

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:
Dr. Iclănzan David Andrei
Profesor universitar

Absolvent:
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

Aprobat,
Decan

DOMNULE DECAN

Subsemnatul/a,
TAKACS NORBERT SÁNDOR
absolvent(ă) al/a Facultății de Științe Tehnice și Umaniste din Târgu Mureș
domeniul CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
specializarea CALCULATOARE
promoția 202, vă rog să-mi aprobați înscrierea la examenul de
licență/diplomă din sesiunea iulie 2024.

Am mai susținut examenul de licență în anul

Am întocmit lucrarea de licență / proiectul de diplomă cu titlul
APLICAȚIE PENTRU GESTIONAREA CICLULUI DE VIAȚĂ AL
PROIECTELOR

sub îndrumarea dr. ICLĂNZAN DAVIDA

Anexez la cerere Lucrarea de licență / Proiectul de diplomă în format electronic, referatul
de evaluare al îndrumătorului lucrării/proiectului, respectiv declarația privind
originalitatea lucrării.

Declar că am înțeles și am luat la cunoștință conținutul notificării privind prelucrarea
datelor cu caracter personal în cadrul Universității Sapientia și sunt de acord cu
prelucrarea datelor mele cu caracter personal conform celor cuprinse în informare.

Data: 24.06.2024

Semnătura

Viza îndrumătorului lucrării/proiectului.....

Viză secretariat,

Informare privind prelucrarea datelor cu caracter personal

UNIVERSITATEA SAPIENTIA, cu sediul în Cluj-Napoca, str. Matei Corvin nr. 4, jud. Cluj, având CIF RO14645945, numită în continuare Universitate, este operator de date cu caracter personal cu privire la datele dvs. personale furnizate instituției noastre prin Fișa de înscriere la concursul de admitere și prin documentele anexate acestuia, conform dispozițiilor Regulamentului UE nr. 2016/679 al Parlamentului European și al Consiliului privind protecția persoanelor fizice în ceea ce privește prelucrarea datelor cu caracter personal și privind libera circulație a acestor date și de abrogare a Directivei 95/46/CE (Regulamentul general privind protecția datelor).

Categoria de date prelucrate:

- date de identitate, de contact și de stare civilă: numele, prenumele, codul numeric personal, data și locul nașterii, sexul, cetățenia, etnia, domiciliul, numărul de telefon, adresa de e-mail, starea civilă, numele și prenumele părinților, semnătura
- date privind studiile dvs.

Scopul prelucrării datelor: datele dvs. personale de mai sus sunt prelucrate în vederea organizării examenului de finalizare a studiilor în cadrul Universității și vor fi utilizate în vederea îndeplinirii de către Universitate a obligațiilor sale legale în acest sens, în vederea întocmirii statisticilor proprii. Datele dvs. de contact vor fi utilizate drept mijloc de comunicare cu dvs. privind procedura examenului de finalizare a studiilor, respectiv urmărirea situației dvs. legată de inserția pe piața muncii, respectiv continuarea studiilor și includerea în sistemul Alumni al absolvenților Universității.

Temeiul juridic al prelucrării datelor cu caracter personal reprezintă obligația legală a Universității, rezultată din dispozițiile Legii nr. 1/2011 privind educația națională, din dispozițiile ordinelor emise de Ministerul Educației, respectiv din legislația privind învățământul superior, respectiv consimțământul dvs.

Datele dvs. cu caracter personal sunt transmise către: sistemul de gestionare a studiilor Neptun, RMU (Registru Matricol Unic), către instituția de învățământ organizatoare a examenului de finalizare a studiilor (după caz), respectiv către dvs. în vederea efectuării comunicărilor administrative cu dvs.

Datele dvs. cu caracter personal sunt păstrate de Universitate pe suport de hârtie și în format electronic pe perioada prevăzută de lege pentru diferitele tipuri de date (de ex. datele de identitate ale studenților Universității și datele privind rezultatele la studii ale studenților se păstrează permanent).


Prin prezenta vă aducem la cunoștință faptul că aveți dreptul să exercitați următoarele drepturi în legătură cu datele dvs. cu caracter personal: drept de acces, drept la rectificare, drept la ștergerea datelor, drept de a vă opune prelucrării, drept la portabilitatea datelor, drept de a retrage consimțământul, drept de opoziție, drept de a depune o plângere în fața unei autorități de supraveghere (Autoritatea Națională de Supraveghere a Prelucrării Datelor cu Caracter Personal, www.dataprotection.ro) și drept de a vă adresa instanțelor de judecată.

Datele de contact ale Universității în probleme privind protecția datelor: dataprotect@sapientia.ro

Declarație

Subsemnatul/a TAKÁCS NORBERT SÁNDOR, absolvent(ă) al/a specializării CALCULATOARE, promoția 2018-2022 cunoscând prevederile Legii învățământului superior nr. 199 din 2023 și a Codului de etică și deontologie profesională a Universității Sapientia cu privire la furt intelectual declar pe propria răspundere că prezenta lucrare de licență/proiect de diplomă/disertație se bazează pe activitatea personală, cercetarea/proiectarea este efectuată de mine, informațiile și datele preluate din literatura de specialitate sunt citate în mod corespunzător.

Localitatea, TÂRGU MUREȘ,
Data: 24.06.2024.

Absolvent
Semnătura.....

Kivonat

A Project Lifecycle Management (ProLife Management) applikáció egy modern webes platform, amely kiemelkedő jelentőséget tulajdonít a Software as a Service (SaaS) által nyújtott előnyöknek. A SaaS lehetővé teszi, hogy több felhasználó egyszerre férjen hozzá alkalmazásokhoz és adatokat tároljon, valamint megossza azokat egymással. Emellett a SaaS lehetővé teszi 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: szolgáltatott szoftver (SaaS), Web alkalmazás, projekt kezelés, ügyfélszolgálat

Rezumat

Aplicația Project Lifecycle Management (ProLife Management) este o platformă web modernă care pune mare preț pe beneficiile Software as a Service (SaaS). SaaS permite mai multor utilizatori să acceseze aplicații, să stocheze și să partajeze date simultan. În plus, SaaS permite controlul accesului pe bază de rol, prin care rolul utilizatorului determină datele pe care acesta le poate vizualiza sau modifica.

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: Software ca un Serviciu (SaaS), aplicație web, management de proiect, serviciu clienți

Abstract

The Project Lifecycle Management (ProLife Management) application is a modern web platform that places a high value on the benefits of Software as a Service (SaaS). SaaS allows multiple users to access applications, store and share data simultaneously. In addition, SaaS enables role-based access control, whereby the role of the user determines which data he or she can view or modify.

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: Software as a Service (SaaS), web application, project management, customer service

Tartalomjegyzék

Bevezető.....	12
1.1. Témaválasztás indoklása	12
Központosított Adatkezelés	12
Jobb Kommunikáció és Együttműködés	12
Biztonság és Hozzáférés Szabályozás	12
Hatékonyabb Dokumentumkezelés	12
Idő- és Erőforrás Megtakarítás.....	13
1.2. Szakirodalmi áttekintő	13
1.3. Kutatási kérdések	19
Back-end	20
Front-end	20
• JavaScript.....	23
Adatbázis.....	23
• MySQL	23
Version Control Systems (VCS).....	24
• GitHub	24
1.4. Célkitűzések	24
Rendszerspecifikáció	25
2.1. Felhasználói követelmények	25
2.1. Rendszerkövetelmények	28
a. Funkcionális követelmények:.....	28
b. Nem funkcionális követelmények:	29
Rendszerarchitektúra	31
• Adatbázis:.....	32
• Szerveroldali alkalmazás	33
• Web Szerver	35
• Felhasználó interfész.....	36
Megvalósítás	37

4.1. ProLifeManagemenet, Webes applikáció	37
4.2. Javascriptek használata	43
4.3. Tesztelések - kérdéses	43
4.4. Projektmenedzsment	43
Összefoglaló	44
5.1 Következtetések	44
5.2. Fejlesztési lehetőségek	44
5.3. Lehetséges felhasználási terület	44
Irodalomjegyzék	44

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

A Software as a Service (SaaS) olyan alkalmazásszoftver, amelyet a felhőben tárolnak és internetkapcsolaton keresztül használnak webböngésző, mobilalkalmazás vagy tablett segítségével. A SaaS szolgáltató felelős a szoftver és az azt futtató infrastruktúra működtetéséért, kezeléséért és karbantartásáért, míg az ügyfél csak fiókot hoz létre, díjat fizet, és máris használhatja a szolgáltatást. Összehasonlítva a helyszíni szoftverekkel, több pozitívumot és hatékonyabb megoldásokat tudunk kiemelni. A SaaS alkalmazások megvásárlása után azonnal használatba vehetők, akár néhány perc alatt, minimális kezdeti költségekkel. Ezzel ellentétben, a hagyományos szoftverek egy hosszadalmasabb folyamaton kell végigmenniük, ami a telepítést jelenti és ezt a folyamatot minden felhasználó el kell végezze a saját eszközén, akár többször is, annak függvényében, hogy hány eszözön szeretné használni az alkalmazást. A SaaS szolgáltatók gyakran frissítenek és adnak hozzá új funkciókat, de minden olyan simán történik, hogy nem beszélhetünk zavaró tényezőkről a kliens munkáját figyelembe véve. A SaaS alkalmazások a skálázhatóságukról is ismertek, mivelis az ügyfelek igényeik szerint csökkenthetik vagy bővíthetik a szolgáltatási szintet vagy több tárhelyet vásárolhatnak.

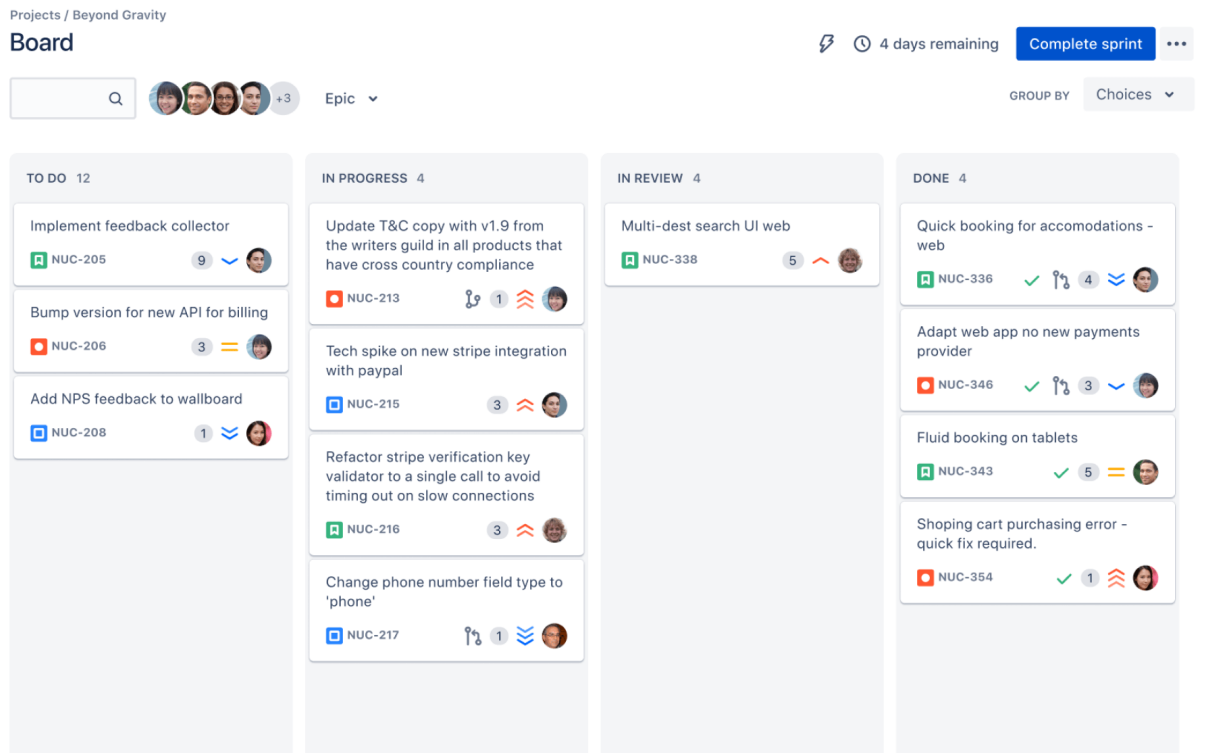
A projektmenedzsment alkalmazások arra készülnek, hogy mind az egyének, mind a csoportok számára segítséget nyújtsanak a projektek szervezésében, végrehajtásában és felügyeletében. Ezek a alkalmazások egy olyan szervezett módszert kínálnak a munka ütemezésére, a csapattagok elosztására, a határidők betartására és az egymással való kommunikációra, amely nagy valószínűséggel garantálja a projektek határidőre történő befejezését.

A népszerű projektmenedzsment-alkalmazások közé tartozik a JIRA, amely a szoftverfejlesztésben széles körben használt megbízható eszköz a problémák és projektek nyomon követésére, a Trello, amely kártyaalapú rendszert használ a feladatok szervezésére, amely egyszerű és összetett projektekhez egyaránt alkalmas, valamint az Asana, amely felhasználóbarát felületéről és robusztus feladatkezelési funkcióiról ismert.

1.2.1 JIRA

A szoftverfejlesztő csapatok rendkívül hasznosnak találják a Jira erős problémakövetési funkcióit. Hatékonyabbá teszi a problémák létrehozását, priorizálását és nyomon követését, ami elengedhetetlen az összetett projektek kezeléséhez. A csapatok testre szabhatják a platform mezőit, problématípusait és munkafolyamatait az egyes projektek igényeinek megfelelően.

A Jira összetettsége azonban nehézségeket okozhat a kezdő felhasználók számára, akik megfelelő képzés hiányában megfélemlítőnek találják a felületét. A költség egy másik figyelembe veendő tényező, ha az alkalmazás negatív részét vizsgáljuk, mivel a kisebb csapatok vagy a szűkebb büdzsével rendelkező szervezetek számára, akik számára fontosabb befektetési lehetőségek állnak rendelkezésre, nem biztos hogy havi szinten ki fogják fizetni a “Standard” 7.16\$ vagy a “Premium” 12.48\$ értéket, mindezt megszorozva a felhasználók számával. Az ismertebb és értékes projektmenedzsment alkalmazásokhoz képest egyes felhasználók úgy érzik, hogy a Jira jelentési és az adatok összefoglalási képességei korlátozottak. [\[12\]](#)



1.2.1. ábra - Képernyőfelvétel a Jira alkalmazás feladat követő oldaláról

<https://www.atlassian.com/software/jira/features/scrum-boards>

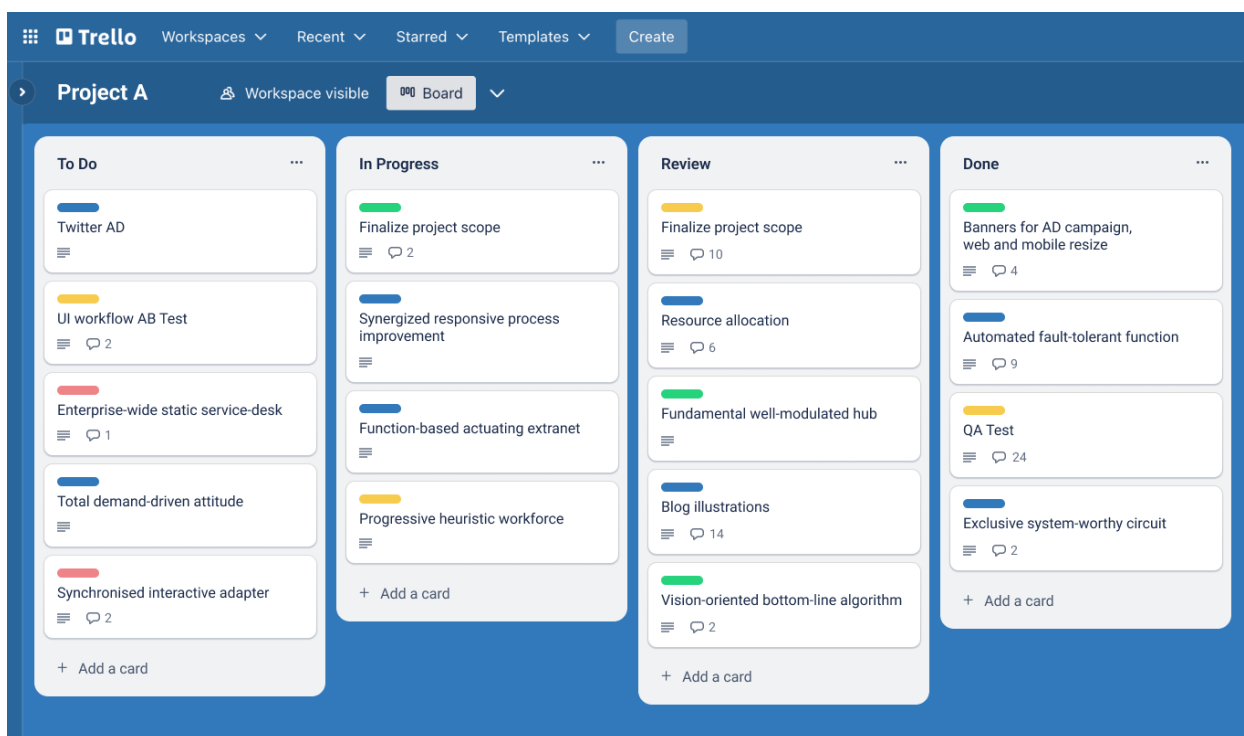
1.2.2 Trello

Trello egy népszerű projektmenedzsment eszköz, amely főleg a startup és a kisebb csapatokkal kedvelteti meg magát. Hasznos azokhoz a projektekhez, amelyek minimális struktúrát igényelnek a szervezéshez. Az interfész alapvető szinten táblákból, kártyákból és oszlopokból áll, ami nem megfelelő bonyolultabb csapatfolyamatokhoz. Azok a szervezetek, amelyek Agile Kanban munkafolyamatokat szeretnének bevezetni, gyakran kezdik Trello használatával. Például egy marketing kampány tervezése megfelelő projektnek számít, ahol a táblák és kártyák segítségével különböző kampányokat és tevékenységeket (például közösségi média posztok, hírlevelek, hirdetések) lehet kezelni és priorizálni. A kártyákhoz hozzáadhatók határidők, felelős személyek, és csatolmányok.

A Trello felhasználóbarát, ami egy nagy előnye. Néhány perc alatt beállítható és nagyon intuitív a használata, így a csapattagok azonnal fel tudják venni a tempót a kínált funkcionalitásokkal. Az ingyenes verzió tökéletesen alkalmas könnyű vagy átmeneti projektekhez és feladatokhoz.

A Trello hátrányai között szerepel, hogy kevés testreszabási lehetőség van, és mivel kínál szűrési és keresési lehetőségeket, az információk hosszú távú nyomon követése problémás lehet.

A Trello ideális a Kanban stílusú projektekhez. Nem alkalmas több csapat vagy az egész szervezeten átívelő Agile projektek kezelésére.



1.2.2. ábra - Képernyőfelvétel a Trello alkalmazás feladat követő oldaláról

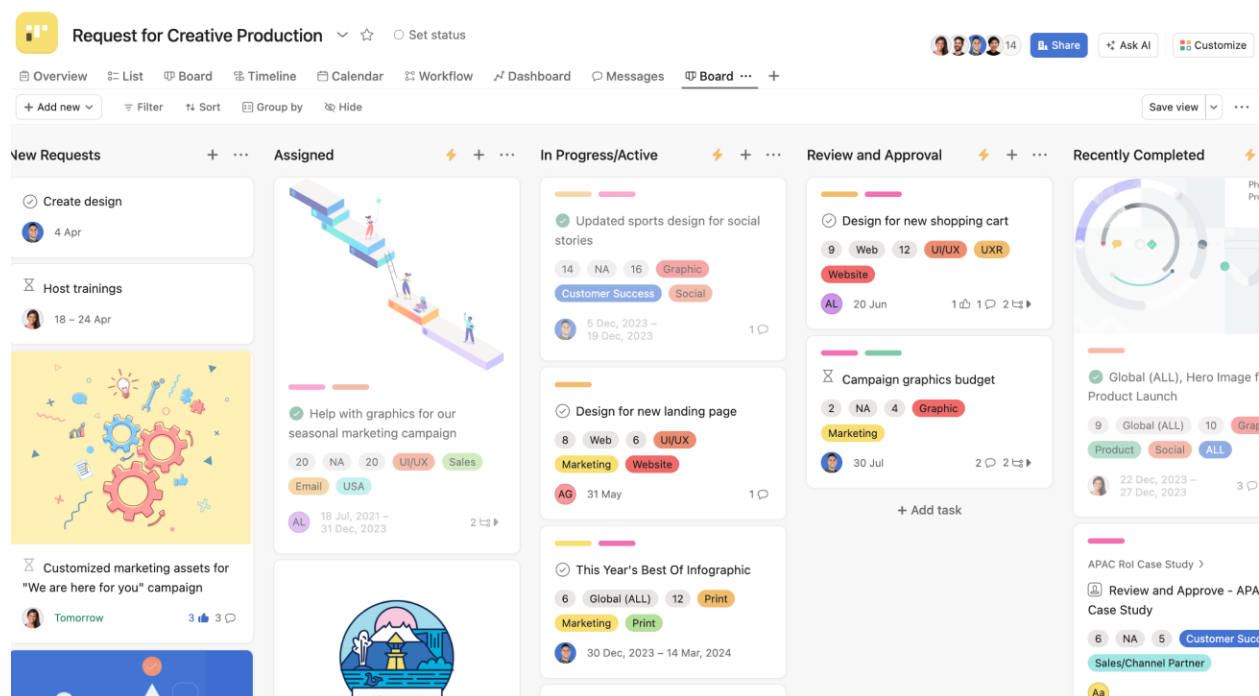
<https://planyway.com/blog/trello-kanban-principles>

1.2.3 Asana

Kezdetekben az Asana a Facebook számára volt lefejlesztve és kisebb csapatokra összpontosított. Azóta nagyot bővült, és olyan funkciók is megjelentek, mint például a Portfolio és a Workloads, amelyek a nagyobb szervezeteket veszik célul. A csapatok és a vezetők a Portfolio használatával gyorsan azonosíthatják, hogy mely projektek futnak jól, és melyek igényelnek több figyelmet. A projektmenedzserek egyenletesebben oszthatják el a munkát a csapattagok között, ha a Workloads segítségével látják, hogy ki hány feladaton és határidőn dolgozik. A jövőben a gépi

tanulás integrálásával tervezik, hogy intelligens javaslatokat tegyenek a csapattagok számára a rendszer használata közben.

Egy eltérő sajátossága az, hogy az Asana bevezette az "AI teammates" első verzióját - egy mesterséges intelligencia-alapú funkciót, amelynek célja, hogy segítse a felhasználókat egyes feladataik elvégzésében. Egy érdekes kijelentésre lettem figyelmes a mesterséges intelligenciával kapcsolatban, ami az Asana AI vezetőjéhez tartozik, aki azt mondta, hogy "Arra számítunk, hogy a mesterséges intelligencia idővel egyre összetettebb munkát fog segíteni az embereknek, de a munka végső soron az ember felelőssége marad".[14] Véleményem szerint ez a megközelítés nagyon kiegyensúlyozott. A munkahelyen alkalmazott AI által nyújtotta segítség jelentős előnyökkel jár, például a rutin feladatok automatizálása, hatékonyabb adatkezelés és jobb döntéshozatali támogatás. Ezáltal az alkalmazottak több időt és energiát fordíthatnak a stratégiai és kreatív munkára.






1.2.3. ábra - Képernyőfelvétel az Asana alkalmazás feladat követő oldaláról

<https://help.asana.com/hc/en-us/articles/14148613639323-Board-view>

1.2.4 Összefoglalás

A bemutatott projektmenedzsment applikációk megvizsgálása és felmérése során számos hasznos, de bonyolult funkciót vettem észre, melyek között olyan is található, amely szerintem túlságosan extrém és ködbe viszi az egyszerű felhasználókat. Ezért én egy középútat próbálok követni a ProLifeManagement applikációban, ahol egy letisztázott, minden felhasználó számára könnyen érthető és kezelhető felületet hoztam létre. Úgy döntöttem, hogy a kliens is egy nagy szerepet játszon az alkalmazás keretén belül, és pedig két módon. Elsőként a projektek belső oldalain, ahol a hírfolyam segítségével meglehet tárgyalni a maradandó fontos információkat, mint például egy WordPress admin bejelentkezési adatokat vagy egy WooCommerce webshop esetén a EuPlătesc fiók adatait és más szükségszerű, kritikus adatokat, amelyek nélkül nem lehet haladni a projektek megvalósításával. Másodjára pedig proa feladatok kezelésében. Ez úgy történik, hogy a kliens által észrevett hibák, helytelenségek vagy működési zavarok észrevételekor a projekten, egy visszajelzést küldhet a csapat számára, ami megjelenik a projekten dolgozó felhasználók listájukban. Annak érdekében, hogy a kliens által észrevett hiba jelzése ne vesszen el, a csapat tagjai e-mailes értesítést is kapnak.

Amit mind a három alkalmazásban kötelezően szükségszerűnek láttam, az a Kanban táblák létezése, amit én is implementáltam. Jelenleg mindegyiküknél egyszerűbb a megvalósítása, de használható a feladatok nyomon követésére. Nagy hasznát látom az online fehér táblák integrálására, ami esetemben a Whiteboard Team, lehetővé teszi a csapattagok számára, hogy közösen dolgozzanak és vizuálisan tervezzenek, ami különösen hasznos brainstorming ülések során. Nagy mértékben elősegíti a gyorsabb és helyes döntésekben, úgy egyéni, mint csapat szinten is. Egy adminhoz több projekt tartózhat, ezért az egyszerűbb és gyorsabb azonosítás érdekében a projektek könnyebb megkülönböztetését szemályreszabott színekkel segítem elő. Ez az alkalmazás ideális választás kisebb csapatok és projektek számára, mivel egyszerű és hatékony eszközöket kínál a munkafolyamatok követésére és a csapattagok közötti együttműködésre. Az applikáció könnyen használható, gyorsan beállítható és nem igényel bonyolult képzést, így a csapatok azonnal elkezdhetik használni és élvezhetik a funkcionalitások előnyeit.

			
	Jira	Trello	Asana
Price (per user)	Free: \$0 Standard: \$5/user Premium: \$10/user Enterprise: Custom	Free: \$0 Standard: \$5/user Premium: \$10/user Enterprise: \$17.50/user	Basic: \$0/free Premium: \$10.99/user Business: \$24.99/user Enterprise: Custom
Pros	Really cheap prices Atlassian marketplace Ideal for tech teams	Simple interface Does the basics well Cheap enterprise plan	Lots of integrations Clean interface Easy automations Great data dashboard
Cons	Not many collab tools Steep learning curve Lots of tech jargon	Limited free option Not for large teams Add-ons cost extra	Very expensive Little customizability Limited free plan
Verdict	A great value piece of software that's ideal for tech, software development and engineering teams.	A fairly-priced, stripped-down option best for small teams who need a central location for basic task management.	Ideal for both small and large projects across various industries.

1.2.4 ábra. Jira, Trello és Asana projektmenedzsment applikációk összehasonlítása ár, előnyök, hátrányok és besorolás alapján [10] [11]

1.3. Kutatási kérdések

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

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

- Milyen nehézségekkel találkozhatunk egy SaaS szoftver 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 SaaS szoftver megvalósításához számos technológiát és eszközt lehet használni.

Az általam választott technológiák és azok alternatív opciói a következők:

Back-end

- **PHP:** Széles körben használt és támogatott nyelv, különösen webes alkalmazások fejlesztéséhez, amelyet számos szolgáltató szoftver használ, mint például a Google Workspace, a Salesforce és a Dropbox.

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(**1.3.1. 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.

1.3.1. ábra. Html leíró nyelvben létrehozott weboldal

Önmagában legtöbbször egy statikus weboldalt tud létrehozni, semmi lehetőséget nem ad arra, hogy az elemeket valahogyan elrendezni tudnánk. Legtöbbször 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 megnevezéssel, 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 SaaS 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. A ProLifeManagement applikáció egy olyan felület legyen, ami biztosítja 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, de 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 és kategorizálásának a lehetősége.
- Egy web alapú alkalmazás fejlesztése annak érdekében, hogy bárki számára elérhető legyen, bármiféle letöltés igénye nélkül.
- A rendszer felosztása két nagy részből áll össze, éspedig a cégek modul és a kliens modul.

2. fejezet

Rendszerspecifikáció

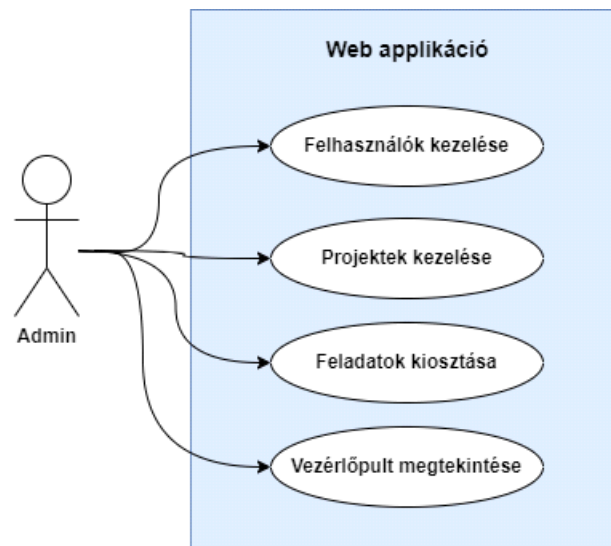
A ProLifeManagement alkalmazás célja a projekt munkafolyamatok racionalizálása, a feladatok nyomon követésének javítása és a csapatokon belüli együttműködés fokozása. A valós idejű értesítések, a feladatkezelés, a projekt létrehozása és szerkesztése, valamint a felhasználói adminisztráció az elsődleges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

2.1. Felhasználói követelmények

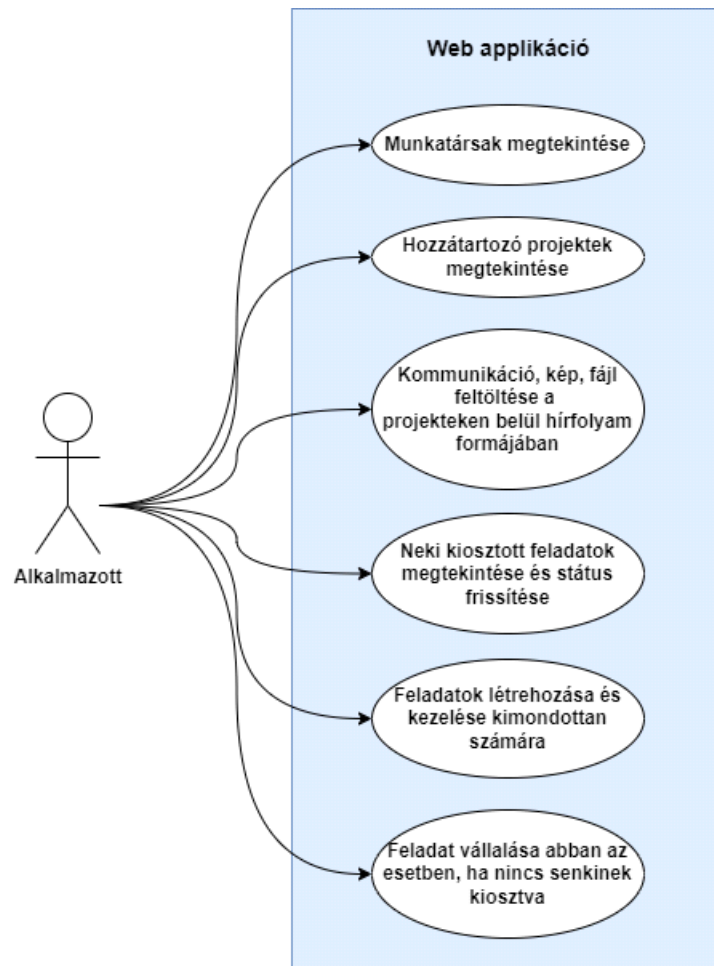
Az applikáció úgy van megtervezve, hogy megfeleljen a vállalatokon belüli különböző felhasználói szerepkörök igényeinek. Három féle felhasználótípus létezik: admin, alkalmazott és kliens.

- 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 jogosultsággal rendelkezik, a céghez fűződő adatokra vonatkozóan. Ezen belül gondolok a projektekre, projekt hozzászólásokra, fájlokra, feladatokra, alkalmazottakra, kliensekre, online fehér táblákra. A létező Dashboard-on neki megjelennek a statisztikák a cégen belül végzett feladatokról, a projektek száma, az alkalmazottak száma, egy tevékenység lista a legrelevánsabb történésekkel.

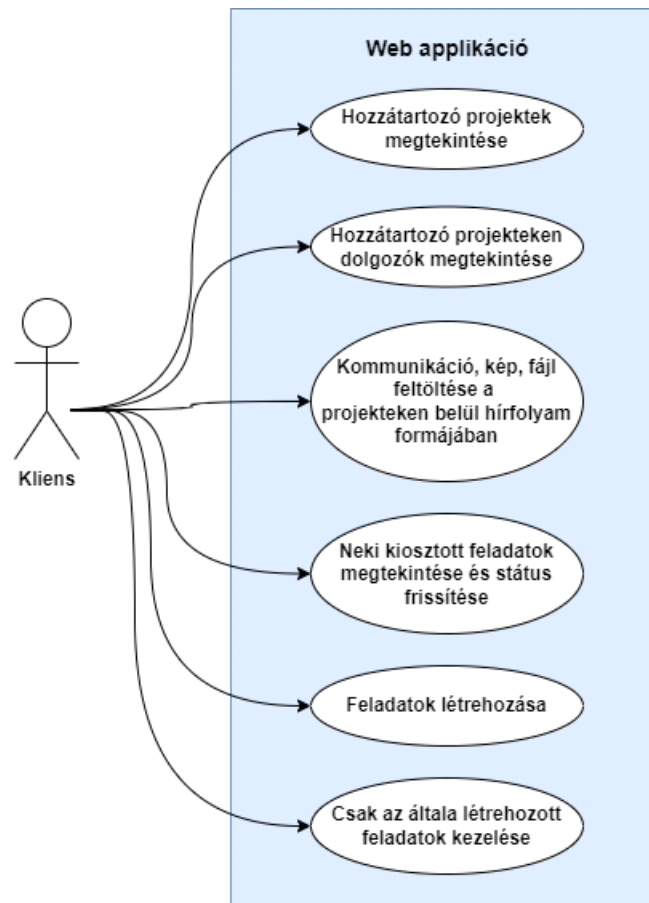
- Az alkalmazott felhasználó csak úgy jöhet létre, ha az admin hozzá adja az alkalmazottak listájához. Ekkor egy felugró ablakban be kérem az alkalmazott email címét, amire küldök egy levelet. A levélben egy webcím szerepel, ami egy generált tokent tartalmaz GET paraméterként. A webcím a regisztrálás 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:

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

Admin szerepkörben, a felhasználó úralkódása 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 műveletekre gondolok, melyek által példa képpen, felhasználókat tud meghívni bizonyos projektekhez, módosítani tudja az adataikat és szükség esetén törölni is tudja. Hasonló képpen cselekedhet az előbbi felsorolt, az applikációban szereplő elemekkel is. Továbbá lehetősége van megtekinteni az összes feltöltött dokumentumot, amik lehetnek szerződések, arculati kézikönyv, árajánlat, számlák, számlaszám kivonatok és különböző csatolmányok.

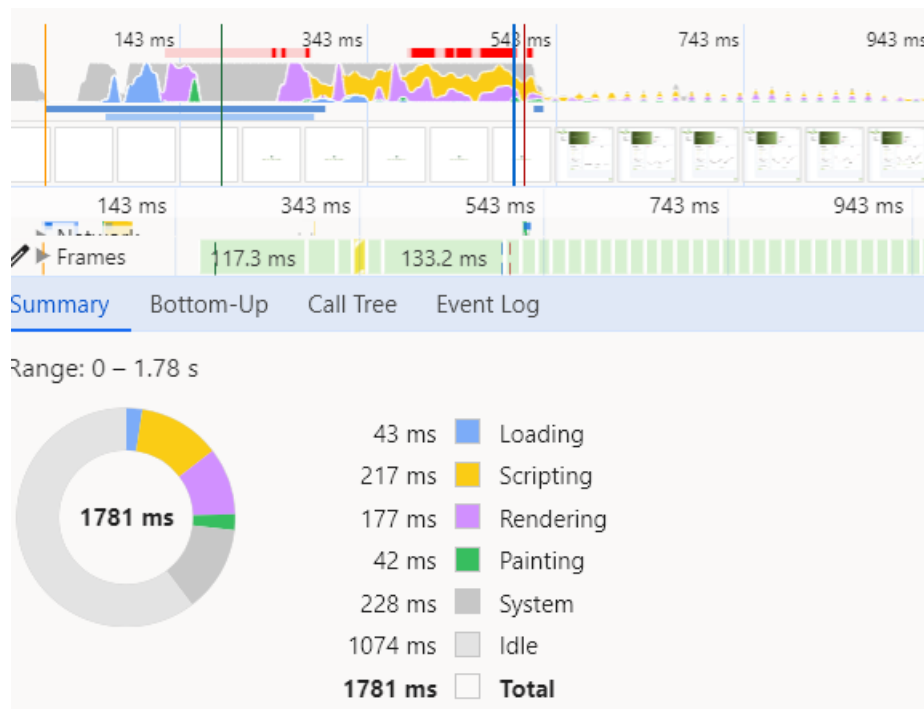
Alkalmazott(kolléga) szerepkörben, a felhasználó hasonló jogosultságokkal rendelkezik mint az admin, kis eltérésekkel. Nem képes a felhasználók és a projektek adatainak a módosítására, vagy törlésére. Csak azokat a projekteket látja, amelyekhez hozzá van adva, tehát dolgozik rajtuk és ezen belül pedig 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 alkalmasak, 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:

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), ami az első vizuális tartalmat jelenti, 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 áshatja 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, ahol a sandbox módot használom és arra csatlakozok az SMTP(Simple Mail Transfer Protocol) hitelesítő adatokkal. Ez a módszer a küldések biztonságos tesztelésre van használva, anélkül hogy az e-maileket kézbesítenék a címzetteknek. Szintén a Toast[9] nevű értesítési könyvtár implementálását

is hasznosnak találtam, 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.

A fejlesztés során törekedtem arra, hogy egy moduláris rendszert hozzak létre, ezért is használtam az MVC programtervezési mintát. A 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 osztályát, 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 használtam 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.

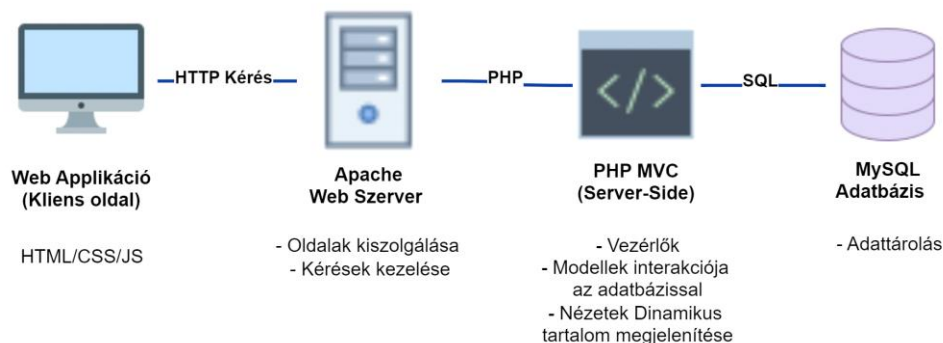
A Bootstrap támogatja a böngészők és platformok legújabb stabil verziókat, viszont ha Internet Explorer böngészőn szeretnénk használni az applikációt, figyeljünk arra, hogy ne használjunk a 10-es verziónál régebbi kiadást.

3. fejezet

Rendszerarchitektúra

A projektmenedzsment-alkalmazás teljes funkcionalitását több komponens biztosítja, amelyek együtt egy rendszerként működnek. Minden komponens kritikus szerepet játszik az

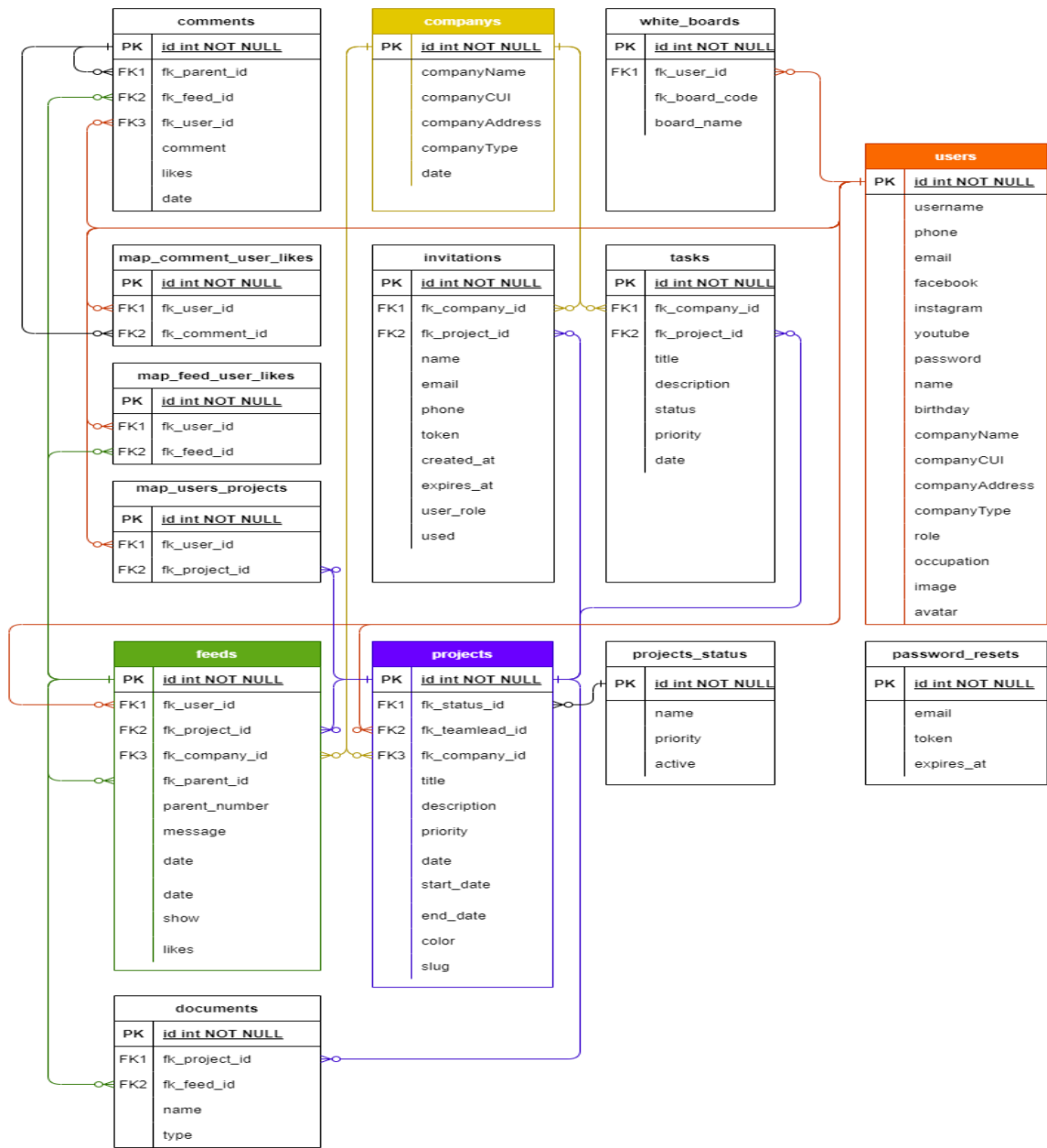
alkalmazás használhatóságának, teljesítményének és karbantarthatóságának biztosításában. Ezek az elemek a következőkből állnak:



3. 1. ábra. Teljes rendszer architektúrája

- **Adatbázis:**

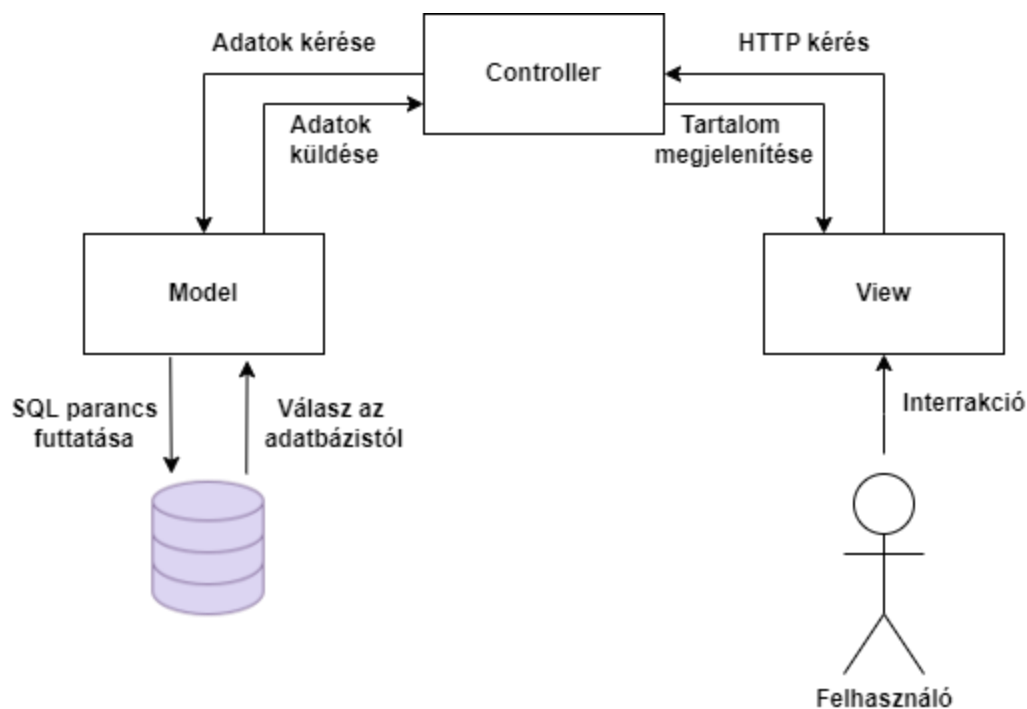
Az adatbázis az összes tartós adat tárolásáért felelős, beleértve a felhasználói információkat, projekteket, feladatokat és tevékenységeket. A hozzáférés a Modell rétegben történik, amely kapcsolatba lép az adatbázissal a PDO-n (PHP Data Objects) keresztül végrehajtott SQL-lekérdezések segítségével. 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**

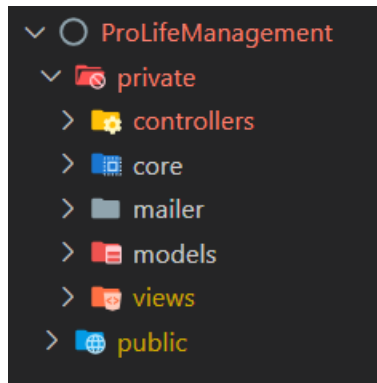
Az applikáció backend részét PHP programozási nyelven írtam meg az MVC (Model – View – Controller) architektúrát követve(3.3. és 3.4. ábrák.), hogy a rendszer átláthatóbb legyen, karbantarthatóbb és könnyebb legyen a további fejlesztése.



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. Jelenleg úgy vélem, hogy a projekt még nem annyira stabil, hogy automatizált egységteszteket hozzak létre hozzá, ezért úgy döntöttem, hogy a manuális teszteléssel fogok kezdeni. Szerintem ez a helyes választás mindaddig, amíg gyors, alkalmi teszteléseket kell végezni, új funkciókat implementálok vagy érdekel az applikáció használhatósága.

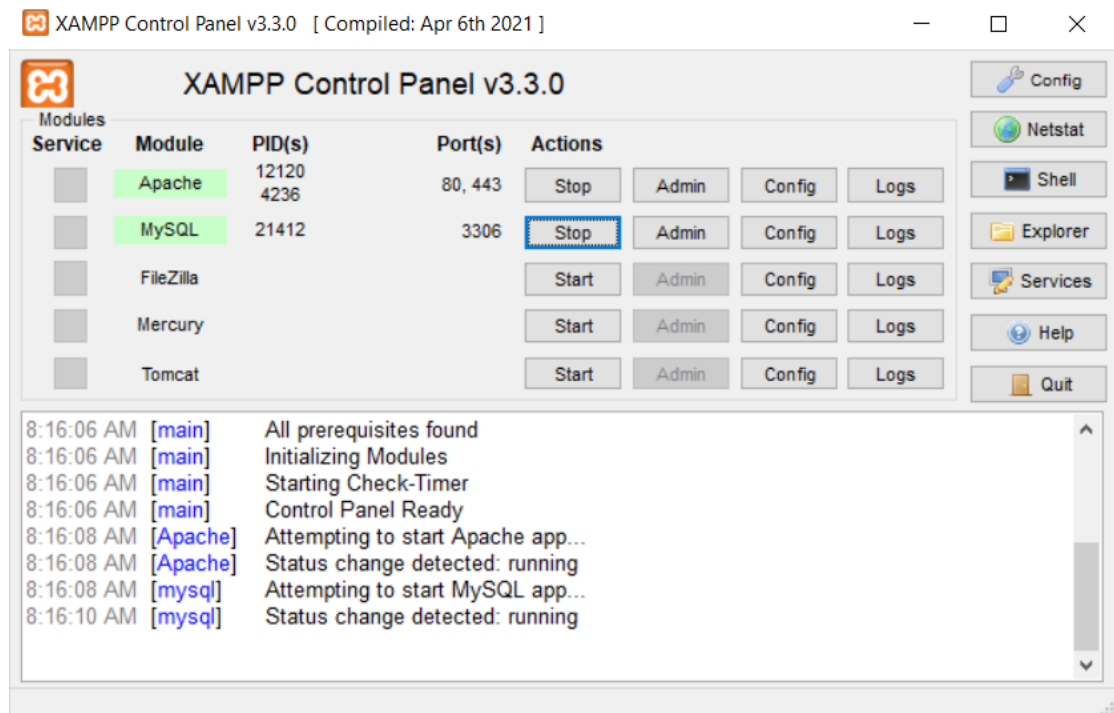
A modellréteg feladata, hogy kapcsolatot létesítsen a MySQL-adatbázissal. A View réteg felelős az adatok megjelenítéséért a felhasználó számára. Beágyazott PHP kódokat tartalmaz, amelyek a vezérlőtől kapott adatok alapján generálnak dinamikus HTML tartalmakat. A vezérlő rétegben kezelem le a bejövő kéréseket, 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**

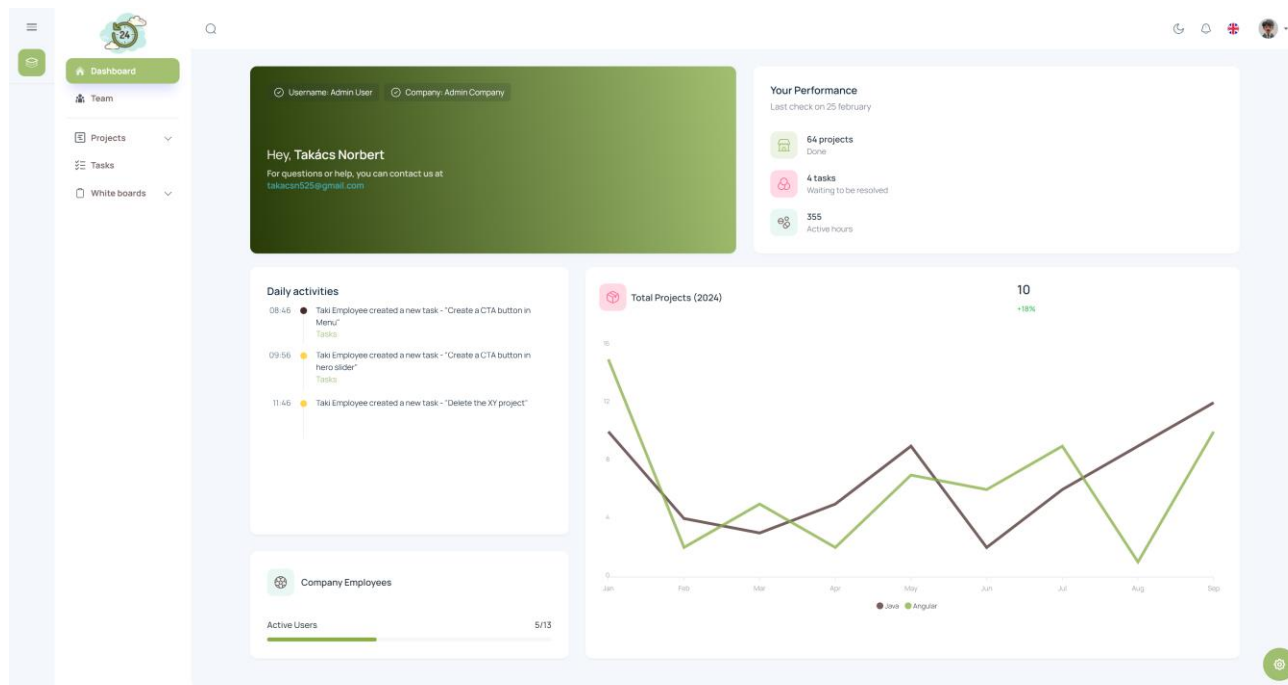
A XAMPP-ben található Apache webszerver kezeli a kliensek (a webböngésző felhasználók) HTTP(Hypertext Transfer Protocol) kéréseit, és kiszolgálja az alkalmazás weboldalait. Feldolgozza a PHP szkripteket, és kommunikál az adatbázissal az adatok lekérdezése és tárolása érdekében. A XAMPP Coracionntrol Panel segítségével egyszerűen indíthatjuk és leállíthatjuk az Apache szolgáltatást, valamint figyelhetjük a szerver állapotát és logjait.



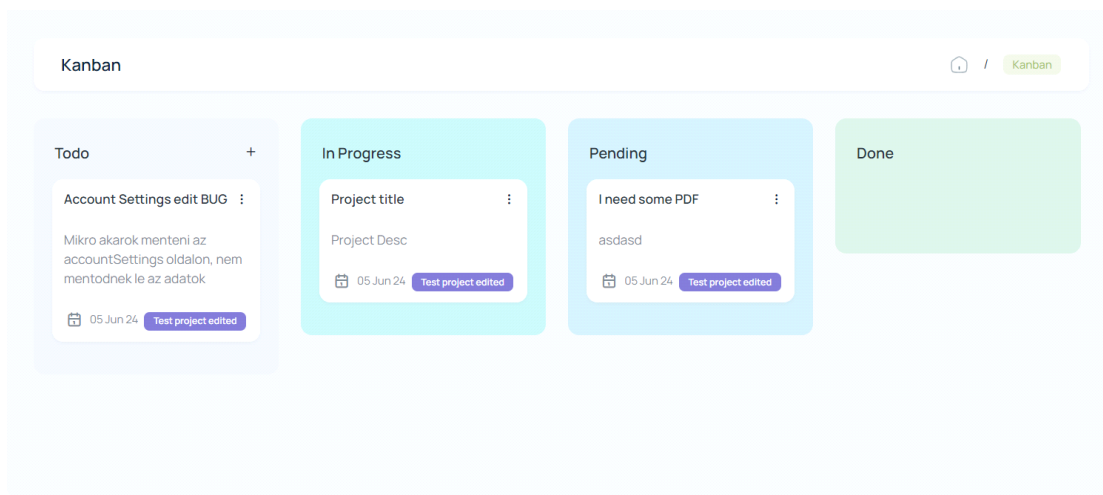
3. 5. ábra. A XAMPP vezérlőfelület

- **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



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, taskok 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 bejelentkezéskor tudjam meghatározni a felhasználó szerepkörét és aszerint kezeljem a megjelenítendő adatokat, funkcionalitásokat 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. 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 résznél. Amíg nincs megadva minden kötelező bemeneti adat, hibaüzenetekkel térek vissza a kliens felé.



Sign Up

Company Name

Only letters allowed in company name

Unique identification code (CUI/CIF)

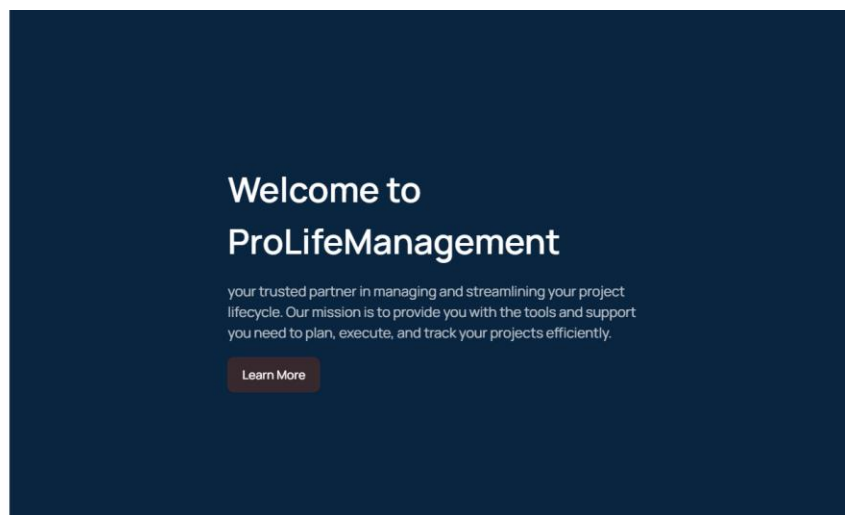
You have entered the wrong CUI/CIF

Company Address

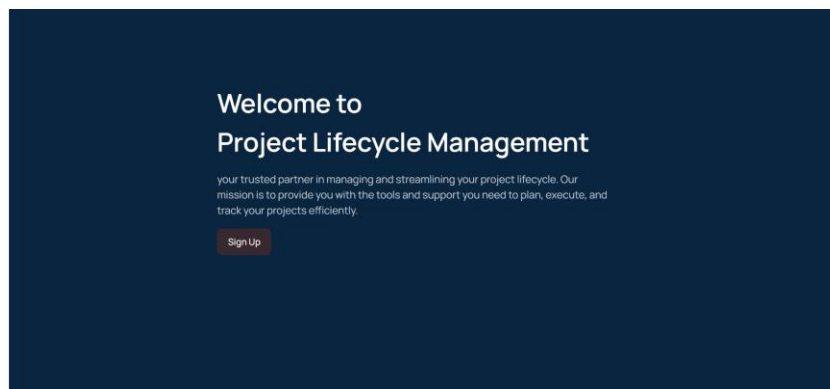
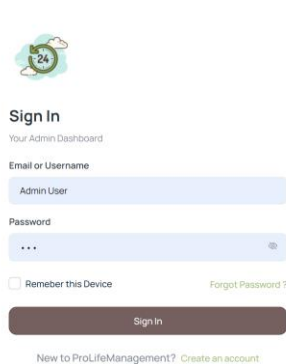
You must give us the address of your company

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

You must give us the type of your company



4.1. ábra. Applikáció - Cégek regisztrálása oldal.



4.2. ábra. Applikáció - Bejelentkezés oldal

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 validálva vannak. Mindegyikük egyediek kell 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, kétszer kérem be a jelszót. 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 generáltak, 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.

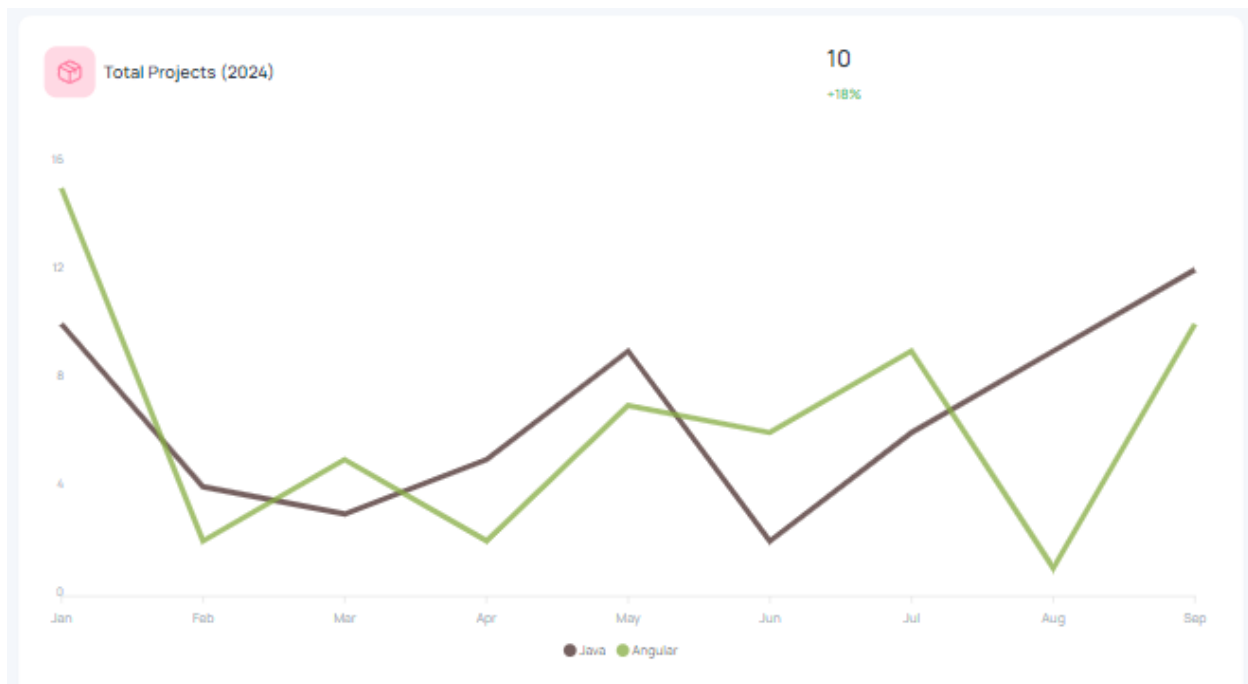
A bcrypt hash-nek van egy specifikus formátuma:

- **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.

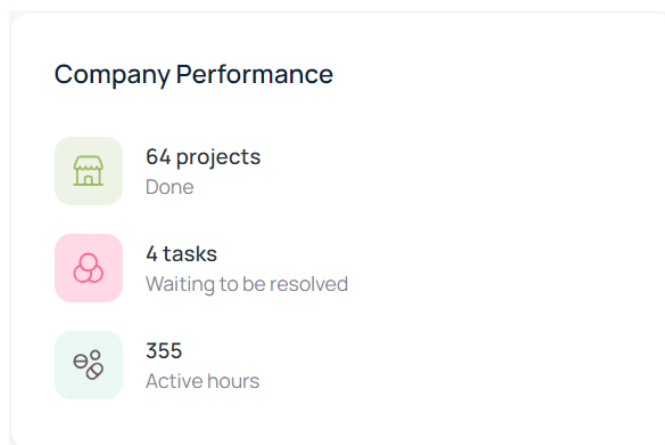
A bejelentkezési adatok sikeres validálása után, a felhasználói adatokat a szervertől oldali `$_SESSION['user']` változóba tárolom addig, amíg a munkamenet le nem jár, vagy amíg a felhasználó kizárólagosan jelentkezni. A bejelentkezést követően, a felhasználónak megjelenik a főoldal, egy baloldalsó és egy felső menüvel. Az oldalsóban a főoldal, a csapat, a projektek, a feladatok és a fehér táblák érhetőek el. A felsőben pedig az újdonságok, egy sötét vagy világos témaváltoztató gomb és az avatár vagy profilkép, ahol egy legördülő menüben el lehet navigálni a felhasználói adatok kezelésére vagy ki lehet jelentkezni.

A főoldalon, felhasználó típustól függő adatok jelennek meg. Mindenkit egy köszöntő szöveggel üdvözlünk, ami szerintem egy plusz löketet ad a felhasználónak, hogy használja és próbálja ki az applikációt. Emellett olyan adatokkal szolgál a felhasználóknak, ami számukra a legfontosabb, annál sem kevesebbet, sem többet nem szeretnék számukra megjeleníteni. Az adminnak projektek összességét és típusait ábrázolom egy vonaldiagrammal(4.3. ábra.), az elmúlt egy éves időszakra. X tengelyen a hónapokat, Y tengelyen pedig a számukat. A projektek típusait különböző színekkel szemléltetem.



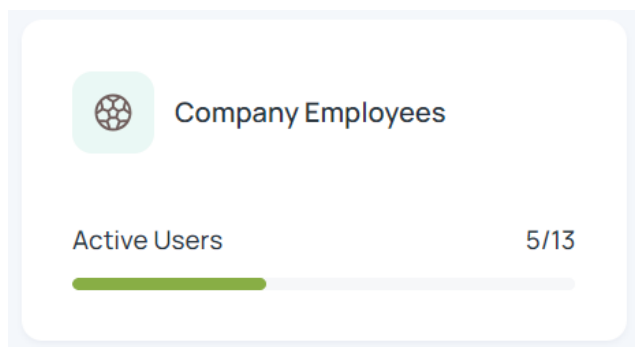
4.3. ábra. Applikáció - Vonaldiagram a projektek ábrázolására.

Következőként egy olyan adatdobozt(4.4. ábra.) hoztam létre, ami fontos egy admin számára hogy tudatában legyen. Az összes projekt és feladat száma és az összes aktív órák, ami abból áll össze, hogy minden alkalmazott aktív óráit egybeítem. Ez a doboz külön minden alkalmazott számára is elérhető, ahol mindenki kimondottan a saját előrehaladását látja.



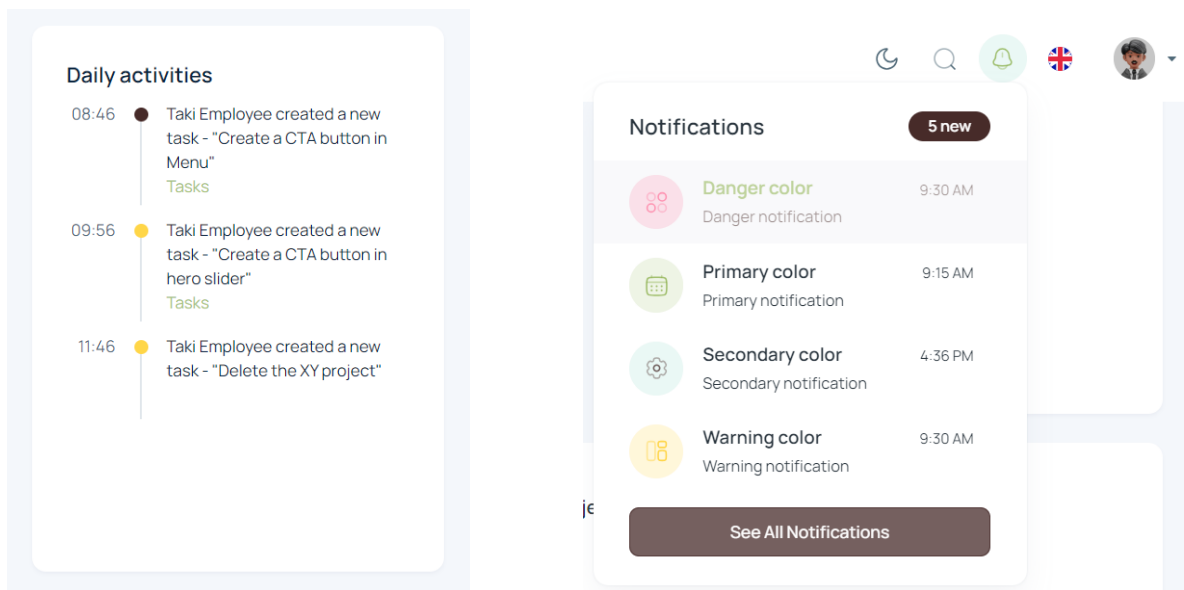
4.4. ábra. Applikáció - Admin és alkalmazott számára fontos adatok ábrázolása

A harmadik doboz egy valós idejű információközlés az admin felé, ami kimutatja, hogy jelenleg hány alkalmazott van bejelentkezve az applikációba(4.5. ábra). Ha megszeretné tekinteni részletesebben hogy kik az aktív és kik az inaktív felhasználók, akkor vagy a “Team” menügombra, vagy a dobozra kattintva át lesz irányítva egy belső oldalra, ahol listászerűen fel lesznek tüntetve.



4.5. ábra. Applikáció - Aktív alkalmazottak kimutatása

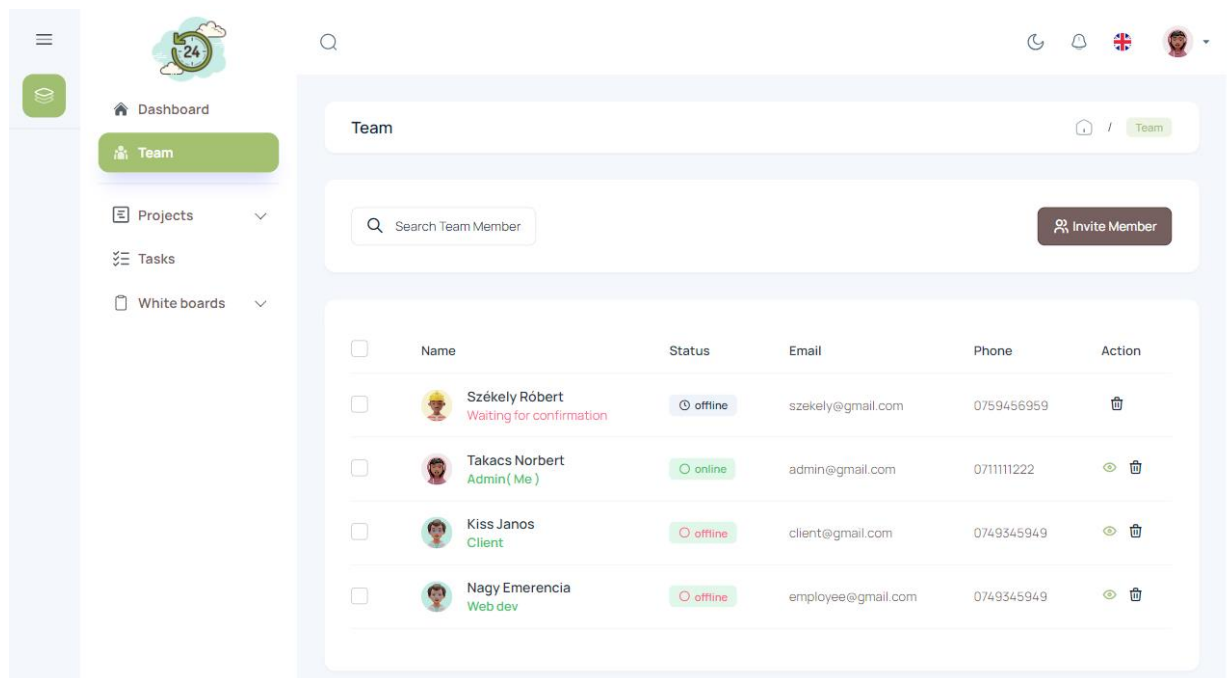
Mindemellett egy napi tevékenység lista(4.6. ábra) is jelen van úgy a főoldalon, mint a felső menüsorban, ahol követni lehet a cégen belüli aktivitást.



4.6. ábra. Applikáció - napi tevékenység lista.

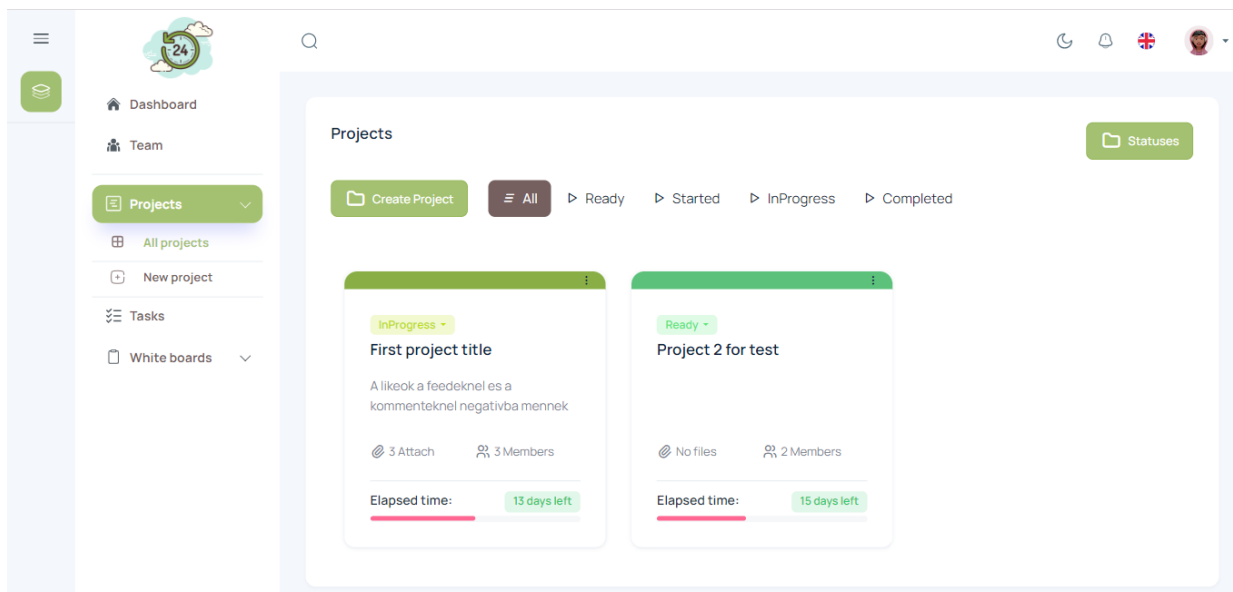
A Csapattagok listája(4. 7. ábra) áttekintést nyújt a projekten dolgozó minden csapattagról. Ez olyan lényeges információkat tartalmaz, mint a profilkép, felhasználónév, státusz (online vagy offline), e-mail valamint a telefonszám.

- Profilkép: A csapattagok feltöltött profilképe vizuálisan mutatja be őket.
- Minden csapattagot a rendszerben kiválasztott felhasználónév vagy e-mail azonosít.
- Státusz: Ez segít meghatározni a csapattagok elérhetőségét a kommunikáció és az együttműködés szempontjából azáltal, hogy jelzi, hogy jelenleg online vannak-e vagy sem.
- E-mail: Kötelező adat, hogy kapcsolatba lehessen lépni egymással.
- Telefonszám: Kötelező adat, sürgősség esetén gyors elérési lehetőség.
- Ha egy új csapattagnak csatlakoznia kell a projekthez, az adminok meghívó e-mailt küldhetnek neki. Ez az e-mail tartalmaz egy linket, benne egy generált egyedi tokent, amely 48 óráig érvényes. Arra külön kitértem, hogy egy felhasználó csak egy meghívót kaphasson ugyanattól a vállalatától a token érvényességi ideje alatt. Az új csapattag egy biztonságos oldalra kerül az emailben kapott link kattintását követve, ahol létrehozhatja saját profilját. Az új tag egy profilkép feltöltésével vagy az avatar kiválasztásával és a szükséges információk, közöttük a felhasználónév és a jelszó megadásával befejezi a profilja beállítását. Amint a profilja be van állítva, máris elkezdhet dolgozni a projekten.



4. 7. ábra. Applikáció - Csapattagok listája

A zökkenőmentes együttműködés és a hatékony projektvégrehajtás érdekében a projektek oldal(4. 8. ábra) alapos áttekintést nyújt a projektek állapotáról és a fontos projekt specifikus adatokról. Az oldalon kártyák sorozata található, mindegyikük adatokkal szolgálva külön-külön a projektről. Testreszabható státuszok, mint a “Nem indult”, “Folyamatban”, vagy a “Befejezve”, cím, leírás, projekten dolgozó csapattagok száma, feltöltött fájlok száma és a projekt határidejéig hátralévő napok száma. . A gyorsabb vizuális megkülönböztetéshez érdekében testreszabható, projekthez kötődő egyedi színek kiválasztását hagytam jóvá a felhasználóknak. Az adminok könnyen létrehozhatnak új projekteket vagy módosíthatják a meglévőket olyan részletek megadásával, mint a cím, a kezdési dátum, a lejárat dátum, a prioritás, a szín, a projektvezető, a kijelölt csapattagok és a leírás. Szerintem projektek sikeres befejezése érdekében, ez a konfiguráció segít a jobb szervezésben, a hatékony és valós idejű előrehaladásban és egy alaposabb felügyeletben.



4. 8. ábra. Applikáció - Projektek oldal

A projektek belső oldalán több interakciós lehetőség adódik. Klienseket csak innen lehet meghívni, a csapat tagok meghívó email-hez hasonló módon, token generálással, ami 48 óráig érvényes. Egyből lehetőség adódik a csapat tagjainak változtatására, projekt adatainak a frissítésére, csapattagok és kliensek profiljainak a megtekintésére, látjuk a projektvezetőt és a projekt létrehozási dátumát, a hátramaradt napokat a projekt lejártáig és a leírást.

Szerintem a legfontosabb funkcionális a hírfolyam, aholis a projekt tagjai kommunikálni tudnak, fájlok megosztására képesek egymás között. Mindenki el tud kezdeni egy témát, ami alá hozzászólások formájában folytatódhat a beszélgetés írás formájában. Reagálni is tudnak egymásnak a kedvelés gombra kattintva, így gyorsan és könnyedén megerősíthetik egymás gondolatait és ötleteit.

A feladatok oldal következik

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

4.3. Tesztelések - kérdéses

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] Jira vs Asana Projektmenedzsment alkalmazások összehasonlítása. <https://tech.co/project-management-software/asana-vs-jira>
- [11] Jira vs Trello Projektmenedzsment alkalmazások összehasonlítása. <https://tech.co/project-management-software/jira-vs-trello>
- [12] Jira Software részletek, előnyök és hátrányok. <https://www.trustradius.com/products/jira-software/reviews?qs=pros-and-cons#comparisons>

[13] Trello és Asana Softwarek részletek, előnyök és hátrányok. <https://blog.veniture.net/en/a-comparison-of-work-management-tools-trello-vs.-asana-vs.-jira-vs.-ms-project-vs.-jira-work-management>

[14] Asana AI. <https://www.axios.com/2024/06/05/asana-ai-teammates-generative-ai-genai-chatbot>