Vizsgaremek

Számlázási Adatkezelő: Intelligens szoftver a hatékony számlázáshoz és kimutatásokhoz.

Invoicing Data Manager: Intelligent software for efficient invoicing and statements.

Bakó Péter

Schwarczenberger Ferenc

Vizsgaremek adatlap

A Vizsgaremek készítői:

Neve: Bakó Péter

E-mail címe: Bakopeter71@gmail.com

Neve: Schwarczenberger Ferenc

E-mail címe: Schwarci@gmail.com

A Vizsgaremek témája:

A vizsgaremek célja egy olyan szoftver megvalósítása, amely számlázással kapcsolatos adatokat tárol, kezel egy adatbázisban, melyből megfelelő hozzáféréssel lekérdezések és kimutatások érhetők el.

A Vizsgaremek címe:

Számlázási Adatkezelő: Intelligens szoftver a hatékony számlázáshoz és kimutatásokhoz.

Konzulens:

Harangozó Zsolt

Kelt: Budapest, 2023.11.11

............................................................ ............................................................

Bakó Péter Schwarczenberger Ferenc

............................................................

Harangozó Zsolt

A Vizsgaremek készítőinek aláírása, a konzulens aláírása.

Eredetiség nyilatkozat

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

Budapest, 2024.04.16.

............................................................ Bakó Péter

Tanuló aláírása

............................................................

Schwarczenberger Ferenc

Tanuló aláírása

Vizsgaremek konzultációs lap

Alulírott „Konzulens Neve” konzulens aláírásommal igazolom Példa Mihály, Fejleszt Elek, Meg Endre nevű tanulók konzultációkon való részvételét[[1]](#footnote-1).

|  |  |  |  |
| --- | --- | --- | --- |
| **Dátum** | **Téma** | **Tanuló aláírása** | **Konzulens aláírása** |
| 2023.11.06 | Vázlat az adatbázisról Terv az elérhető funkciókról (lehetőség szerint use-case) - Használni kívánt API tesztelése legalább Testfully-val. |  |  |
|  |
|  |
|  |  |  |  |
|  |
|  |
|  |  |  |  |
|  |
|  |
|  |  |  |  |
|  |
|  |
|  |  |  |  |
|  |
|  |
|  |  |  |  |
|  |
|  |

Tartalom

[1 Bevezetés 6](#_Toc160754328)

[2 Alapfogalmak 7](#_Toc160754329)

[2.1 Weboldal elrendezése 7](#_Toc160754330)

[2.2 Az alkalmazás interfésze 7](#_Toc160754331)

[2.3 Folyamatábrák 7](#_Toc160754332)

[2.4 Adatbázis 9](#_Toc160754333)

[2.5 Front End 9](#_Toc160754334)

[2.6 Back-end 10](#_Toc160754335)

[3 Fejlesztés/telepítés 11](#_Toc160754336)

[3.1 Munkamegosztás 11](#_Toc160754337)

[3.2 Adatbázis részletes bemutatása 12](#_Toc160754338)

[3.3 Telepítés 13](#_Toc160754339)

[4 FrontEnd fejlesztés bemutatása 15](#_Toc160754340)

[5 Backend fejlesztés bemutatása 16](#_Toc160754341)

[6 Tesztelés 17](#_Toc160754342)

[7 Továbbfejlesztési lehetőségek 18](#_Toc160754343)

[8 Zárszó 19](#_Toc160754344)

# Bevezetés

A modern üzleti környezetben a hatékony és pontos számlázási folyamatok kritikus fontosságúak a vállalkozások számára. A számlázási adatok kezelése és tárolása kulcsfontosságú az üzleti tevékenységek szervezett és hatékony irányításához, valamint az üzleti döntések megalapozásához. Egy jól felépített és funkcionalitásban gazdag számlázási szoftver lehetővé teszi a vállalkozások számára, hogy hatékonyan nyomon kövessék az ügyfelekkel kapcsolatos pénzügyi tranzakciókat, optimalizálják a bevételi folyamatokat és biztosítsák a jogi követelményeknek való megfelelést.

Ebben a szakdolgozatban egy olyan szoftver megvalósítását mutatjuk be, amely specifikusan a számlázási adatok kezelésére és tárolására fókuszál. A vizsgaremek célja egy olyan szoftver létrehozása, amely lehetővé teszi a számlázással kapcsolatos adatok tárolását, kezelését egy adatbázisban. Ennek a szoftvernek a célja továbbá, hogy felhasználóbarát felületen keresztül lehetőséget biztosítson megfelelő hozzáféréssel lekérdezések és kimutatások elérésére, melyek segítik az üzleti döntéshozatalt és a pénzügyi teljesítmény elemzését.

A szoftver fejlesztése során hangsúlyt fektetünk az adatbiztonságra, a megbízhatóságra és a skálázhatóságra, hogy biztosítsuk a széles körű alkalmazhatóságot és a stabil működést akár kis- és középvállalkozások, akár nagyobb vállalatok számára. Emellett figyelembe vesszük a felhasználói igényeket és visszajelzéseket annak érdekében, hogy a szoftver megfeleljen az ügyfelek elvárásainak és könnyen integrálható legyen az üzleti folyamatokba.

Ezen szakdolgozat során részletesen bemutatjuk a szoftver tervezését, fejlesztését és tesztelését, valamint kiemelten foglalkozunk az adatbázis tervezésével és optimalizálásával. Emellett áttekintjük a felhasználói felület kialakításának folyamatát és funkcionalitását, valamint bemutatjuk a szoftver működését szimuláló esettanulmányokat és példákat.

A cél az, hogy a szakdolgozat eredményeként egy jól működő, hatékony és felhasználóbarát számlanyilvántartó szoftvert mutassunk be, amely segíti a vállalkozásokat a pénzügyi folyamatok optimalizálásában és a hatékonyabb üzleti döntések meghozatalában.

# Alapfogalmak

## Weboldal elrendezése

A megjelenés egyszerűségét és letisztultságát tartottuk szem előtt a tervezés során. Az egyszerűen navigálható és átlátható menürendszerrel, valamint tömör, világos megfogalmazásokkal igyekeztünk biztosítani, hogy minden felhasználó számára könnyen érthető legyen az információk elérése. Az oldal dizájnja a modern stílusjegyeket követi, elsősorban az alapszínek felhasználásával. A vizuális megjelenés harmonikus felületet eredményez. Célunk egy összehangolt megjelenés létrehozása volt.

## Az alkalmazás interfésze

A grafikus felület könnyen kezelhető letisztult dizájnnal rendelkezik. A cél az volt, hogy mindenki számára egyértelmű legyen, ezért kerültük a bonyolult funkciókat és megfogalmazásokat, hogy biztossá tegyük a bátrabb használatot Az információk túlzott mennyiségének átadása azonban nehezítheti az azonnali felismerést, különösen hosszú napok után. A mindennapi ingerek tömkelege csökkenti a teljesítményt, így a letisztult elemek különösen értékesek lehetnek ebben az összefüggésben.

## Folyamatábrák

Ezek az elemek segítenek megérteni a folyamatokat és az elképzeléseket. Az elgondolásoktól indultunk el, és a folyamat során számos fontos lépést tettünk meg. A kezdeti tervezéstől egészen a megvalósításig sok apró részletet vettünk figyelembe, és ezek az ábrák segítenek ezt a folyamatot szemléltetni. Az alábbi ábrákon látható a web- és asztali alkalmazás felépítése:

A képen diagram, tervezés látható

Automatikusan generált leírás

1. ábra- Weboldal beléptetés (forrás: https://creately.com/diagram/example/gy2h2rov2/login-process)

## Adatbázis

Az adatbázis definíciója a következőképpen fogalmazható meg: "Az adatbázis egy strukturált adathalmaz, melyet számítógépen tárolnak és rendeznek, annak érdekében, hogy az elemei közötti kapcsolatok megőrzésre kerüljenek. Az adatbázisok célja, hogy hosszú távon megőrizzék az adatokat, valamint gyorsan kereshetőek legyenek."

Egy relációs adatbázist hoztunk létre, melyben entitások, attribútumok, táblák, mezők és rekordok találhatóak. Az entitások az adatok tárolását szolgálják, az attribútumok az egyedi tulajdonságokat definiálják, míg a táblák átjáróként kapcsolódnak egymáshoz. A mezők az oszlopokat jelölik, míg a rekordok a sorokat képviselik az adatok tárolásában.

Az adatbázisban lévő kapcsolatok különböző formákat ölthetnek:

1:1 kapcsolatnál minden egyes entitáshoz pontosan egy másik entitás tartozhat. Ezt úgy jelöljük, hogy a kapcsolat minden entitás felé egy nyíllal mutat.

1:N kapcsolatnál az egyik entitáshoz több másikat is hozzárendelhetünk, azonban a másik entitáshoz minden példányhoz csak egyetlen entitás tartozhat.

N:M kapcsolatnál egy entitás több másikkal van kapcsolatban, és fordítva is.

## Front End

A front-end megfogalmazása így hangzik: "A front-end, más néven frontend vagy front end, azon programok és weboldalak része, amely közvetlen kapcsolatban áll a felhasználóval. Fő feladata az adatok megjelenítése és fogadása a felhasználó vagy ritkábban más rendszerek részéről."

A VUE.Js keretrendszert alkalmaztuk, mely segítségével JavaScript programozási nyelven valósítottuk meg interaktív weboldalunkat. A böngésző a forráskódban található nyelvet tölti le, majd értelmezi és futtatja. A stílusokat a CSS (Cascading Style Sheet) segítségével állítottuk be, mely felelős a weboldal kinézetéért, azaz a megjelenésért.

Az HTML kódok ismerete szükséges a JavaScript és CSS használatához, mivel ezek az elemek összekapcsolódnak.

A teljesen reszponzív webes front-end standard technológiákkal készült, és rendkívül hasznos, mivel támogatja az összes elterjedt böngészőt, mind asztali gépeken, mind mobil eszközökön egyaránt.

## Back-end

A backend fogalma a következő: "A back-end, más néven backend vagy back end, a programok és weboldalak olyan rejtett része, amely a felhasználó elől rejtve, a tényleges számításokat végzi. Fő feladata a front end által küldött adatok feldolgozása és az eredmények visszajuttatása a front end felé."

A Laravel keretrendszert választottuk a backend megvalósításához, ami egy nyílt forráskódú PHP rendszer. Gyorsítja a fejlesztési folyamatot és biztonságosan használható. Az adatok migrálása könnyen megvalósítható fejlesztés közben. A jelszavak biztonságosan kezelhetők, mivel PHP kódot használunk az SQL kód helyett, így csökkentve az SQL injekciós támadások kockázatát.

A back-end létrehozása során olyan logikai struktúrát alakítottunk ki, amely stabil működést biztosít a háttérben.

# Fejlesztés/telepítés

## Munkamegosztás

A folyamatok kialakítása során törekedtünk arra, hogy minden területet külön-külön definiáljunk és megtervezzünk. Az adatbázis létrehozásakor közösen gondolkodtunk, először ábrákkal próbáltuk szemléltetni a kapcsolatokat, majd megértettük az elsődleges és idegen kulcsokat, valamint azok összefüggéseit, amelyek elengedhetetlenek voltak a számlák nyilvántartásához. Mindig lépésről lépésre haladtunk, és ha problémába ütköztünk, közösen megbeszéltük és kerestük a megoldást.

A feladatok megosztása a következőképpen alakult:

* Péter felelt az adatbázis és a FrontEnd létrehozásáért.
* Ferenc pedig az API kéréseket kezelte, a BackEnd fejlesztésén dolgozott, és a dokumentációt készítette el.

Összehangoltan dolgoztunk a projektmunka során. Ha nem volt egyértelmű, hogy mit szeretnénk elérni, segítettünk egymásnak tisztázni a célokat és terveket. A különböző területeket együtt ellenőriztük, közösen megvitattuk a kódokat, és egymás munkáját kiegészítettük. A csapat minden tagja teljes erőbedobással vett részt a projektben, segítettük egymást, ha szükség volt rá.

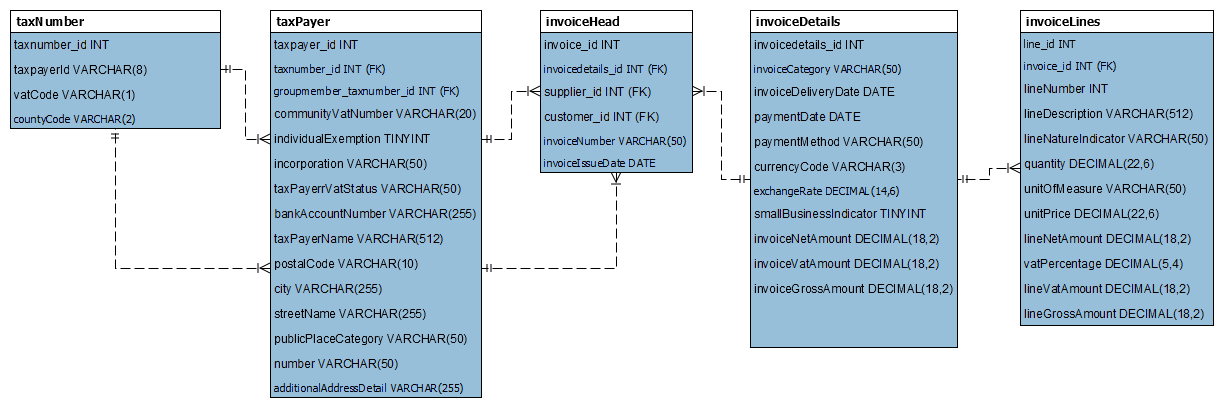
A kommunikáció kiemelkedő fontosságú volt számunkra. Figyeltünk egymás kéréseire és igényeire, rendszeresen tartottunk megbeszéléseket szabadidőnkben is. Hetente tájékoztatókat tartottunk az aktuális helyzetről, és különböző platformokon mutattuk be, hogy ki hol tart éppen. Használtuk a Teams, Google Meet és Facebook Messenger platformokat a konzultációkhoz és tanuláshoz. A GitHub-et állandóan használtuk, így a csapat minden tagja és a konzulens is nyomon követhette a fejlődésünket, és ellenőrizhette, hogy haladunk-e előre.

Összességében egy nagyon hatékony csapatként dolgoztunk, és elértük a kitűzött célokat.

## Adatbázis részletes bemutatása

Minden tábla rendelkezik egyedi azonosítóval (id), amely int típusú.

A táblák közötti kapcsolatokat az alábbi módon alakítottuk ki:



2. ábra- Az adatbázisunk felépítése

**invoicedetails:** Az elsődleges információk, amelyek az adatokat tartalmazzák. Ide tartozik az értékesítés kategóriája, szállítás és fizetés dátuma, fizetési mód, valuta kódja, árfolyam, kisvállalkozás jelző, számla nettó összege, áfa összege és bruttó összege.

**invoicehead:** Ez a tábla tartalmazza a számla fejlécéhez kapcsolódó információkat, mint például az ügyfél és szállító azonosítók, számlaszám, kibocsátás dátuma stb.

**invoicelines:** Itt találhatóak az egyes számlasorok adatai, például a sor sorszáma, leírása, termék vagy szolgáltatás jellege, mennyiség, egységár, nettó összeg, áfa százalék, áfa összeg és bruttó összeg.

**taxnumber:** Az adószámokat tartalmazó tábla, amely az adózók azonosítóit, áfa kódját és országkódját tárolja.

**taxpayer:** Az adózók adatait tartalmazó tábla, mint például a csoportos adószám, közösségi áfa szám, cégforma, adófizetői státusz, bankszámlaszám, adózó neve, postai irányítószám, város, utca stb.

A táblák az alábbi módon kapcsolódnak egymáshoz:

1. **invoicedetails** és **invoicehead** táblák:
   * Az **invoicedetails\_id** oszlop azonosítja az **invoicedetails** tábla rekordjait.
   * Az **invoice\_id** oszlop azonosítja az **invoicehead** tábla rekordjait.
   * Az **invoice\_id** oszloppal van kapcsolatban az **invoicedetails\_id** oszlop a kulcsként (**FOREIGN KEY**) az **invoicehead** táblában.
2. **invoicedetails** és **invoicelines** táblák:
   * Az **invoicedetails\_id** oszlop azonosítja az **invoicedetails** tábla rekordjait.
   * Az **invoice\_id** oszlop azonosítja az **invoicelines** tábla rekordjait.
   * Az **invoicelines** tábla tartalmazza a számla egyes sorait, és ezeket az **invoicedetails\_id** oszlop kapcsolja össze a számla részleteivel.
3. **taxnumber** és **taxpayer** táblák:
   * A taxnumber\_id oszlop azonosítja a **taxnumber** tábla rekordjait.
   * A taxnumber\_id oszlop a kulcs (**FOREIGN KEY**) a **taxpayer** táblában.
4. **invoicehead** és **taxpayer** táblák:
   * A **supplier\_id** és **customer\_id** oszlopok azonosítják a számla fejlécéhez tartozó beszállítót és ügyfelet.
   * Mindkét oszlop a **taxpayer\_id** oszlopra mutat, ami azonosítja az adott adózót a **taxpayer** táblában.
   * A **supplier\_id** és **customer\_id** oszlopok kulcsok (**FOREIGN KEY**) a **taxpayer** táblában.

Ezek a kapcsolatok biztosítják, hogy az adatok összekapcsolódjanak egymással és a megfelelő hivatkozásokat tegyék lehetővé az adatok közötti navigáláshoz és lekérdezéshez. Az SQL struktúra lehetővé teszi az adatok hatékony kezelését és tárolását a számlakezelő rendszerben.

## Telepítés

A különböző felületek használatához alapvető ismeretekre van szükség a felhasználó részéről. Az alábbi programok telepítését javaslom az áttekintés során: VsCode, PhpStorm, Xampp, Vue.js, Laravel. A VsCode, a PhpStorm és a Xampp telepítése a klasszikus módon végezhető el az adott szoftver weboldaláról letöltve majd a telepítőt futtatva. Fontos figyelni a számítógép operációs rendszerére és az adott program telepítése során meghatározott követelményekre.

A Vue.js telepítése a következő lépésekből áll:

1. Hozz létre egy könyvtárat a Vue projektnek.
2. Lépj be a könyvtárba.
3. Indítsd el a parancssort (vagy terminált, ha PhpStorm-ot használsz). Itt most a Windows parancssort alkalmazzuk.
4. Add ki a következő parancsot: **npm install -g @vue/cli**
5. Telepítsd a Vue CLI-t a globálisan -g kapcsolóval.
6. Ha a Vue CLI telepítése befejeződött, hozd létre a Vue projektet a **vue create project\_name** parancs segítségével.
7. Válaszd ki a projekt inicializálásának beállításait az interaktív kezelőfelület segítségével.
8. A projekt inicializálása után lépj be a projekt könyvtárába.
9. Indítsd el a Vue projektet a **npm run serve** paranccsal.
10. Az alkalmazás elindul a böngésződben a **localhost:8080** címen.

Ezek az utasítások segítenek a Vue.js telepítésében és a projekt létrehozásában, valamint az alkalmazás futtatásában. Fontos, hogy minden lépést kövess a megfelelő sorrendben, és ügyelj arra, hogy a szükséges programok telepítve legyenek a Vue projekt működéséhez.

A Laravel telepítése továbbra is hasonlóan zajlik:

1. Töltsd le a Laravelt.
2. Hozz létre egy új mappát a gépeden, vagy ha helyi webszervert használsz, nyisd meg a gyökérkönyvtárat.
3. Írd be a terminálba a következő parancsot: **composer create-project --prefer-dist laravel/laravel backend**
4. Ha elkészült, nyisd meg az IDE-dben.
5. A projektedet a .env kiterjesztésű fájlban tudod konfigurálni. Itt adhatod meg az elérési utat, az adatbáziskapcsolatot stb. Ha még nincs .env fájl, másold le az .env.example-t és nevezd át .env-re.
6. A projektet a PHP-ban elindítod a terminálban kiadott **php artisan serve** paranccsal.
7. A böngészőben nyisd meg a következő címet: **http://127.0.0.1:8000**

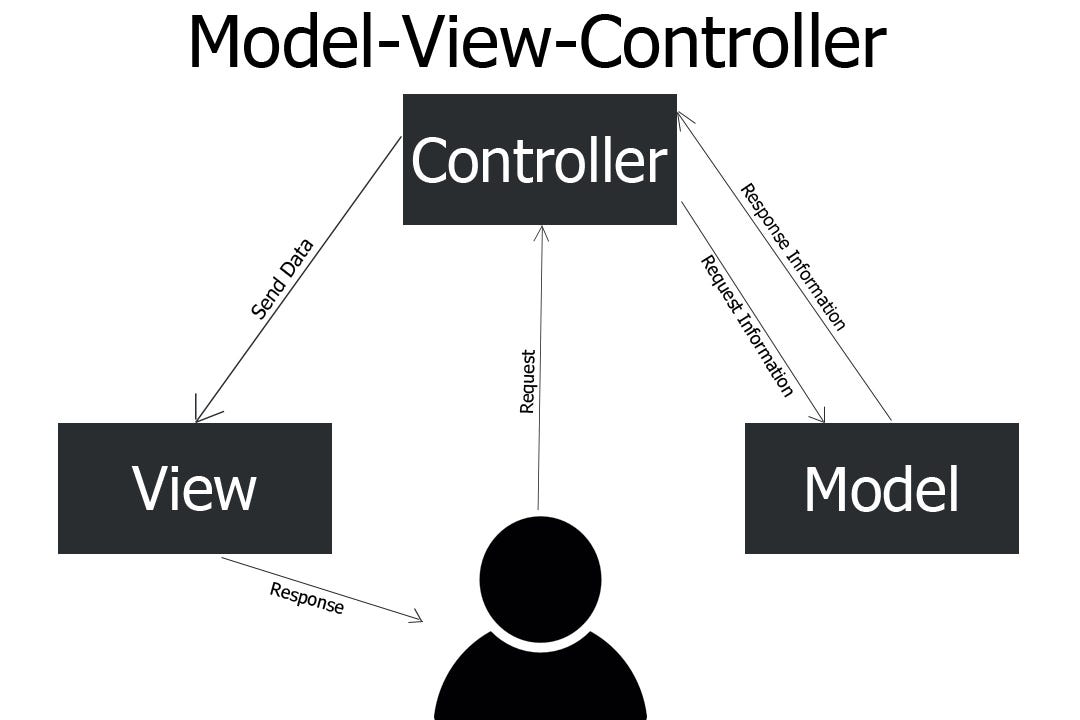
Nagyon fontos, hogy minden karaktert helyesen írj be, és kövesd az előírt lépéseket a megfelelő sorrendben. Ügyelj arra, hogy a szükséges programok telepítve legyenek, mivel ezek nélkül a Laravel és a Vue.js nem fog megfelelően működni. Legyél körültekintő és figyelmes a végrehajtás során.

# FrontEnd fejlesztés bemutatása

A Front-end felület fejlesztéséhez a Vue.js keretrendszert használtuk, amelyben az alábbi fontosabb elemeket szeretnénk bemutatni:

A Components mappában került létrehozásra a MenutItems.vue, Navbar.vue, valamint a NavbarStyles.css kiterjesztésű fájl. Az export kifejezéssel egy másik .vue fájlban lehet majd használni a const-ként megjelölt MenuItems elemeket, amelyek a következők lennének: kezdőlap, számlák, kapcsolat. Itt kapják meg a title-t (azaz a címet) valamint az url-t az elérési útvonalban szereplő nevüket.

# Backend fejlesztés bemutatása



3. ábra- MVC modell

Az ábrával szeretnénk bemutatni a controller, modell és felhasználói felület működési folyamatát. A controller, más néven vezérlő szerkezet, felelős a felhasználói bemenetek kezeléséért és a modellekben történő módosítások végrehajtásáért. Gyakorlatilag az adatok feldolgozását végzi és kimeneti adatokat szolgáltat a felhasználó számára. A controllerben lévő módszerek döntik el, hogy melyik felületet jelenítik meg, vagy melyik HTML-t rendelik hozzá.

A back-end során kiemelten fontosak a controllerek és modellek. A controllerek felelnek az adatok kezeléséért és a felhasználói bemenetek feldolgozásáért, míg a modellek adataink strukturálásáért és kezeléséért felelnek.

# Tesztelés

A weboldalt különböző környezetekben teszteltük, beleértve a Windows és Mac operációs rendszereket is, ahol minden rendben működött. A menüpontok és a weboldal elrendezése megfelelt a reszponzív kialakítás követelményeinek. Fontos megjegyezni, hogy egyes webhelyek nem engedik a beágyazott megjelenítést, ezért ezeket csak külön ablakban lehet megnyitni a felkínált hivatkozásra kattintva.

Az adatokat a fent említett táblázatban rögzítettük az adatbázisunkban, majd sikeresen bejelentkeztünk a megadott adatokkal. Amennyiben helytelen felhasználónév-jelszó kombinációt adtunk meg, a bejelentkezés sikertelen volt, és a korábban a háttérben rögzített hibaüzeneteket kaptuk meg. Ugyanez a helyzet akkor is, ha új regisztrációt próbálunk végrehajtani: ha minden adatot megfelelően kitöltünk, akkor a regisztráció sikeres lesz, és tovább lepünk a bejelentkezési felületre. Ha azonban hiányosan töltjük ki a regisztrációs űrlapot, akkor nem lehet befejezni ezt a folyamatot.

Összességében sikeresen lezárult a tesztelés, és eredményesen megoldottuk a projektünket.

# Továbbfejlesztési lehetőségek

A jövőben további lehetőségek állnak rendelkezésre, amelyek segítségével még népszerűbbé tehetjük a létrehozott Számlaelemző szoftvert.

Elképzelhető, hogy további elemzési és listázási lehetőségekkel színesítjük az alkalmazást, így még vonzóbbá téve azt a felhasználók számára. Ezeket a témaköröket specializálhatjuk akár az különböző cégek, kisvállalkozások részére.

A jobb nyomon követhetőséghez elkészíthetnénk üzletemberek számára létrehozott mobilalkalmazást, így könnyedén és gyorsan bárhol, bármikor hozzáférhetnének a számláinkhoz. Ezen kívül egy online chat felületet is kialakíthatnánk, ami az érintettek közötti kommunikáció segítené.

A szoftvert használhatnánk cégcsoporton belül is, hogy ösztönözzük a résztvevőket az együttműködésre és a versenyszellemre. Továbbá cégek bemutatását is elvégezhetnénk a szoftveren belül, hogy segítsünk az cégek vezetőinek a tevékenységük fejlesztésében.

A jövőben még számos területen alkalmazhatnánk és fejleszthetnénk ezt az alkalmazást, remélve, hogy másokat is ihletünkkel és motivációnkkal látunk el a továbbfejlesztésre.

# Zárszó

Összefoglalva, a projektünk magába foglal egy rendkívül összetett és bonyolult rendszert parancsokkal, utasításokkal és logikai folyamatokkal. Ez valódi csapatmunka volt, ahol agilis fejlesztői munkamorálunk és képességeink segítségével sikeresen megbirkóztunk az előttünk álló kihívásokkal. Célunk az volt, hogy motiváló hatást váltunk ki azokból is, akik csak most érdeklődnek ezen területen. Az online platform lehetővé tette, hogy könnyebben és gyorsabban valósítsuk meg minden tevékenységünket, ami talán kedvet csinál azoknak is, akik eddig papíron végezték ezt a folyamatot. Reméljük, hogy mások is tovább viszik fejlesztésünk ötletét, és új irányokat nyitunk azoknak, akiket talán kissé demotiválttá tett a környezeti hatások által. Projektünknek köszönhetően reméljük, hogy megújító szereplővé váltunk.

1. *(A vizsgaremek leadásig kötelező minden tanulónak legalább három konzultációs alkalom, ennek hiányában a záródolgozat nem értékelhető!)*  [↑](#footnote-ref-1)