

**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 9

1.1. Témaválasztás indoklása .....	9
Központosított Adatkezelés .....	9
Jobb Kommunikáció és Együttműködés .....	9
Biztonság és Hozzáférés Szabályozás .....	10
Hatékonyabb Dokumentumkezelés .....	10
Idő- és Erőforrás Megtakarítás .....	10
1.2. Szakirodalmi áttekintő .....	10
1.3. Kutatási kérdések .....	10
Backend.....	11
Frontend.....	11
• JavaScript.....	13
Adatbázis.....	13
• MySQL .....	13
Version Control Systems (VCS) .....	14
• GitHub .....	14
1.4. Célkitűzések .....	14
Rendszerspecifikáció .....	14
2.1. Felhasználói követelmények .....	14
2.1. Rendszerkövetelmények .....	17
a. Funkcionális követelmények:.....	17
b. Nem funkcionális követelmények: .....	17
Rendszerarchitektúra .....	17
• Adatbázis:.....	18
• Szerveroldali alkalmazás .....	19
• Web Szerver .....	20
• Felhasználó interfész .....	21
Megvalósítás .....	22



4.1. ProLifeManagemenet Applikáció .....	22
4.2. Javascriptek használata .....	23
4.3. Tesztelések .....	23
4.4. Projektmenedzsment .....	23
Összefoglaló .....	23
5.1 Következtetések .....	23
5.2. Fejlesztési lehetőségek .....	23
5.3. Lehetséges felhasználási terület .....	23
Irodalomjegyzék .....	23

# 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 gyorsítótárazás, 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:

## Backend

- **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. és újabb) kiadások még a korábbiaknál is gyorsabbak.  
(<https://anywhere.epam.com/en/blog/pros-and-cons-of-php>)

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. (<https://anywhere.epam.com/en/blog/pros-and-cons-of-php>)

Alternatív opciók: Python, Java, Node.js

## Frontend

- **HTML5:** Alapvető technológiák a weboldalak szerkezetének és megjelenésének meghatározásához.

#### 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. (<https://unstop.com/blog/advantages-and-disadvantages-of-html>)
- 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. (<https://unstop.com/blog/advantages-and-disadvantages-of-html>)

#### 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. (<https://unstop.com/blog/advantages-and-disadvantages-of-html>)
  - 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. (<https://unstop.com/blog/advantages-and-disadvantages-of-html>)
- **CSS3:** Stílusozás 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.  
(<https://unstop.com/blog/advantages-and-disadvantages-of-css>)
- 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. (<https://unstop.com/blog/advantages-and-disadvantages-of-css>)

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.  
(<https://unstop.com/blog/advantages-and-disadvantages-of-css>)

- **CSS Frameworks: Bootstrap**
- **JavaScript:** Az interaktív elemek megvalósításához használt alapvető programozási nyelv a webfejlesztés során.

Előnyök:

-

Hátrányok:

-

Alternatív opciók: React, Vue.js, Angular

## Adatbázis

- **MySQL:** A legelterjedtebb relációs adatbázis-kezelő rendszer, amelyet a CMS rendszerek széles körben alkalmaznak.

Előnyök:

-

Hátrányok:

-

Alternatív opciók: MongoDB, PostgreSQL

## Version Control Systems (VCS)

- **GitHub:** A forráskód verziókezelése és nyomon követése.

### 1.4. Célkitűzések

## 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 tartozik. Skálázhatósága, biztonsága és könnyű kezelhetősége miatt a rendszer kis és nagy csapatok igényeinek egyaránt megfelel.

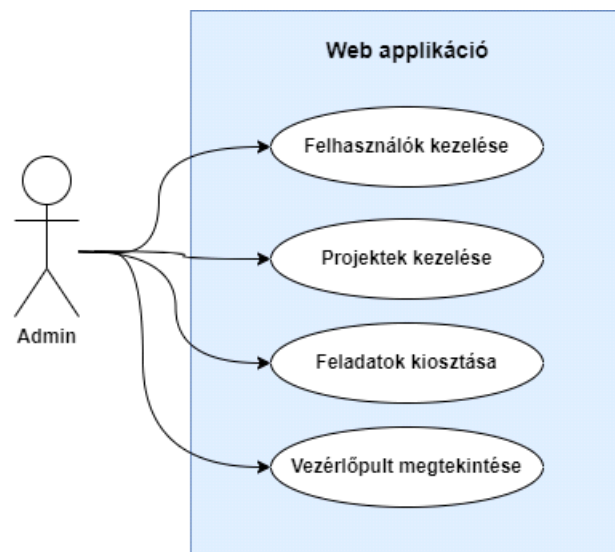
### 2.1. Felhasználói követelmények

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

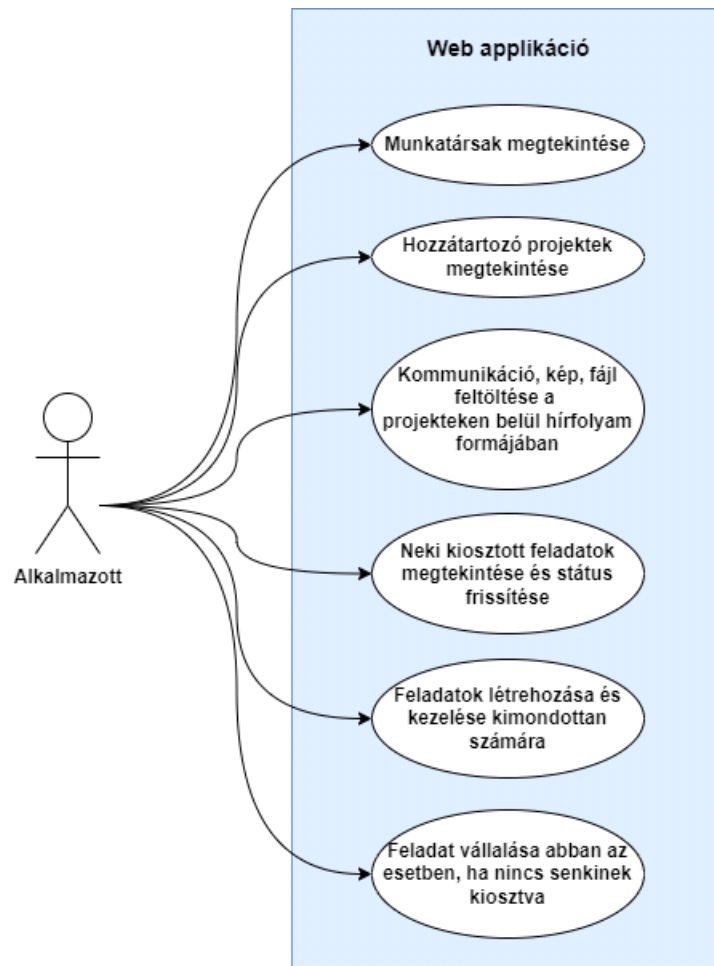
- Az admin felhasználó regisztráláskor jön létre. A regisztrálás csak cégek számára van engedélyezve, így alapértelmezetten az admin egy cégvezető. Ez a típusú felhasználó valójában CRUD(Create-Read-Update-Delete) jogosultsággal rendelkezik, ami a céget jelenti. Ezen belül gondolok a projektekre, projekt 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 email-es 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.

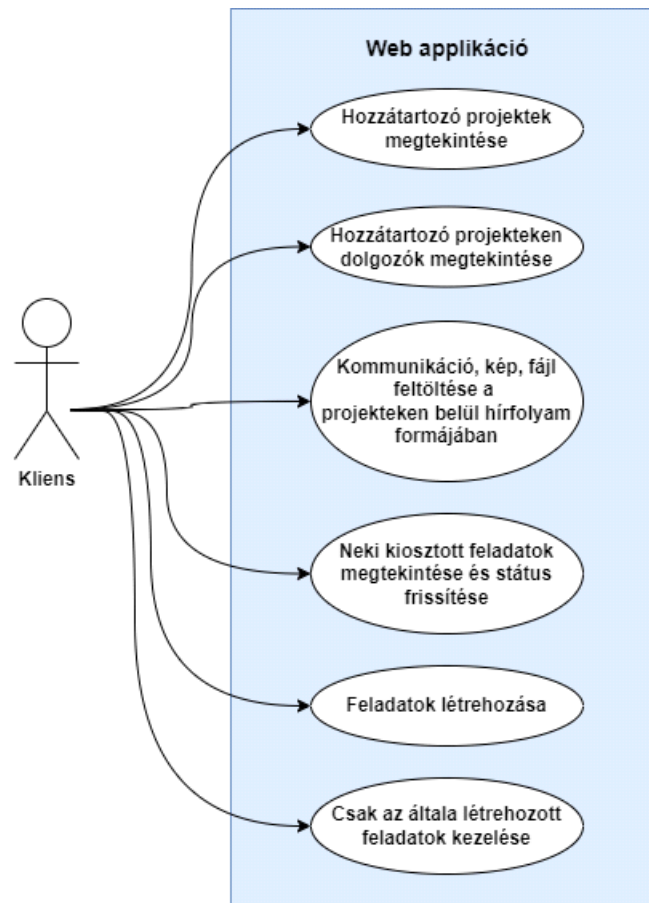


**2. 1. ábra.** Use Case diagram - Admin felhasználó



**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)

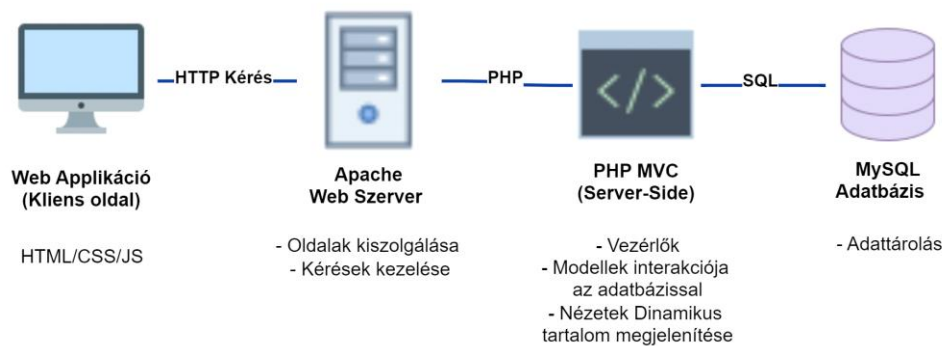
### b. Nem funkcionális követelmények:

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

# 3. fejezet

## Rendszerarchitektúra

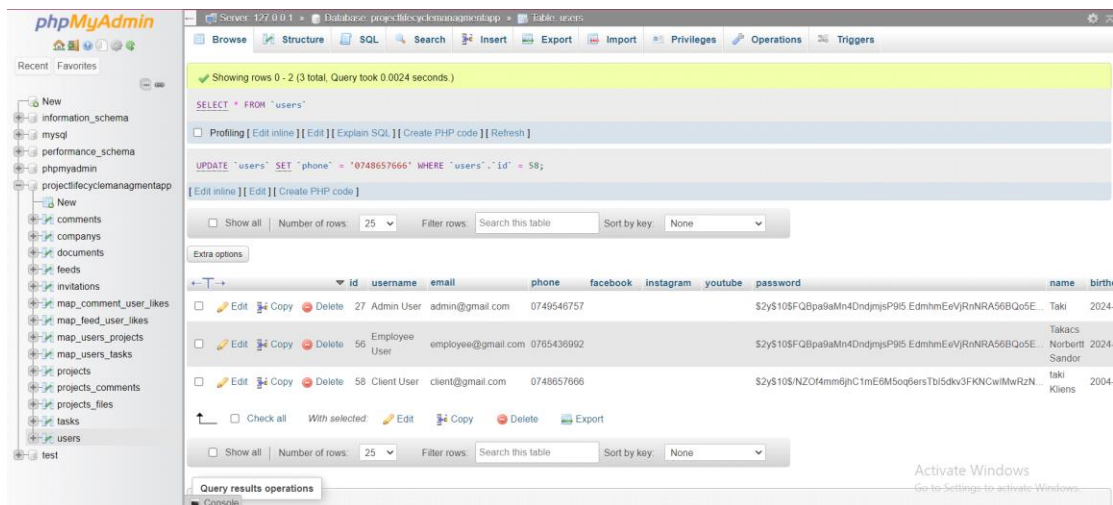
A projektmenedzsment-alkalmazás teljes funkcionalitását több komponens biztosítja, amelyek együtt egy rendszerként működnek. Minden komponens kritikus szerepet játszik az alkalmazás használhatóságának, teljesítményének és karbantarthatóságának biztosításában. Ezek az elemek a következőkből állnak:



**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 használlok, ami lehetővé teszi a táblák egyszerű létrehozását, SQL lekérdezések tesztelését.

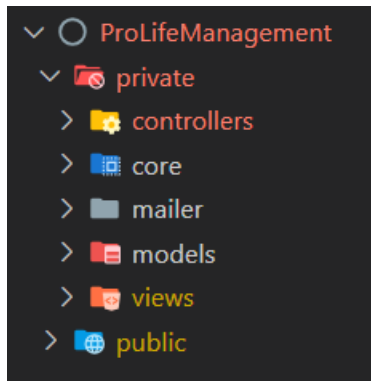


3. 2. ábra. PhpMyAdmin grafikus felület, az adatbázisban látező 'users' tábla

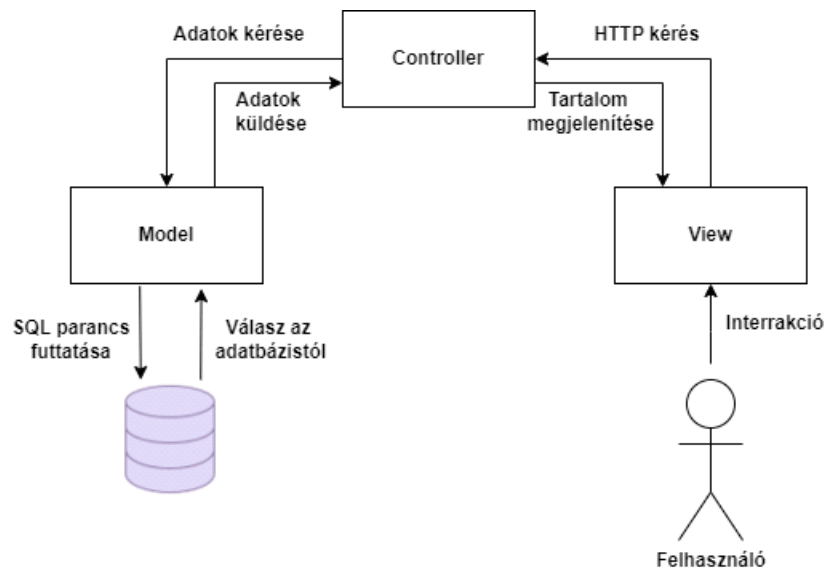
### • Szerveroldali alkalmazás

Az applikáció backend részét PHP programozási nyelven írtam meg az MVC mintát követve, ami javítja a karbantarthatóságot.

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.



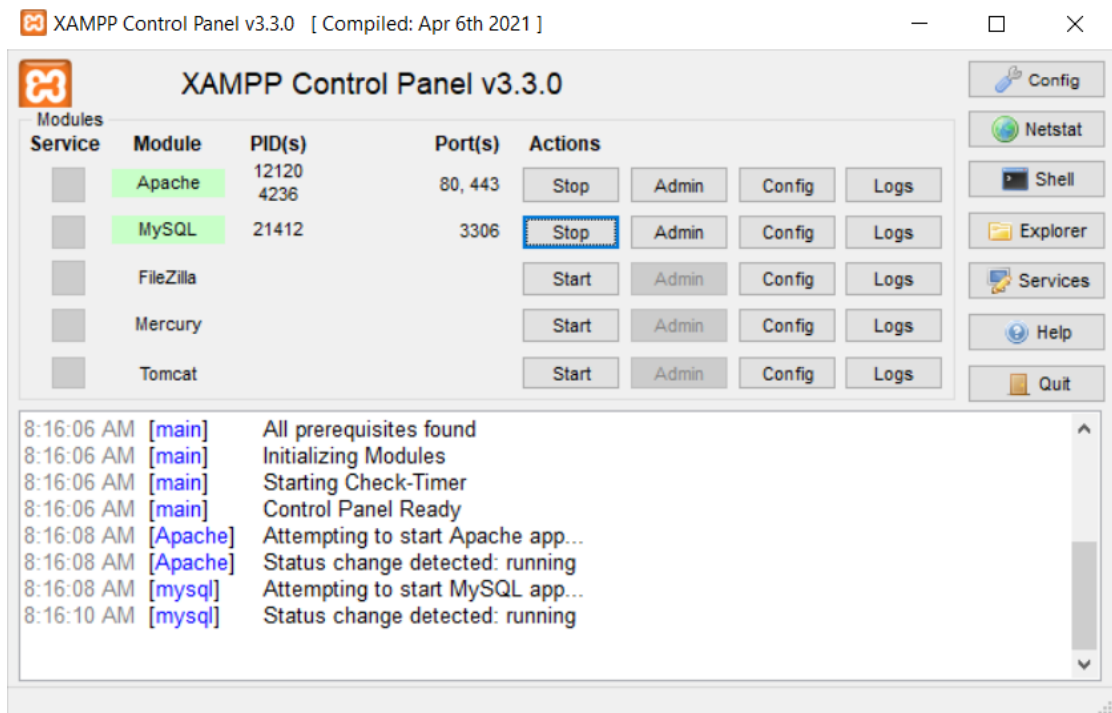
3. 3. ábra. Projekt struktúrája



3. 4. ábra. MVC architektúra

- **Web Szerver**

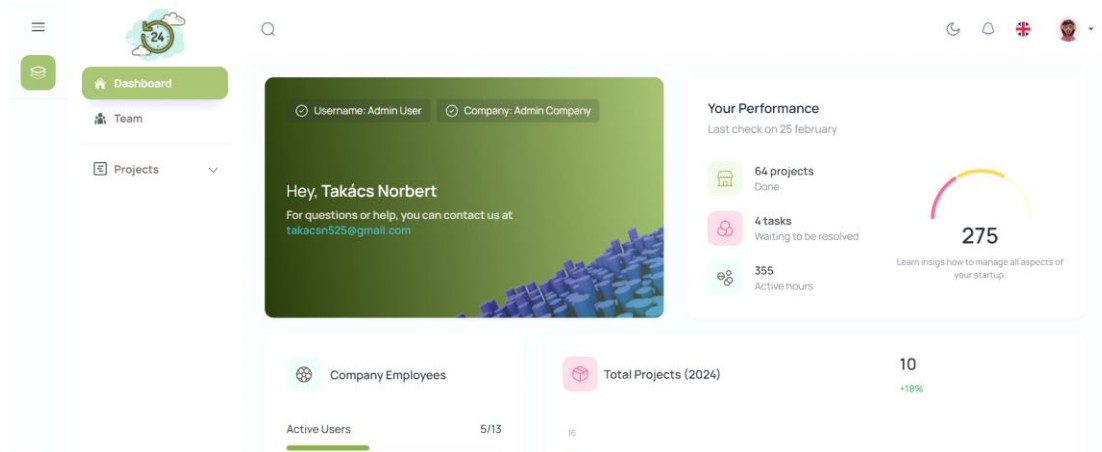
A XAMPP-ben található Apache webszerver kezeli a kliensek (a webböngésző felhasználók) HTTP-kérelmeit, és kiszolgálja az alkalmazás weboldalait. Feldolgozza a PHP szkripteket, és kommunikál az adatbázissal az adatok lekérdezése és tárolása érdekében.



### 3. 5. ábra. A XAMPP vezérlőfelület

- **Felhasználó interfész**

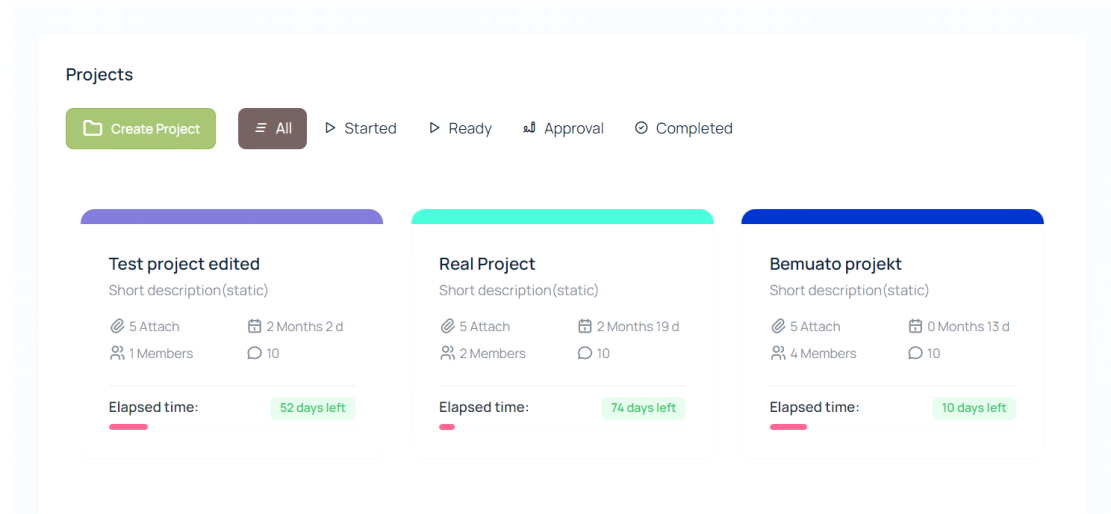
A frontend felület HTML, CSS és JavaScript segítségével készült. A szerveroldali alkalmazással együttműködve jeleníti meg a felhasználó számára az adatokat és rögzíti a felhasználói adatokat. Az ügyféloldali keretrendszerek és könyvtárak, mint például a Bootstrap és a jQuery fokozzák a felhasználói élményt.



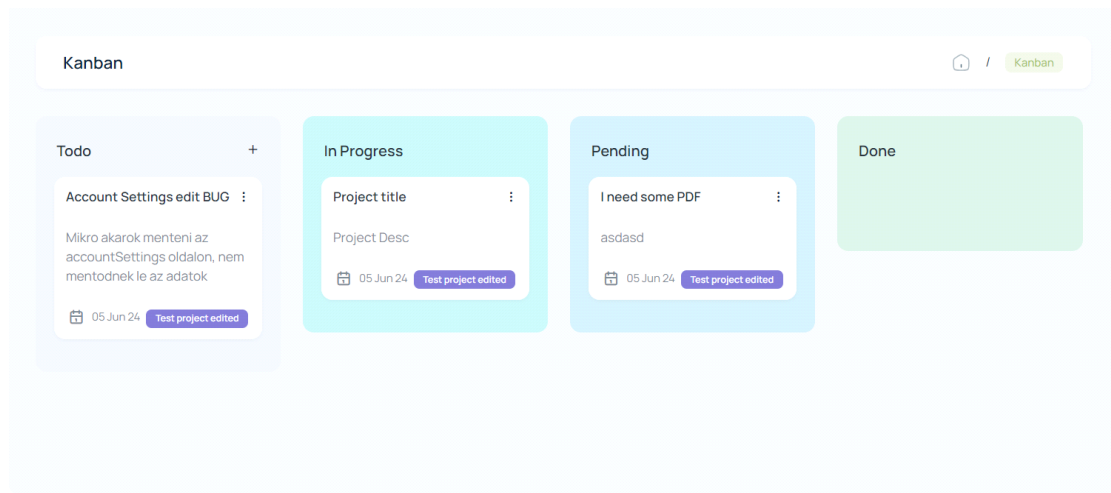
### 3. 6. ábra. Applikáció - Kezdőoldal

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

### 3. 7. ábra. Applikáció - Csapat beállítások



3. 8. ábra. Applikáció - Projektek oldal



3. 9. ábra. Applikáció - Kanban feladatok oldal

## 4. fejezet

# Megvalósítás

4.1. ProLifeManagemenet Applikáció (Ebben a részben le szeretném írni az applikációban létező folyamatokat, bejelentkezés, regisztrálás, projekt

létrehozása, taskok menedzselése, csapat létrehozása, e-mailek küldésének a folyamata és működése, third-party tool-ok bemutatása és előnyeik)

**4.2. Javascriptek használata** (Szekvencia diagrammal példázom, leírom a POST request folyamatokat anélkül hogy a weboldalnak 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