

SAPIENTIA ERDÉLYI MAGYAR TUDOMÁNYEGYETEM
MAROSVÁSÁRHELYI KAR,
SZÁMÍTÁSTECHNIKA SZAK



SAPIENTIA
ERDÉLYI MAGYAR
TUDOMÁNYEGYETEM

Tartalomkezelő rendszer projektek menedzselésére Web applikációval

DIPLOMAMUNKA

Témavezető:

Dr. Iclănzan David Andrei

Egyetemi tanár

Végzős hallgató:

Takács Norbert

2024

**UNIVERSITATEA SAPIENTIA DIN CLUJ-NAPOCA
FACULTATEA DE ȘTIINȚE TEHNICE ȘI UMANISTE,
CALCULATOARE**



**UNIVERSITATEA
SAPIENTIA**

Sistem de gestionare a conținutului pentru urmărirea proiectelor cu
aplicație Web

LUCRARE DE DIPLOMA

Coordinator științific:

Absolvent:

Dr. Iclănzan David Andrei
Profesor universitar

Takács Norbert

2024

**SAPIENTIA HUNGARIAN UNIVERSITY OF
TRANSYLVANIA
FACULTY OF TECHNICAL AND HUMAN SCIENCES
COMPUTER SCIENCE SPECIALIZATION**



SAPIENTIA
HUNGARIAN UNIVERSITY
OF TRANSYLVANIA

Content management system for project tracking with Web application

BACHELOR THESIS

Scientific advisors:
Dr. Iclănzan David Andrei
Full Professor

Student:
Takács Norbert

2024

Kivonat

A Project Lifecycle Management(ProLife Management) applikáció egy modern webes platform, amely kiemelkedő jelentőséget tulajdonít a tartalomkezelő rendszerek (TKR) által nyújtott előnyöknek. Ezek a rendszerek lehetővé teszik, hogy több felhasználó egyszerre tárolhasson adatokat és ezeket egymással megoszthassa. Emellett lehetővé teszik a hozzáférés szerep szerinti szabályozását, amelynek révén a felhasználó szerepe határozza meg, hogy mely adatokat láthatja vagy módosíthatja.

A ProLife Management applikáció célja a könnyű adattárolás és adatelérés lehetőségeinek biztosítása, minimalizálva ezzel a redundáns adatbevitelt. Ezen túlmenően segíti a kimutatások összeállítását, elősegítve ezzel a hatékonyabb döntéshozatalt és üzleti stratégiák kidolgozását. A platform továbbá támogatja a felhasználók közti kommunikációt, létrehozva egy közös térként szolgáló virtuális környezetet, ahol a csapatok könnyen megoszthatják információikat és együtt dolgozhatnak a projektekkel kapcsolatos feladatokon.

A rendszer két fő modulból áll: a Cégek modulból és a Kliens modulból. A Cégek modul lehetővé teszi a vállalatok számára, hogy regisztráljanak, projekteket hozzanak létre és kezeljenek, valamint meghívják munkatársaikat és klienseiket. A Kliens modul pedig lehetővé teszi a meghívott kliensek számára a projektekhez való hozzáférést, dokumentumok feltöltését és kezelését, valamint a cégek munkatársaival való kommunikációt.

Az alkalmazás fejlesztése során kiemelt figyelmet fordítottunk az adatbiztonságra és a felhasználói élményre, hogy biztosítsuk a stabil és intuitív működést. Célunk az, hogy elősegítsük a projektmenedzsment folyamatok hatékonyságát és átláthatóságát, ezáltal támogatva a vállalatok sikeres működését és ügyfélkapcsolatait.

Kulcsszavak: tartalomkezelő rendszer (TKR/CMS), Web alkalmazás, projekt kezelés, ügyfélszolgálat

Rezumat

Aplicația Project Lifecycle Management(ProLife Management) este o platformă modernă bazată pe web, care pune un accent deosebit pe avantajele oferite de sistemele de gestionare a conținutului (CMS). Aceste sisteme permit mai multor utilizatori să stocheze date simultan și să le partajeze între ei. În plus, ele permit controlul accesului bazat pe roluri, care determină ce date pot vizualiza sau modifica utilizatorii în funcție de rolurile lor.

Scopul aplicației ProLife Management este de a oferi posibilități ușoare de stocare și acces la date, minimizând astfel introducerea redundantă a datelor. În plus, facilitează întocmirea rapoartelor, promovând luarea deciziilor mai eficiente și dezvoltarea strategiilor de afaceri. Platforma sprijină, de asemenea, comunicarea între utilizatori, creând un mediu virtual care servește ca un spațiu comun în care echipele pot împărtăși ușor informații și colabora la sarcinile legate de proiecte.

Sistemul constă din două module principale: modulul Companii și modulul Clienți. Modulul Companii permite întreprinderilor să se înregistreze, să creeze și să gestioneze proiecte, și să-și invite angajații și clienții. Modulul Clienți permite clienților invitați să acceseze proiectele, să încarce și să gestioneze documente, și să comunice cu angajații companiei.

În timpul dezvoltării aplicației, am acordat o atenție deosebită securității datelor și experienței utilizatorului pentru a asigura o funcționare stabilă și intuitivă. Scopul nostru este de a îmbunătăți eficiența și transparența proceselor de management al proiectelor, sprijinind astfel funcționarea de succes a companiilor și relațiile lor cu clienții.

Cuvinte cheie: sistem de management al conținutului (CMS), aplicație web, management de proiect, serviciu clienți

Abstract

The Project Lifecycle Management(ProLife Management) application is a modern web-based platform that emphasizes the advantages offered by content management systems (CMS). These systems enable multiple users to store data simultaneously and share it with each other. Additionally, they allow access control based on roles, which determines which data users can view or modify according to their roles.

The goal of the ProLife Management application is to provide easy data storage and access, minimizing redundant data entry. Furthermore, it aids in compiling reports, promoting more efficient decision-making and business strategy development. The platform also supports communication among users by creating a virtual environment that serves as a shared space where teams can easily share information and collaborate on project-related tasks.

The system consists of two main modules: the Companies module and the Clients module. The Companies module allows businesses to register, create, and manage projects, and invite their employees and clients. The Clients module enables invited clients to access projects, upload and manage documents, and communicate with company employees.

During the development of the application, we paid particular attention to data security and user experience to ensure stable and intuitive operation. Our goal is to enhance the efficiency and

transparency of project management processes, thereby supporting the successful operation of companies and their client relationships.

Keywords: content management system (CMS), web application, project management, customer service

Tartalomjegyzék

Bevezető.....	10
1.1. Témaválasztás indoklása	10
Központosított Adatkezelés	10
Jobb Kommunikáció és Együttműködés	10
Biztonság és Hozzáférés Szabályozás	10
Hatékonyabb Dokumentumkezelés	10
Idő- és Erőforrás Megtakarítás.....	11
1.2. Szakirodalmi áttekintő	11
1.3. Kutatási kérdések	11
Back-end	12
Front-end	12
• JavaScript.....	15
Adatbázis.....	15
• MySQL	15
Version Control Systems (VCS).....	16
• GitHub	16
1.4. Célkitűzések	16
Rendszerspecifikáció	17
2.1. Felhasználói követelmények	17
2.1. Rendszerkövetelmények	20
a. Funkcionális követelmények:.....	20
b. Nem funkcionális követelmények:	21
Rendszerarchitektúra	24
• Adatbázis:.....	24
• Szerveroldali alkalmazás	25
• Web Szerver	27
• Felhasználó interfész.....	27
Megvalósítás	29

4.1. ProLifeManagemenet Applikáció	29
4.2. Javascriptek használata	32
4.3. Tesztelések	32
4.4. Projektmenedzsment	32
Összefoglaló	32
5.1 Következtetések	32
5.2. Fejlesztési lehetőségek	32
5.3. Lehetséges felhasználási terület	32
Irodalomjegyzék	32

1.fejezet

Bevezető

1.1. Témaválasztás indoklása

Gyakran találkoztam olyan helyzetekkel, amikor az információk és dokumentumok kezelése, például a WhatsApp vagy az email használatával nehézkessé és időigényessé vált. Az applikáció választása mellett több fontos érv szól.

Központosított Adatkezelés

Az applikáció lehetővé teszi a központosított adatkezelést, ahol minden releváns információ és dokumentum egy helyen érhető el. Ezzel elkerülhető az információ elvesztése és az adatok szétszóródása különböző platformokon. A központosított rendszer révén minden projekt résztvevője mindig hozzáférhet a legfrissebb információkhoz és dokumentumokhoz, ami jelentősen növeli a munkafolyamatok hatékonyságát.

Jobb Kommunikáció és Együttműködés

Az applikáció kialakítása lehetővé teszi a hatékonyabb kommunikációt és együttműködést a csapattagok és a kliensek között. A felhasználók könnyedén megoszthatják egymással az információkat, hozzászólásokat fűzhetnek a projektekhez, és valós időben követhetik nyomon a projekt előrehaladását. Ezzel szemben a WhatsApp és email használata gyakran áttekinthetetlenné teszi a kommunikációt, és nehéz követni, ki mit és mikor mondott.

Biztonság és Hozzáférés Szabályozás

A Project Lifecycle Management applikáció fejlesztése során különös figyelmet fordítottam az adatbiztonságra. A rendszer lehetővé teszi a hozzáférés szerep szerinti szabályozását, így biztosítva, hogy csak azok férjenek hozzá bizonyos adatokhoz, akiknek valóban szükségük van rá. Ez jelentősen növeli a projekt biztonságát, és megakadályozza az illetéktelen hozzáférést.

Hatékonyabb Dokumentumkezelés

Az applikációban egyszerűen feltölthetők, megoszthatók és kezelhetők a projekt dokumentumai. A PDF-ek és képek közvetlenül megnyithatók, míg más típusú dokumentumok letölthetők. Ez a funkció különösen hasznos a szerződések, árajánlatok és számlák kezelésében, mivel minden szükséges dokumentum egy helyen található, és könnyen hozzáférhető a projekt összes résztvevője számára.

Idő- és Erőforrás Megtakarítás

Az applikáció használata jelentős idő- és erőforrás-megtakarítást eredményez. Azáltal, hogy minden információ és kommunikáció egyetlen platformon történik, a csapatok és kliensek kevesebb időt töltenek az információk keresésével és megosztásával. Ez lehetővé teszi, hogy több időt fordítsanak a tényleges munkavégzésre és a projekt előrehaladására.

Összességében a ProLife Management applikáció fejlesztése és használata nagyban megkönnyíti a projektek kezelését és a klienskommunikációt, növeli a hatékonyságot, és biztosítja az információk és dokumentumok biztonságos kezelését. Ezek az előnyök teszik ezt a témaválasztást logikus és indokolt döntéssé számomra.

1.2. Szakirodalmi áttekintő

1.3. Kutatási kérdések

- Milyen technológiai eszközök állnak rendelkezésre CMS rendszer felállításához?
- Hogyan garantálható a felhasználói adatok biztonsága?
- Milyen felhasználói élményt javító technikákat alkalmazhatunk a frontend fejlesztése során?
- Hogyan optimalizálható a teljesítmény egy nagy felhasználói bázissal rendelkező CMS rendszer esetében?

Teljesítményoptimalizálási technikák, mint például a gyorsítótárzás, adatbázis indexelés, és aszinkron műveletek.

- Milyen nehézségekkel találkozhatunk egy CMS rendszer létrehozása és kezelése során?
- Hogyan integrálhatók az API-k és a harmadik féltől származó szolgáltatások?
- Hogyan biztosítható a rendszer bővíthetősége és rugalmassága az elkövetkező frissítések és fejlesztések esetén?

Egy hatékony CMS (Content Management System) rendszer megvalósításához számos technológiát és eszközt lehet használni. Az alábbiakban bemutatom az általam választott technológiákat és azoknak alternatív opcióit:

Back-end

- **PHP:** Széles körben használt és támogatott nyelv, különösen webes alkalmazások fejlesztéséhez, amelyet számos CMS rendszer használ, mint például WordPress.

Előnyök:

- A nagy közösségnek és a kiterjedt dokumentációnak köszönhetően könnyedén lehet találni információt és segítséget.
- A PHP egyik legnagyobb előnye, a villámgyorsaság. Sok szakértő állítja, hogy ez a programozási nyelv körülbelül háromszor olyan gyors, mint a Python, és az aktuális (7. verzió és ennél újabb) kiadások még a korábbiaknál is gyorsabbak.[1]

Hátrányok:

- Mivel a PHP nyílt forráskódú szoftver, ezért egy könnyen hozzáférhető, ASCII szöveges fájl is tartozik hozzá. Ez egyszerűen azt jelenti, hogy bármilyen kódot is írunk, az egész nyilvánosság számára könnyen megtekinthető, az esetleges hibákkal együtt.[1]

Alternatív opciók: Python, Java, Node.js

Front-end

- **HTML5:** A HTML(HyperText Markup Language) egy leíró nyelv(**1.1. ábra**), nem egy programozási nyelv, melynek fő feladata az, hogy a forráskódot átalakítsa gépi nyelvvé.



Text Only

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Nested List

Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

- Lorem ipsum dolor sit amet, consectetur adipiscing elit.
 - Lorem ipsum dolor sit amet, consectetur adipiscing elit.

1.1. ábra. Html leíró nyelvben létrehozott weboldal

Önmagában legtöbbet egy statikus weboldalt tud létrehozni, semmi lehetőséget nem ad arra, hogy az elemeket valahogyan elrendezni tudnánk. Legtöbbet egy képet, egy videót tudunk kimutatni vagy egy második weboldalt tudunk beágyazni.

Ahhoz, hogy dizájnos, interaktív és dinamikus weboldalt hozzassunk létre, használni kell a CSS(Cascading Style Sheets) stílusleíró nyelvet és a JS(Java Script) programzási nyelvet. A HTML legújabb változata, a HTML5, megkönnyíti a fejlesztők dolgát azzal, hogy előre megírt JS programozási kódokat kínál.

Előnyök:

- Az összes nagyobb böngésző, beleértve a Chrome-ot, a Firefoxot, a Safarit és az Internet Explorert, támogatja a HTML-t. Ez garantálja, hogy a HTML használatával készült webhelyek különböző eszközökön megfelelően megtekinthetők és navigálhatók.[2]
- A HTML két objektum létezésével támogatja az adattárolást. Ezek a következők:
A window.sessionStorage: Amikor egy adott böngészőablak bezárul, az adatok törlődnek.
localStorage.window: Ez olyan adatokat tárol, amelyeknek nincs lejáratási ideje.[2]

Hátrányok:

- A HTML nem programozási nyelv, ezért nem alkalmas bonyolult programozási feladatokra. Ez egy olyan szöveges leírónyelv, amely meghatározza, hogyan kell a webes anyagokat megjeleníteni és rendszerezni. Ez azt jelenti, hogy az olyan programozási nyelvekkel ellentétben, mint a Python vagy a JavaScript, a HTML nem képes bonyolult logikát kezelni, számításokat végezni vagy adatokat manipulálni. Ahhoz, hogy kifinomultabb funkciókat valósítsanak meg webhelyeiken, a fejlesztőknek ezért gyakran extra programozási nyelveket vagy keretrendszereket kell használniuk.[2]
- HTML nem képes bonyolult interakciók vagy dinamikus információk kezelésére, elsősorban a webes tartalom strukturálására és megjelenítésére használják. Ez azt jelenti, hogy interaktív elemek, például űrlapok, animációk és bonyolult felhasználói felületek nem hozhatók létre pusztán a HTML segítségével.[2]

- **CSS3:** Dizájnolás és elrendezés egyszerűsítése és gyorsítása előre definiált osztályok segítségével.

Előnyök:

- A karbantartás kevesebb erőfeszítést igényel, mert egyetlen sornyi kódfrissítés a weboldal egészét módosíthatja. Továbbá kevesebb munkára van szükség a weblap kódjának módosításához, ha fejlesztésekre van szükség.[3]
- Az egyének különböző okoseszközökkel férnek hozzá egy adott weboldalhoz. Ez lehet laptop, PC vagy okostelefon. A weboldalnak készülékkompatibilisnek kell lenniük, hogy ezt a szerepet ki tudják szolgálni. A CSS jobb kompatibilitást kínál, ami egy jobb felhasználói élményt garantál.[3]

Hátrányok:

- A böngészők eltérően működnek. Ezért meg kell győződni arról, hogy a weboldalon a CSS-kódok által végrehajtott módosítások minden böngészőben helyesen jelennek meg, ez által a fejlesztő munkaterhelése megnő.[3]

- **Bootstrap:** Egy front-end, a Twitter által fejlesztett, nyílt forráskódú keretrendszer, amely azért felelős, hogy megkönnyítse a fejlesztők számára a weboldalak létrehozását, és pedig azzal, hogy az előre definiált HTML, CSS és JavaScript komponensek által biztosítja a konzisztenciát és a reszponzivitást.[4]

Előnyök:

- Reszponzív dizájn, ami azt jelenti, hogy a weboldalak automatikusan alkalmazkodnak a képernyőméretekhez, tehát ezzel együtt mobilbarát weboldalakat, webapplikációkat hozunk létre.
- Az előre megtervezett komponensek, mint például a gombok, űrlapok, navigációs menük és ennél több más komponensnek hála, a fejlesztőnek nem kell minden egyes weboldal tervezését nulláról kezdenie.[4]

Hátrányok:

- Nagy mennyiségű kód, adat, ami lelassítja az oldal betöltésének az idejét. Úgy lehet ezt megoldani, hogy a Bootstrap telepítését testreszabjuk, hogy csak azokat a komponenseket tartalmazza, amire szükség van.

- Nagyon sok weboldal felépítésében észre lehet venni a használatát, ezért találkozni lehet azzal a megnehezítéssel, hogy hogyan is tűnjünk ki a tömegből.[4]
- **JavaScript:** Az interaktív és dinamikus elemek megvalósításához használt alapvető programozási nyelv a webfejlesztésre tekintve, amely szerintem jelentősen javítja a felhasználói élményt a weboldalakon való navigálás idején azzal, hogy az oldalak tartalmát újratöltés nélkül kezeli. Például HTML elemek létrehozása vagy törlése, dizájn módosítása, Ajax (Asynchronous JavaScript and XML) segítségével akár az is megvalósítható, hogy a háttérben kommunikálni tudjunk az adatbázissal. Érdekesség: “ Brendan Eich alkotta meg mindössze 10 nap alatt 1995-ben, amikor a Netscape Communications Corporation-nél dolgozott. ”.[5]

Előnyök:

- Rövid idő alatt nagy mennyiségű adatot lehet szállítani.[5]
- A programozási nyelv állandó fejlődésben van, új frissítéseket és változtatásokat hoznak létre, amelyek segítenek a teljesítmény optimalizációjában, emellett új funkciókat és szintaxisjavításokat hoznak magukkal.[5]
- Aszinkron műveletek elvégzésére is alkalmas, ami jelentősen növeli a teljesítményt.[5]

Hátrányok:

- A kód bárki számára látható, ami azzal jár, hogy kimondottan csökken a rendszer biztonsága, mivel az adatok nyitottak a hackelésre. Ezt a lehetőséget én a back-end, PHP programozási nyelvben próbálom megszüntetni, mivel minden adatbázis kérést leszűrök és hitelesítem.[5]
- A hibák csak futásidőben jelentkeznek, így a fejlesztőknek kissé nehezebbé válik a hibakeresés, főleg az aszinkron kódokban történő hibakeresés.[5]

Alternatív opciók: React, Vue.js, Angular

Adatbázis

- **MySQL:** A legelterjedtebb relációs adatbázis-kezelő rendszer, amelyet a CMS rendszerek széles körben alkalmaznak. Fontos tudni róla, hogy különböző operációs rendszeren futhat, mint a Windows, Linux, Solaris, stb. Éppen ezért, jó választásnak tartom a ProLifeManagement applikáció adatainak a tárolására és kezelésére.[6]

Előnyök:

- Mivel nyílt forráskódú szoftver, ezért ingyen bárki szabadon használhatja, s ezért a legtöbb vállalat előnybe részesíti.[6]
- Különböző optimalizálási technikáknak köszönhetően, mint az indexelés vagy a gyorsítótárazás(cache), a MySQL ismert a nagy teljesítménye miatt, különösen azoknál az alkalmazásoknál, ahol nagy igény van az olvasásra.[6]
- A négy kulcsfontosságú tulajdonság, az ACID(Atomicity, Consistency, Isolation and Durability) ennél az adatbázis-kezelő rendszernél is megtalálható, így magas fokú megbízhatóságot ígér az adatok tárolására.[6]

Hátrányok:

- Az optimalizálási technikák kihívást jelenthet a fejlesztők számára.[7]
- Egy központi SQL adatbázis meghibásodásának a következménye az lehet, hogy a teljes rendszer meghibásodik, s következésképpen az összes rá támaszkodó alkalmazás.[7]

Alternatív opciók: MongoDB, PostgreSQL

Version Control Systems (VCS)

- **GitHub:** A forráskód verziókezelése és nyomon követése. Az applikáció fejlesztése során kulcsfontosságú szerepet játszott a biztonsági mentések szempontjából. Abban az esetben, amikor nem voltam megelégedve, vagy a fejlesztés során rájöttem, hogy fölösleges az, amit addig produkáltam, ott volt a biztonsági mentés, így vissza tudtam lépni relatív könnyen és gyorsan az előző verzióhoz.

1.4. Célkitűzések

- Egy olyan webes applikáció megvalósítása, amely eleget tegyen a jelenlegi modern világnak, úgy interaktivitás és működés szempontjából, mint dizájnilag is. A tervem elérésének érdekében úgy döntöttem, hogy létező, előre lefejlesztett kódrészletek, interfészek után keresek, ezzel megnövelve esélyeimet arra, hogy kövessem a webapplikációk modern világát. Erre alkalmas példa a Whiteboard Team által lefejlesztett online fehér tábla API(Application Programming Interface). Ez annyit jelent, hogy egy olyan leprogramozott interfész áll a rendelkezésemre, amely által sikerül kapcsolatot

teremteni az általam lefejlesztett applikáció és a Whiteboard Team által fejlesztett szoftver között, tehát kommunikálnak egymással.

- A ProLifeManagement applikáció egy olyan felület legyen, ami biztosítsa a cégek számára a regisztrálást, a projektek létrehozását és hogy meghívják a projektekhez tartozó klienseket is a felületre. Ezáltal létre jöjjön egy “szóba” a cég és a kliens között, amely elősegíti a kommunikációt, a releváns dokumentumok, képek tárolását és az ügyfélszolgálati folyamatokat.
- A kommunikáció hírfolyam formájában történjen, a Facebook-hoz hasonlóan, számos funkcionalitás hiányával.
- Három felhasználói szerepkör és jogosultság meghatározása és kezelése, amelyek az admin, a fejlesztő és a kliens.
- Értesítési rendszer kialakítása, ami E-mail és push értesítések formájában történjen.
- Projektek és feladatok szűrésének a lehetősége.
- Egy web alapú alkalmazás fejlesztése annak érdekében, hogy bárki számára elérhető legyen, bármiféle letöltés igénye nélkül.
- A rendszer felosztása két nagy részből szól, éspedig egy cégek modul és egy kliens modul.

2. fejezet

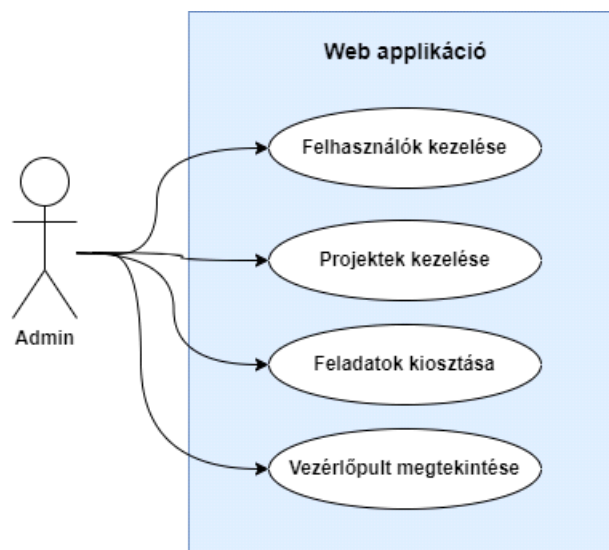
Rendszerspecifikáció

A ProLifeManagement alkalmazás célja a projekt munkafolyamatok racionalizálása, a feladatok nyomon követésének javítása és a csapatokon belüli együttműködés fokozása. A valós idejű értesítések, a feladatkezelés, a projekt létrehozása és szerkesztése, valamint a felhasználói adminisztráció az elsődleges funkciók közé tartoznak. Skálázhatósága, biztonsága és könnyű kezelhetősége miatt a rendszer kis és nagy csapatok igényeinek egyaránt megfelel.

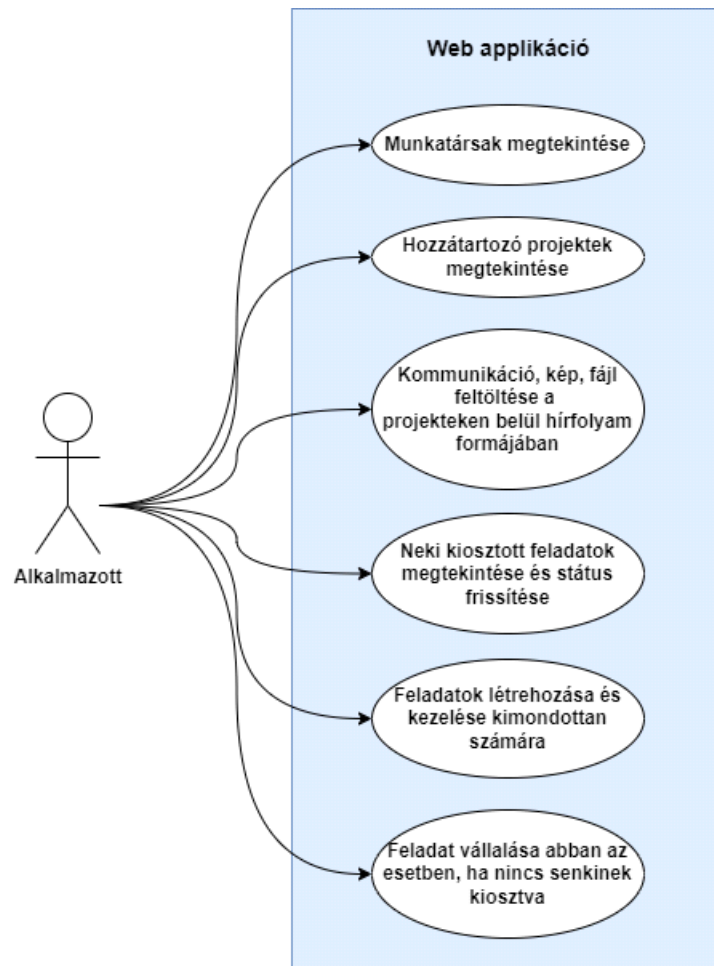
2.1. Felhasználói követelmények

Az applikáció úgy van megtervezve, hogy megfeleljen a vállalatokon belüli különböző felhasználói szerepkörök igényeinek. Három féle felhasználótípus létezik: admin, alkalmazott és kliens.

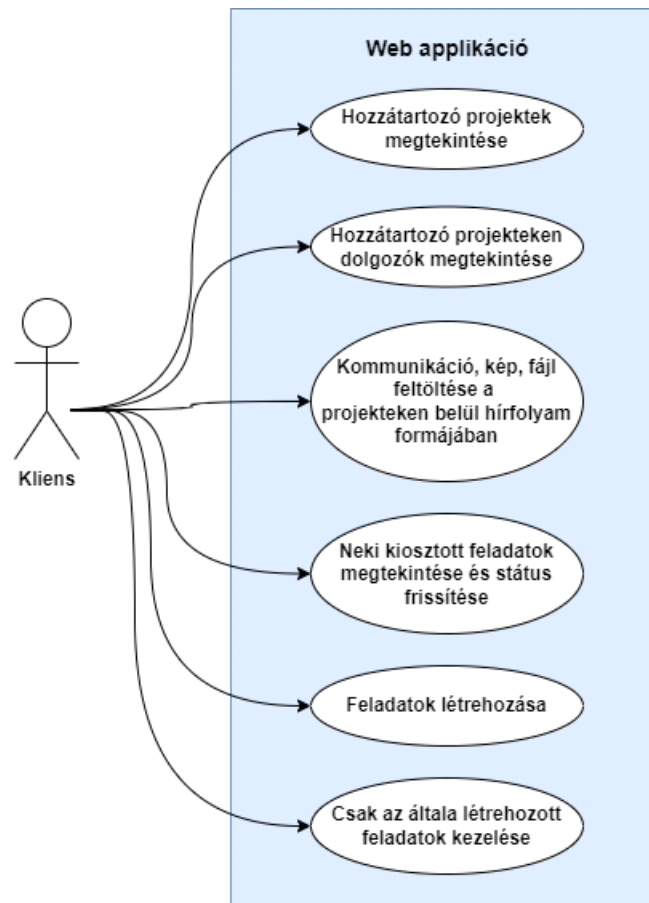
- Az admin felhasználó regisztráláskor jön létre . A regisztrálás csak cégek számára van engedélyezve, így alapértelmezetten az admin egy cégvezető. Ez a típusú felhasználó valójában CRUD(Create-Read-Update-Delete) jogosultsággal rendelkezik, ami a céget jelenti. Ezen belül gondolkodom a projektekre, projekt hozzászólásokra, fájlokra, feladatokra, alkalmazottakra. A létező Dashboard-on neki megjelennek a statisztikák a cégen belül végzett feladatokról, a projektek száma, az alkalmazottak száma, egy tevékenység lista a legrelevánsabb történetekkel.
- Az alkalmazott felhasználó csak úgy jöhet létre, ha az admin hozzá adja az alkalmazottak listájához. Ekkor egy felugró ablakban be kérem az alkalmazott email címét, amire küldök egy levelet. A levélben egy webcím szerepel, ami egy generált tokent tartalmaz GET paraméterként. A webcím a regisztrálás oldalra dobja át, ahol le van ellenőrizve a token érvényessége. A token 48 óra elteltével érvénytelen lesz. Ennek a típusú felhasználó is rendelkezik a CRUD jogosultságokkal, hasonlóképpen az adminhoz, annyi eltéréssel, hogy nem tudja a cég alkalmazottjait módosítani vagy törölni, csak megtekinteni az adataikat.
- A harmadik felhasználó a kliens, aki ismét egy emailen meghívón keresztül tud regisztrálni. Ő csak egy vagy több projekthez adható hozzá és számára csak azok a projektek léteznek az applikációban. Mindegyik projektnél létezik egy hírfolyam, ahol beszélgetéseket lehet indítani, dokumentumokat, képeket lehet feltölteni.



2. 1. ábra. Use Case diagram - Admin felhasználó



2. 2. ábra. Use Case diagram - Alkalmazott felhasználó



2. 3. ábra. Use Case diagram - Kliens felhasználó

2.1. Rendszerkövetelmények

a. Funkcionális követelmények:

(Felhasználókezelés, Projektkezelés, Feladatkezelés, Dashboard, Notifications)

A felhasználó sikeres bejelentkezése után, annak függvényében, hogy melyik szerepkörbe tartozik, különböző funkcionális követelmények lépnek előtérbe.

Admin szerepkörben, a felhasználó úralkodása alatt van minden felhasználó, legyen az alkalmazott vagy kliens, projekt, feladat, online fehér tábla kezelése, ami az általa létrehozott céghez tartoznak. A kezelés alatt pontosabban a CRUD (Create, Read, Update, Delete) műveletekre gondolok, melyek által létre tud hozni példa képpen egy felhasználót, vagy meghívni bizonyos projektekhez, módosítani tudja az adatait és szükség esetén törölni is tudja. Hasonló képpen cselekedhet az előbbi felsorolt, az applikációban szereplő elemekkel is. **(Továbbá az admin műszerfalán számos hasznos információk vannak kimutatva, mint például a bejelentkezett alkalmazottak száma, akik jelenleg**

dolgoznak, egy diagram az elmúlt időszakban bejött projektek számáról, hogy követhesse a cég eredményeit, akár új stratégiákat tudjon bevonni annak érdekében hogy a cég milyen irányban halad. Ezek mellett egy “Legutóbbi tevékenységek” lista is jelen van, ahol követni tudja a cégen belül zajló eseményeket, feladatokat. EZ MEGY A MEGVALÓSÍTÁSHOZ) Továbbá lehetősége van megtekinteni az összes feltöltött dokumentumot, amik lehetnek szerződések, arculati kézikönyv, árajánlat, számlák, számlaszám kivonatok és különböző csatolmányok, például hibák esetén egy képernyőkép feltöltése szemléltető képpen.

Alkalmazott(kolléga) szerepkörben, a felhasználó hasonló jogosultságokkal rendelkezik mint az admin, kis eltérésekkel. Nem képes a felhasználók és a projektek adatainak a módosítására, vagy törlésére. Csak azokat a projekteket látja, amelyekhez hozzá van adva, tehát dolgozik rajtuk és ezen belül pedig csak azokat a feladatokat, amelyekhez hozzá lett adva vagy amelyekhez nincs senki hozzárendelve. Továbbá csak azokat az online fehér táblákat látja, amelyekhez hozzá lett adva vagy ő maga hozott létre. (Az online táblák bármire alkamasak, nincsenek hozzákötve projektekhez, sem feladatokhoz. Bárminek a vázlatát, fontos személyes vagy céges adatokat le lehet menteni ezeken, rajzokat, táblázatokat, diagramokat lehet létrehozni, ami segít a projektek és feladatok megértésében és megoldásában. EZ MEGY A MEGVALÓSÍTÁSHOZ)

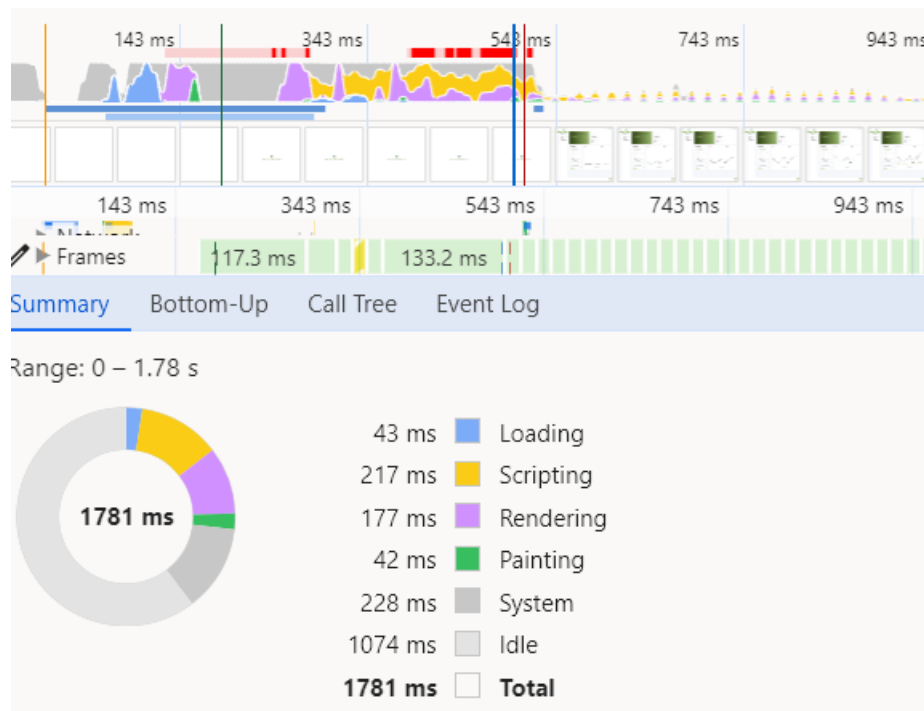
Kliens szerepkörben, a felhasználónak jóval korlátozottabb jogosultságai vannak az előző két felhasználó típushoz viszonyítva. Csak abban a projektben létező felhasználókat látja, amelyben ő is benne van. Itt a CRUD alapl műveletek csak a saját adataira vonatkoznak.

Az applikáció lényege és központja az a felhasználók közötti kommunikációra alapszik különböző módszerek által, legyen az feladat létrehozása, beszélgetés indítása vagy online tábla megosztása. Az implementált táblára tekintve van egy negatív észrevételem, éspedig az, hogy a rajzolt vagy illesztett elemek nem jelennek meg a többi felhasználónál, csak az oldal frissítését követően.

b. Nem funkcionális követelmények:

(Biztonság - Titkosított jelszavak, Role-based access control (RBAC) - Szerepkör-alapú hozzáférés)

Az új technológiákat és az internet felhasználók elvárásait figyelembe véve, egy webapplikációnak legtöbb 3 másodperc alatt be kell töltsse az oldalakat. Minden másodperc ami ezt meghaladja, 16%-al növeli az esélyt arra, hogy a felhasználó elhadj a weboldalt. Ebből kifolyólag szerintem egy webapplikáció gyorsasága ennél is nagyobb kell legyen.



2.4. ábra. Webapplikáció sebessége

Jelenlegi helyzetben, az applikáció betöltésének az összsebessége 1.7mp, de ami ennél is fontosabb, hogy az FCP (First contentful Paint/első vizuális tartalom) 0.6mp alatt tölt be.

A felhasználó bejelentkezése után, minden adatot, mint például a neve, telefon száma, e-mail, le mentem egy munkamenetbe, PHP nyelvben ez a `$_SESSION`. Ezzel még inkább gyorsítom az applikációt azzal, hogy nem kell minden oldal betöltésekor lekérni az adatbázisból az ID-t. Ezzel a módszerrel az RBAC(Role-Based Access Control) kezelése is könnyebben történik, így biztosítani tudom azt, hogy a felhasználók csak azokhoz az adatokhoz és funkciókhoz férjenek hozzá, amelyekhez jogosultságuk van.

A felhasználók navigálását az applikációban minél könnyebbnek és egyszerűbbnek próbáltam megvalósítani. A bal oldalsó menüben minden fontosabb oldalt elérhetővé tettem, ahonnan szintén beljebb áthatja magát a rendszerben, más-más releváns oldalakra navigálva. Integrálva van az e-mail küldés, ami a Mailtrap[8] e-mail kézbesítési platformon keresztül történik jelenleg, és a Toast[9] nevű értesítési könyvtár, ami egy kis felugró ablak formájában jelenik meg szöveggel a jobb felső sarokban néhány másodpercig. Ezt arra használom, hogy a különböző műveletek elvégzésére egy visszajelzést kapjon a felhasználó arról, hogy sikeres vagy sikertelen lett a művelet eredménye.

Az fejlesztés során törekedtem arra, hogy egy moduláris rendszert hozzak létre, ezért is használtam az MVC programtervezési mintát. Az ProLifeManagement applikáció különböző komponensei és szegmensei, így könnyen lecserélhetőek vagy módosíthatóak az elkövetkező fejlesztések vagy hibák észrevétele esetében.

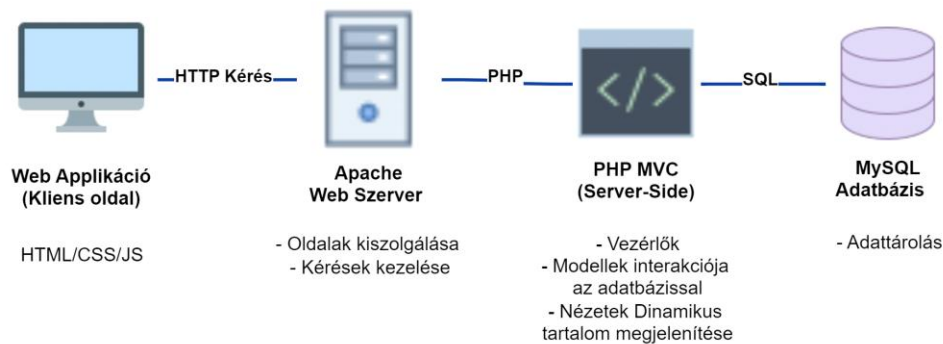
Mivel egy webes applikációról van szó, ezért különös figyelmet kellett szánjak a különböző eszközökön való használatára, mint az asztali gépek vagy laptopok, táblagépek és okostelefonok. Nagy segítségemre volt itt a Bootstrap frontend fejlesztési rendszer. Használtam a konténer osztályt, amely által limitáltam az elemeket tartalmazó konténer maximális szélességét 1320px-re, azért hogy az óriási képernyőkön ne legyen elcsúszás, így mindent egységessé tudtam tenni. Mindemelett hasznát vettem az 5 definiált képernyőszélességnek, amelyek az extra kicsi(xs) amely az 576px-nél kisebb képernyőkre vonatkozik, a kicsi(sm), amely 576px és 767px között érvényes, a közepes(md), amely 768px és 991px között érvényes, a nagy(lg), 992px és 1119px között, az extra nagy(xl), 1200px és 1399px, valamint extra extra nagy(xxl), amely 1400px-nél nagyobb képernyőkön lép érvénybe. Ezek segítségével, az elemeket képernyőméret függővé tehettem, pozíciókat, méreteket változtatva, eltüntetni akár őket.

A Bootstrap támogatja a böngészők és platformok legújabb stabil verziókat, ha Internet Explorer böngészőn szeretnénk használni az applikációt, figyeljünk arra, hogy ne használjunk a 10-es verziónál régebbi kiadást.

3. fejezet

Rendszerarchitektúra

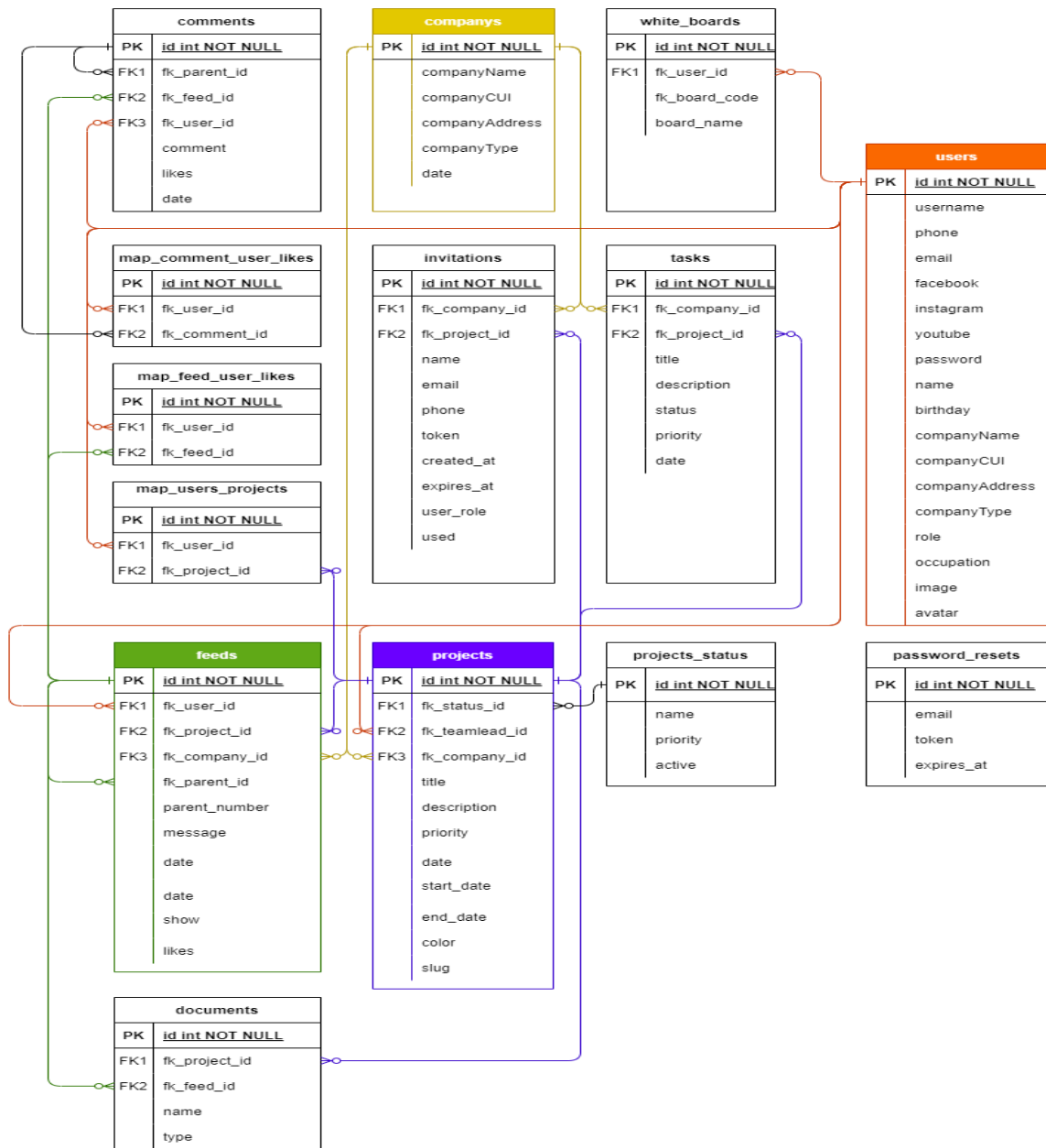
A projektmenedzsment-alkalmazás teljes funkcionalitását több komponens biztosítja, amelyek együtt egy rendszerként működnek. Minden komponens kritikus szerepet játszik az alkalmazás használhatóságának, teljesítményének és karbantarthatóságának biztosításában. Ezek az elemek a következőkből állnak:



3. 1. ábra. Teljes rendszer architektúrája

- **Adatbázis:**

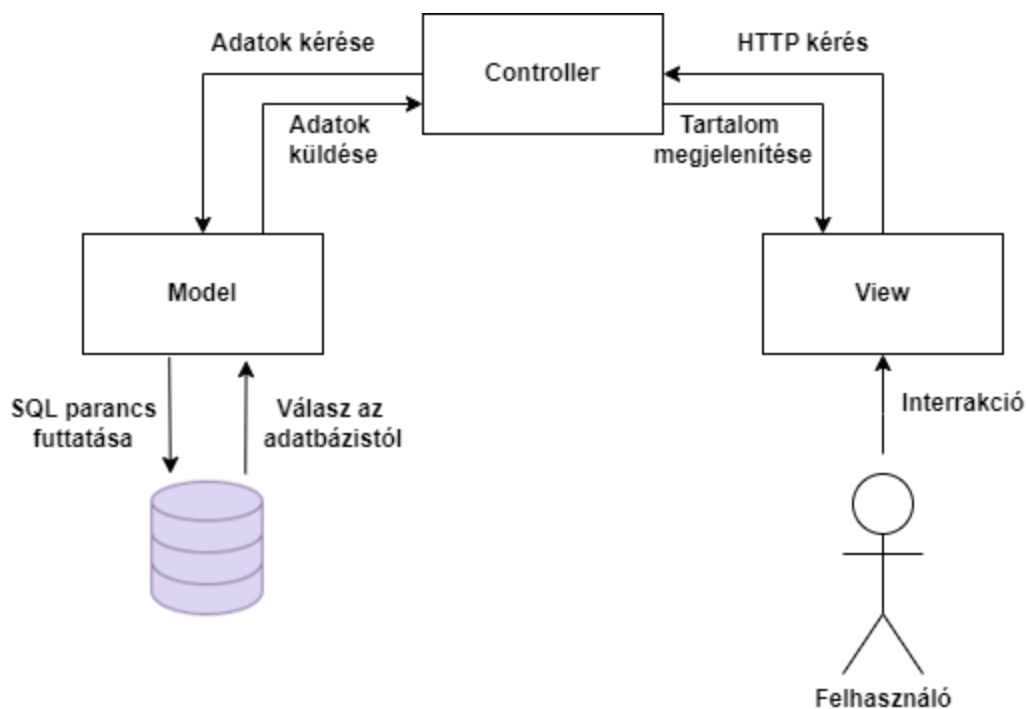
A adatbázis az összes tartós adat tárolásáért felelős, beleértve a felhasználói információkat, projekteket, feladatokat és tevékenységeket. A hozzáférés a Modell rétegben történik, amely kapcsolatba lép az adatbázissal a PDO-n (PHP Data Objects) keresztül végrehajtott SQL-lekérdezések segítségével. MySQL adatbázissal dolgozok, mivel megengedi azt, hogy komplex lekérdezésekhez és adatmanipulációhoz használjam az SQL-t. A PDO használatával biztonságosabbá sikerül tenni a rendszert az SQL-injekciós támadások megelőzésével. Az adatbázis jobb átláthatóságáért és könnyebb kezeléséért a PhpMyAdmin grafikus felületet használom, ami lehetővé teszi a táblák egyszerű létrehozását, SQL lekérdezések tesztelését.



3. 2. ábra. ER-modell(egyed-kapcsolat modell)

- Szerveroldali alkalmazás**

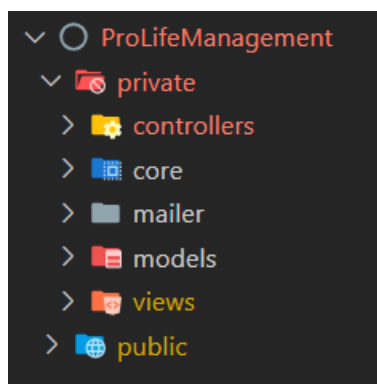
Az applikáció backend részét PHP programozási nyelven írtam meg az MVC (Model – View – Controller) architektúrát követve(3.3. és 3.4. ábrák.), hogy a rendszer átláthatóbb legyen, karbantarthatóbb és könnyebb legyen a további fejlesztése.



3. 3. ábra. MVC architektúra

Ezek mellett fontos megemlíteni, hogy sokat segít a tesztelés fázisában is, mivel a komponensek külön tesztelhetők, ezért egyszerűbb a hibák megtalálása és javítása.

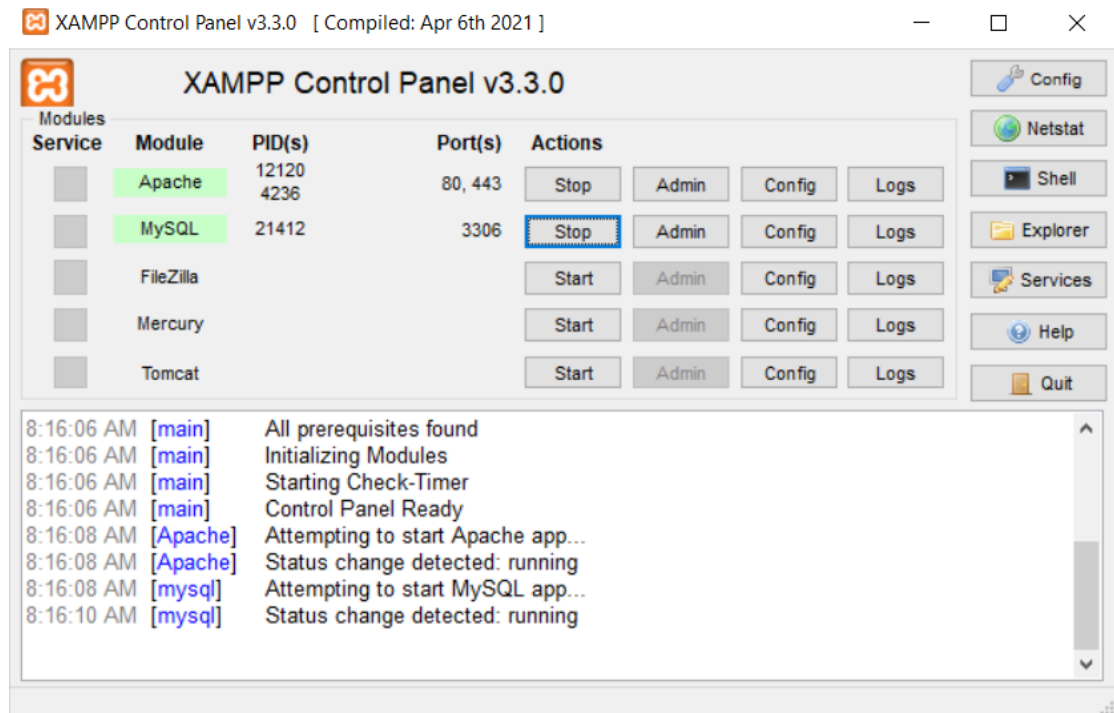
A modellréteg feladata, hogy kapcsolatot létesítsen a MySQL-adatbázissal. A View réteg felelős az adatok megjelenítéséért a felhasználó számára. Beágyazott PHP kódokat tartalmaz, amelyek a vezérlőtől kapott adatok alapján generálnak dinamikus HTML tartalmakat. A vezérlő rétegben kezelem le a bejövő kéréseket, majd kapcsolatot létesítek a modellel, annak érdekében, hogy adatok kérjek le, frissítsek vagy töröljek az adatbázisból, majd elhatározom hogy melyik nézet jelenjen meg.



3. 4. ábra. Projekt struktúrája

- **Web Szerver**

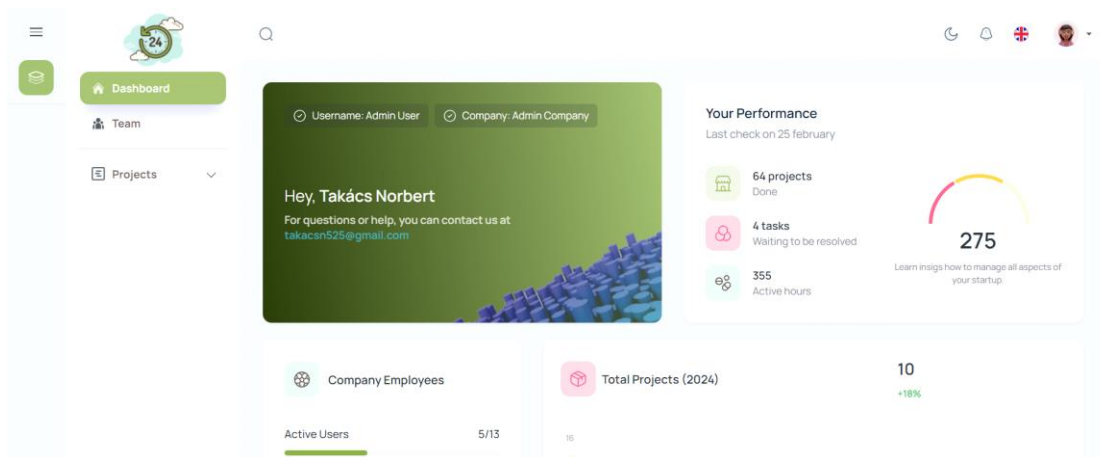
A XAMPP-ben található Apache webszerver kezeli a kliensek (a webböngésző felhasználók) HTTP(Hypertext Transfer Protocol)-kérelmeit, és kiszolgálja az alkalmazás weboldalait. Feldolgozza a PHP szkripteket, és kommunikál az adatbázissal az adatok lekérdezése és tárolása érdekében.



3. 5. ábra. A XAMPP vezérlőfelület

- **Felhasználó interfész**

A frontend felület HTML, CSS és JavaScript segítségével készült. A szerveroldali alkalmazással együttműködve jeleníti meg a felhasználó számára az adatokat és rögzíti a felhasználói adatokat. Az ügyféloldali keretrendszerek és könyvtárak, mint például a Bootstrap és a jQuery fokozzák a felhasználói élményt.



3. 6. ábra. Applikáció - Kezdőoldal

Team

Search Team Members...

Invite Member

<input type="checkbox"/>	Name	Status	Email	Phone	Action
<input type="checkbox"/>	Waiting for confirmation	offline	takacs555@gmail.com		
<input type="checkbox"/>	Takacs Norbertt Sador Web dev	online	employee@gmail.com	0765436992	
<input type="checkbox"/>	Takács Norbert Admin(Me)	online	admin@gmail.com	0749546757	

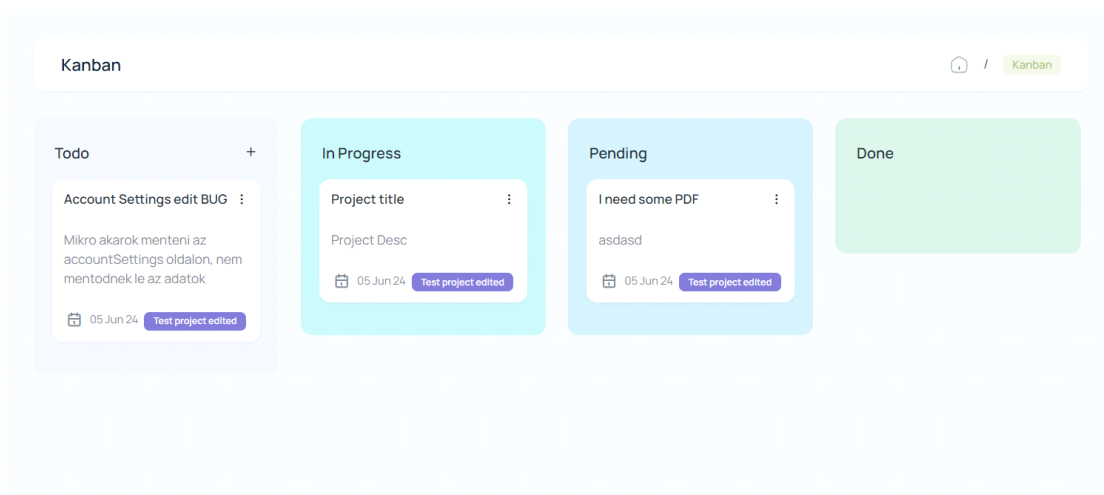
3. 7. ábra. Applikáció - Csapat beállítások

Projects

Create Project All Started Ready Approval Completed

Test project edited	Real Project	Bemuato projekt
Short description(static)	Short description(static)	Short description(static)
5 Attach 2 Months 2 d	5 Attach 2 Months 19 d	5 Attach 0 Months 13 d
1 Members 10	2 Members 10	4 Members 10
Elapsed time: 52 days left	Elapsed time: 74 days left	Elapsed time: 10 days left

3. 8. ábra. Applikáció - Projektek oldal



3. 9. ábra. Applikáció - Kanban feladatok oldal


4. fejezet

Megvalósítás

4.1. ProLifeManagemenet, Webes applikáció (Ebben a részben le szeretném írni az applikációban létező folyamatokat, bejelentkezés, regisztrálás, projekt létrehozása, taskek menedzselése, csapat létrehozása, e-mailek küldésének a folyamata és működése, third-party tool-ok bemutatása és előnyeik)

Az alkalmazás megnyitásakor egyből a login oldal jelenik meg a felhasználó számára(4.2. ábra.). Ez azért szükséges, hogy tudjam meghatározni a felhasználó szerepkörét és aszerint kezeljem az adatokat és a felhasználói felületet. A bejelentkezés történhet email vagy felhasználó névvel és a hozzá tartozó jelszóval. Sikertelen próbálkozás esetén egy hibaüzenetet jelenítek meg, sikeres bejelentkezés esetén a főoldalra irányítom át a felhasználót, ahol számára releváns adatokkal fog találkozni(4.3. ábra.). A bejelentkezési oldalról közvetlenül át tudunk navigálni a regisztrálás oldalra(4.1. ábra.), ahol csakis a cégek tudnak új admin felhasználót létrehozni. Létrehozása előtt megvizsgálom, hogy minden céges adat meglegyen adva, így ezek kötelező bemeneti mezőként szerepelnek az űrlapban. Ilyen adatok a cég neve, a CUI cégjegyzékszámot, a címet, a jogi személy típusát(4.1. ábra.). Ezek mellett minden olyan adatra szükség van, amire kitérek a

továbbiakban a kliens részénél. Amíg nincs megadva minden kötelező bemeneti adat, hibaüzenetekkel térnek vissza a kliens felé.



Sign Up

Company Name

Only letters allowed in company name

Unique identification code (CUI/CIF)

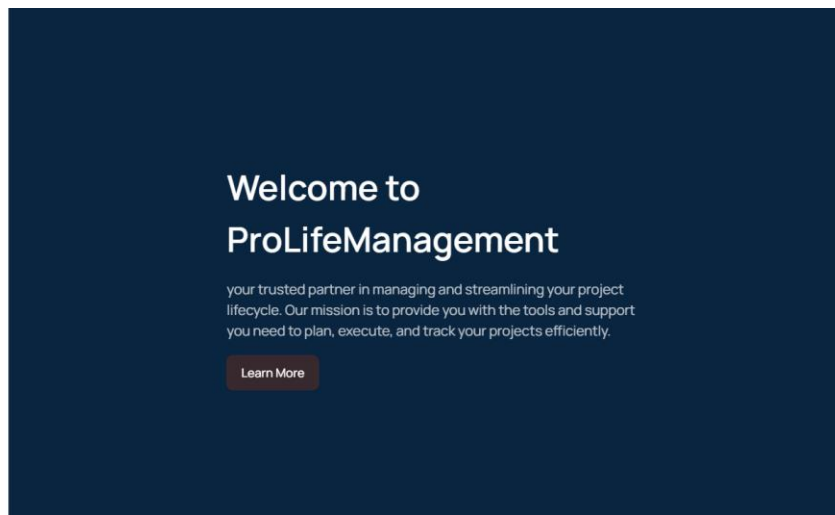
You have entered the wrong CUI/CIF

Company Address


You must give us the address of your company

Type of legal entity (SRL, PFA, etc.)

You must give us the type of your company



4.1. ábra. Applikáció - Cégek regisztrálása oldal.



Sign In

Your Admin Dashboard

Email or Username

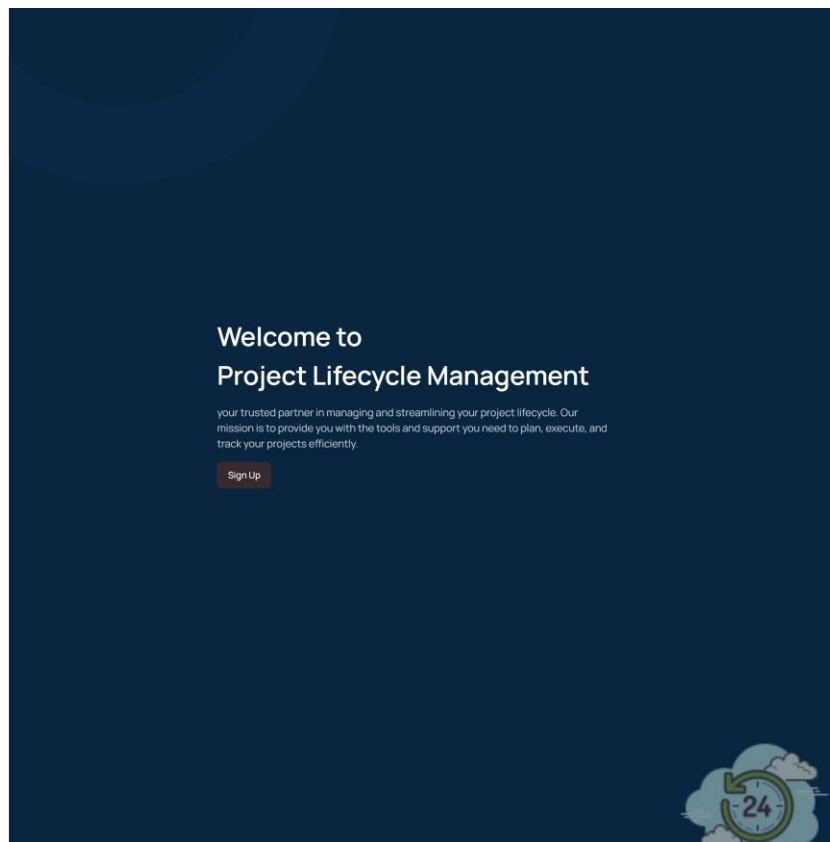
Admin User

Password

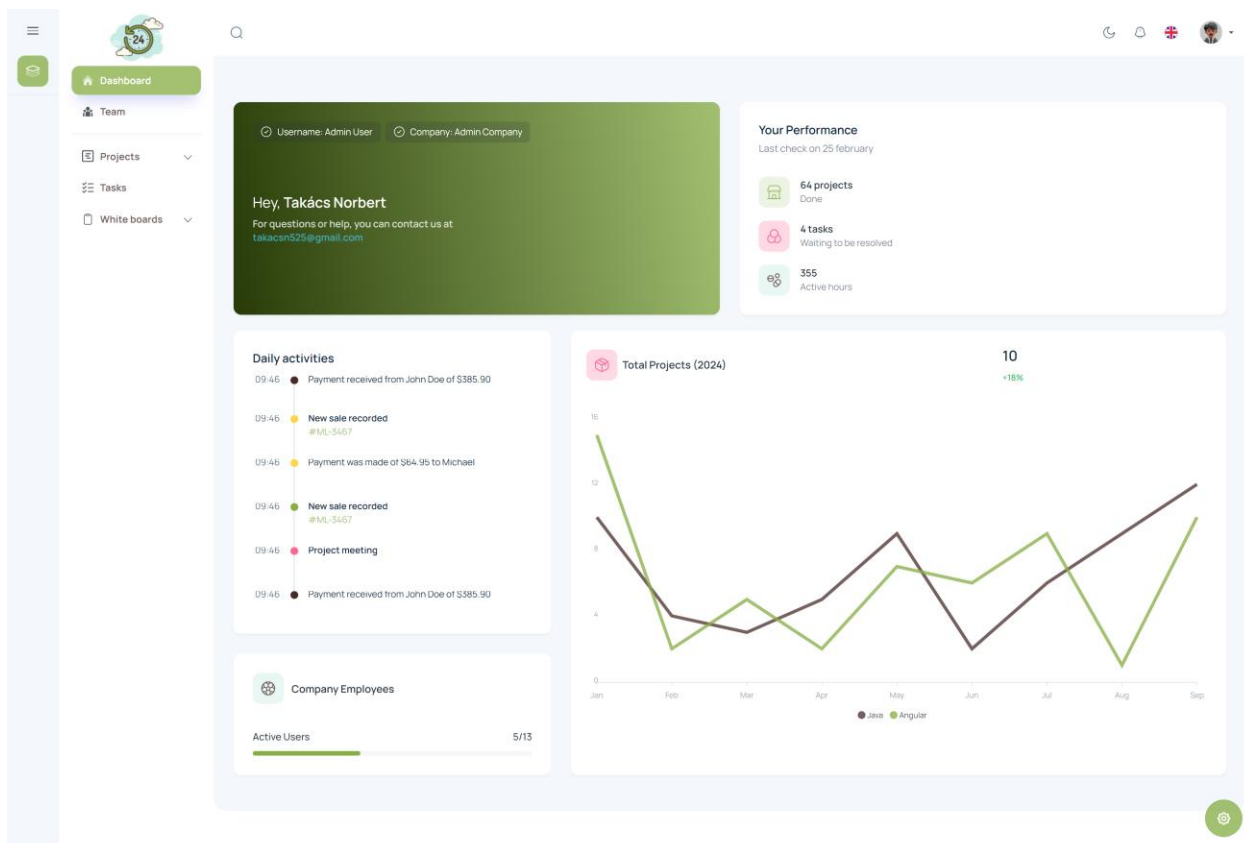
☐ Remember this Device [Forgot Password?](#)

[Sign In](#)

New to ProLifeManagement? [Create an account](#)



4.2. ábra. Applikáció - Bejelentkezés oldal



4.3. ábra. Applikáció - Admin, bejelentkezés utáni főoldal.

Az összes regisztrálási űrlapban több adatot igényelek, különösen olyan adatokat, amelyekkel kötelezően rendelkeznie kell egy felhasználónak, mint például egy felhasználónév, teljes név, születési dátum, egy email cím, telefonszám, opcionálisan az Instagram, Facebook vagy YouTube linkek, egy profilkép és/vagy egy avatar kiválasztása és természetesen a jelszó. Az email cím, a telefonszám és a jelszó különös képpen le vannak validálva. Mindegyikük egyediéeknek kell hogy legyenek, és be kell tartaniuk a formátumokat, telefonszám legkevesebb 10, legtöbb 14 számjegyből állhat és azért, hogy a felhasználó kisebb valószínűséggel írja el a jelszavát, a két jelszónak meg kell egyeznie. Minden adat a MySQL relációs adatbázisban van lementve, a "Users" nevű táblában. A jelszó titkosítása érdekében, ennek a hash értéke van eltárolva, amit a PHP integrált "password_hash" függvénnyel generáltatok, ami egy 60 karakterből álló karakterláncot hoz létre. Ezzel az eljárással jelentősen növelem a jelszó biztonságát, mivel ha az adatbázist feltörik, a tárolt értékből nem lehet egyszerűen visszaállítani a jelszavakat.

A bcrypt hash-nek van egy specifikus formátuma:

- **Mágikus string:** "\$2y\$" vagy "\$2b\$" előtag, amely azt jelzi, hogy ez egy bcrypt hash-ről van szó.
- **Cost paraméter:** "10\$" vagy ennél nagyobb szám, mondjuk azt hogy "n", amely a hash-elés költségét jelenti. Pontosabban az iterációk számát, ami 2^n -iken. Minél nagyobb cost faktort adunk meg, annál több időt és erőforrást igényel az algoritmus.
- **Salt:** véletlenszerűen generált, 22 karakter hosszú rész. Emiatt lehetséges az is, hogy két felhasználó ugyanazt a jelszót választhatja, de az adatbázisban ez két külön értéket fog jelenteni.
- **Hash:** a tényleges hash érték, amely a jelszó és a salt kombinációjából jön létre az iterációk során.

4.2. Javascriptek használata (Szekvencia diagrammal példázom, leírom a POST request folyamatokat anélkül hogy a weboldálnak frissítenie keljen)

4.3. Tesztelések

4.4. Projektmenedzsment (Git-ről írok)

5. fejezet

Összefoglaló

5.1 Következtetések (Mit is sikerült megvalósítani, leírom röviden az elért funkcionalitásokat)

5.2. Fejlesztési lehetőségek (Milyen ötleteim támadtak a fejlesztés során, amiket a további fejlesztés során fogok implementálni)

5.3. Lehetséges felhasználási terület

Irodalomjegyzék

- [1] PHP előnyök és hátrányok. <https://anywhere.epam.com/en/blog/pros-and-cons-of-php>
- [2] HTML előnyök és hátrányok. <https://unstop.com/blog/advantages-and-disadvantages-of-html>
- [3] CSS előnyök és hátrányok. <https://unstop.com/blog/advantages-and-disadvantages-of-css>
- [4] Bootstrap előnyök és hátrányok. <https://www.bootstrapdash.com/blog/pros-cons-using-bootstrap-web-development>
- [5] JavaScript előnyök és hátrányok. <https://unstop.com/blog/advantages-and-disadvantages-of-javascript>
- [6] MySql előnyök. <https://www.linkedin.com/pulse/importance-mysql-database-advantages-disadvantages-influence-bk-xhfce/>
- [7] MySql hátrányok. <https://unstop.com/blog/advantages-and-disadvantages-of-sql>
- [8] Mailtrap e-mail kézbesítési platform. <https://mailtrap.io/>
- [9] Toast értesítési könyvtár. <https://apvarun.github.io/toastify-js/>
- [10]
- [11]
- [12]