyel pedig egy útvonalra tudunk hivatkozni, ahová szeretnénk a gomb lenyomást követően navigálni. Opcionálisan itt különböző paramétereket is – mint például az egyéni azonosítót – át tudunk adni az útvonalnak a gomb lenyomásával.

Például a `PrintAction` osztály felelős azért, hogy a megfelelő árajánlat nyomtatásra kerüljön. Ennek az osztálynak a `getDefaultRoute()` függvényében az `downloadPDF` útvonalat hívjuk meg, és átadjuk neki az adott projekt egyedi azonosítóját, vagyis az `id` mezőt.

```
return route('projects.downloadPDF', ['id' => $this->data->{$this->data->getKeyName()}]);
```

### 3.4. Tesztelési terv és az eredményei

Az alábbiakban összegzem a tesztelési terveket és azok eredményeit.

14. táblázat: Bejelentkezés tesztelése

<i>Teszt leírása</i>	Bejelentkezési funkció tesztelése		
<i>Előfeltételek</i>	Internet hozzáférés, böngésző használata		
<i>Teszt adatok</i>	e-mail cím: menedzser@email.com / jelszó: password		
<i>Lépés leírása</i>	<i>Várható eredmény</i>	<i>Valós eredmény</i>	<i>Sikeresség</i>
URL cím megadása	Megjelenik a bejelentkező felület	Megjelent a bejelentkező felület	Sikeres tesztelés
E-mail cím és a jelszó beírása	A mezőkbe beíródnak az adatok	Bekerültek a mezőbe az adatok	Sikeres tesztelés
Bejelentkezés gombra kattintás	A program bejelentkezik	A program bejelentkezett	Sikeres tesztelés

15. táblázat: Új ügyfél felvitelének a tesztelése

<i>Teszt leírása</i>	Új ügyfél rögzítése a rendszerben		
<i>Előfeltételek</i>	Internet hozzáférés, böngésző használata		
<i>Teszt adatok</i>	ügyfél neve: Partnercég kv. e-mail cím: partnercegkv@gmail.com telefonszám: 06304445550 lakcím: Sopron, Bocskai út 134, 9400 ügyféltípus: Céges ügyfél		
<i>Lépés leírása</i>	<i>Várható eredmény</i>	<i>Valós eredmény</i>	<i>Sikeresség</i>
Ügyfelek oldalon az Új hozzáadása gombra kattintás	Az új ügyfél hozzáadása oldal jelenik meg	Megjelent az új ügyfél hozzáadása oldal	Sikeres tesztelés
Adatok beírása	A mezőkbe beíródnak az adatok	Bekerültek a mezőbe az adatok	Sikeres tesztelés
Mentés gombra kattintás	A program elmenti az ügyfél adatait	A program elmentette az ügyfél adatait	Sikeres tesztelés

16. táblázat: Feladat projekthez adásának a tesztelése

<i>Teszt leírása</i>	Feladat hozzáadása egy árajánlatban lévő projekthez		
<i>Előfeltételek</i>	Internet hozzáférés, böngésző használata, meglévő projekt árajánlat fázisban		
<i>Teszt adatok</i>	feladat megnevezése: feladat_delta		
<i>Lépés leírása</i>	<i>Várható eredmény</i>	<i>Valós eredmény</i>	<i>Sikeresség</i>
<i>Árajánlat fázis oldalon Feladat hozzáadása gombra kattintás</i>	A feladat felvitele oldal jelenik meg, ahol a <i>projekt</i> és <i>alkalmazott</i> mezők már ki vannak töltve	Megjelent az új feladat felvitele oldal, a <i>projekt</i> és <i>alkalmazott</i> mezők már ki vannak töltve	Sikeres tesztelés
<i>Adat beírása</i>	A mezőbe beíródik a feladat neve	Bekerült a mezőbe a feladat neve	Sikeres tesztelés
<i>Mentés gombra kattintás</i>	A program elmenti a feladatot	A program elmentette a feladatot	Sikeres tesztelés

17. táblázat: Árajánlat nyomtatásának a tesztelése

<i>Teszt leírása</i>	Árajánlat nyomtatása PDF formátumban		
<i>Előfeltételek</i>	Internet hozzáférés, böngésző használata, meglévő projekt árajánlat fázisban		
<i>Lépés leírása</i>	<i>Várható eredmény</i>	<i>Valós eredmény</i>	<i>Sikeresség</i>
<i>Árajánlat fázis oldalon Árajánlat nyomtatása gombra kattintás</i>	PDF formátumban kinyomtatódik a megfelelő árajánlat	A projekthez tartozó árajánlat nyomtatásra került PDF formátumban	Sikeres tesztelés

18. táblázat: Feladat lezárásának a tesztelése

<i>Teszt leírása</i>	Értékesítési fázisban lévő feladatnak a lezárása		
<i>Előfeltételek</i>	Internet hozzáférés, böngésző használata, meglévő projekt értékesítési fázisban		
<i>Lépés leírása</i>	<i>Várható eredmény</i>	<i>Valós eredmény</i>	<i>Sikeresség</i>
<i>Értékesítési fázis oldalon a Lezárás (sikeres) gombra kattintás</i>	A projekt átkerül a sikeresen lezárt projektek közé	A projekt átkerült a sikeresen lezárt projektek közé	Sikeres tesztelés

19. táblázat: Új kampány felvitelének a tesztelése

<i>Teszt leírása</i>	Új kampány rögzítése a rendszerben		
<i>Előfeltételek</i>	Internet hozzáférés, böngésző használata		
<i>Teszt adatok</i>	kampány neve: Kampány 04 tárgy (e-mail): Ez itt a 04-es kampány body (e-mail): E-mail törzsszövege csoport: Csoport02		
<i>Lépés leírása</i>	<i>Várható eredmény</i>	<i>Valós eredmény</i>	<i>Sikeresség</i>
<i>Kampányok oldalon az Új hozzáadása gombra kattintás</i>	Az új kampány hozzáadása oldal jelenik meg	Megjelent az új kampány hozzáadása oldal	Sikeres tesztelés
<i>Adatok beírása</i>	A mezőkbe beíródnak az adatok	Bekerültek a mezőbe az adatok	Sikeres tesztelés
<i>Mentés gombra kattintás</i>	A program elmenti a kampányt	A program elmentette a kampányt	Sikeres tesztelés

20. táblázat: Hírlevél kiküldésének a tesztelése

<i>Teszt leírása</i>	Hírlevél küldése a csoportban lévő ügyfeleknek		
<i>Előfeltételek</i>	Internet hozzáférés, böngésző használata, meglévő kampány		
<i>Lépés leírása</i>	<i>Várható eredmény</i>	<i>Valós eredmény</i>	<i>Sikeresség</i>
<i>Kampányok oldalon az E-mail küldése gombra kattintás</i>	Az e-mail kiküldésre kerüljön a csoportban lévő ügyfeleknek	A csoportban lévő ügyfeleknek megérkezett az e-mail	Sikeres tesztelés



## 4. Összefoglalás és további fejlesztési lehetőségek

### 4.1. Összefoglalás

A feladatom egy szoftverfejlesztő cég CRM, vagyis ügyfélkapcsolat kezelő rendszerének a megalkotása volt. A program minden olyan, fontosabb funkcióval rendelkezik, amelyek egy cég mindennapi munkájához szükségesek lehetnek. Beleértve az ügyfelek kezelését, hozzájuk tartozó projektek menedzselését, a meglévő és potenciális vevőknek célzott kampányok készítését. De a program többek között tud digitális árajánlatot készíteni, az alkalmazottak feladatait megjeleníteni, a projekteket különböző fázisba kategorizálni, jogosultsági szinteket kezelni, új alkalmazottakat rögzíteni.

A program egyik kritériuma volt a hordozhatóság, ezért webes környezetben hoztam létre. Így nemcsak a cég telephelyén, az irodában, de akár otthoni környezetben is használható a program. Megjelenését tekintve a webalkalmazást nem csak asztali számítógépen, de akár táblagépen vagy mobiltelefonon is tökéletesen lehet használni. Használatához csupán csak internet hozzáférésre és egy böngészőre van szükség, amelyekkel a programot teljeskörűen lehet használni.

A Laravel keretrendszernek és a Voyager kiegészítőnek köszönhetően, az alkalmazás könnyen továbbfejleszthető, az MVC programtervezési minta miatt jól elkülönülnek a program egyes részei, így könnyedén lehet módosítani az egyes komponenseket.

Úgy gondolom, hogy sikerült megvalósítanom a kitűzött célt, egy komplex, több funkciós ügyfélkapcsolat-kezelő rendszert hoztam létre, mely tökéletesen lefedi az ilyen programok főbb moduljait, az egyes munkafolyamatok menedzselését megkönnyíti, és a cégek mindennapjai során számos adminisztrációs folyamatot leegyszerűsít.

#### 4.2. Továbbfejlesztési lehetőségek

A program számos olyan funkcióval rendelkezik, ami manapság már elvárás egy CRM rendszertől, rengeteg különféle területet lefed. Ezek mellett természetesen még több olyan funkció is van, amelyekkel tovább lehet fejleszteni a programot.

Az egyik ilyen egy számlázó modul beépítése, amelynek köszönhetően a program nem csak digitális árajánlatot lenne képes nyomtatni, hanem a sikeres munka végén automatikusan egy számlát is készítené az ügyfélnek. Ez egy jóval nagyobb informatikai fejlesztés, hiszen amellett, hogy több funkcióval kellene kiegészíteni az alkalmazást, nagyon fontos a számlázás során, hogy minden hatályos jogszabálynak, a számviteli törvénynek megfeleljen a modul.

Második továbbfejlesztési lehetőség a projektek oldalán a fejlesztése. Elsőként a program jelezhetné, hogy mennyi idő van még hátra a határidőig, mennyi feladat teljesült eddig, az adott projektből. Ezenkívül különböző statisztikák jelenhetnének meg a projektekről. Mennyi sikeres és sikertelen projektünk volt eddig, mennyi munkaóránk van az egyes projekteken, mennyi pénzt kerestünk.

Harmadik továbbfejlesztési lehetőségként, a program a különböző változásokról értesítést küldhetne. Például, amikor egy feladatot egy munkatárshoz rendelünk, akkor a program küldhetne neki egy jelzést, például e-mail formájában, hogy érkezett egy új feladata. Természetesen ugyanez fordítva is működhetne, amikor valamelyik alkalmazott végez egy feladattal, ami egy általunk menedzselt projekthez tartozik, akkor arról mi is kaphatnánk értesítést.

## 5. Táblázatjegyzék

1. táblázat: Előre beállított szerepkörök, felhasználónevek és jelszavaik .....	8
2. táblázat: Customers tábla definíciója .....	39
3. táblázat: Customertypes tábla definíciója .....	39
4. táblázat: Contacts tábla definíciója .....	40
5. táblázat: Contact_customers kapcsolótábla definíciója .....	40
6. táblázat: Groups tábla definíciója .....	41
7. táblázat: Customer_groups kapcsolótábla definíciója .....	41
8. táblázat: Campaigns tábla definíciója .....	42
9. táblázat: Currencies tábla definíciója .....	42
10. táblázat: Projects tábla definíciója .....	43
11. táblázat: Tasks tábla definíciója .....	44
12. táblázat: Users tábla definíciója.....	44
13. táblázat: Roles tábla definíciója .....	45
14. táblázat: Bejelentkezés tesztelése .....	50
15. táblázat: Új ügyfél felvitelének a tesztelése .....	50
16. táblázat: Feladat projekthez adásának a tesztelése .....	51
17. táblázat: Árajánlat nyomtatásának a tesztelése.....	51
18. táblázat: Feladat lezárásának a tesztelése .....	51
19. táblázat: Új kampány felvitelének a tesztelése .....	52
20. táblázat: Hírlevél kiküldésének a tesztelése .....	52

## 6. Ábrajegyzék

1. ábra: XAMPP szervercsomag letöltése a hivatalos weboldalról.....	4
2. ábra: A XAMPP web szerver telepítő varázslója .....	4
3. ábra: A XAMPP web szerver vezérlőpanelja .....	5
4. ábra: A Composer letöltése a hivatalos weboldalról.....	6
5. ábra: A Composer telepítő varázslója .....	6
6. ábra: A Node.js letöltése a hivatalos weboldalról .....	7
7. ábra: A Laravel .env fájlban az adatbázis kapcsolat beállításai .....	8
8. ábra: A CRM rendszer bejelentkező felülete .....	9
9. ábra: A program kezdőlapja, a widget-ekkel .....	10
10. ábra: A menürendszer .....	10
11. ábra: Az „Ügyfelek” oldal .....	11
12. ábra: Hibásan vagy üresen kitöltött adatnál lévő figyelmeztetés .....	12
13. ábra: Az ügyfél és a kapcsolattartó összekapcsolása.....	13
14. ábra: Új projekt létrehozása .....	17
15. ábra: A „Projektek - Árajánlat fázis” oldal .....	17
16. ábra: A PDF formátumban kinyomtatott digitális árajánlat .....	19
17. ábra: A „Feladatok” oldal.....	21
18. ábra: A „Kampányok” oldal.....	23
19. ábra: A kiküldött e-mail-ek listája .....	23
20. ábra: Az Adminisztrátor felhasználóknak elérhető BREAD létrehozása oldal .....	31
21. ábra: Az Adminisztrátor felhasználóknak elérhető BREAD létrehozása oldalon az egyes mezők beállítása .....	32
22. ábra: Az MVC tervezési minta bemutatása a Projekt modellel és a hozzá kapcsolódó vezérlőkkel, nézettel.....	33
23. ábra: A program adatbázis tervezete .....	46

## 7. Irodalomjegyzék

- [1] „Vállalatirányítási rendszer,” [Online]. Available: <https://vallalatiranyitasi-rendszer.hu/crm-rendszer-jelentese/>. [Hozzáférés dátuma: 25. április 2021].
- [2] „Web-server.hu,” [Online]. Available: [https://web-server.hu/kezdő\\_felhasználók/mi-az-a-php-mysql/](https://web-server.hu/kezdő_felhasználók/mi-az-a-php-mysql/). [Hozzáférés dátuma: 1. május 2021].
- [3] „Laravel hivatalos weboldala,” [Online]. Available: <https://laravel.com/>. [Hozzáférés dátuma: 1. május 2021].
- [4] „Digitális tankönyvtár,” [Online]. Available: [https://regi.tankonyvtar.hu/hu/tartalom/tamop425/0038\\_informatika\\_Projektlabor/ch01s04.html](https://regi.tankonyvtar.hu/hu/tartalom/tamop425/0038_informatika_Projektlabor/ch01s04.html). [Hozzáférés dátuma: 1. május 2021].
- [5] „Voyager - a hiányzó Laravel admin felület,” [Online]. Available: <https://voyager.devdojo.com/>. [Hozzáférés dátuma: 1. május 2021].
- [6] „Linuxportál,” [Online]. Available: <https://www.linuxportal.info/enciklopedia/m/mysql>. [Hozzáférés dátuma: 2. május 2021].