

TDK-dolgozat

GÓLYA Gergő, Sapiientia Erdélyi Magyar Tudományegyetem, Marsovásárhelyi Kar,
Matematika-Informatika Tanszék
KONCZ Hunor, Sapiientia Erdélyi Magyar Tudományegyetem, Marsovásárhelyi Kar,
Matematika-Informatika Tanszék

**SAPIENTIA ERDÉLYI MAGYAR TUDOMÁNYEGYETEM
MAROSVÁSÁRHELYI KAR,
INFORMATIKA SZAK**

BudgetHub: Modern és Intelligens Ingatlankeresés

Informatikatudományi szekció

GÓLYA Gergő, Sapiientia Erdélyi Magyar Tudományegyetem, Marosvásárhelyi Kar,
Matematika-Informatika Tanszék

KONCZ Hunor, Sapiientia Erdélyi Magyar Tudományegyetem, Marosvásárhelyi Kar,
Matematika-Informatika Tanszék

Témavezető: dr. JÁNOSI-RANCZ Katalin Tünde, Adjunktus, Sapiientia Erdélyi
Magyar Tudományegyetem, Marosvásárhelyi Kar, Matematika-Informatika Tanszék

Kivonat

Az ingatlanpiac gyorsan változó és egyre inkább szerteágazó piacág, amely az emberek nagy részét közvetlenül vagy közvetett módon érinti. A vásárlási folyamat kihívásokkal teli és időigényes lehet, különösen a számos ingatlanhirdető oldal, platform, irodák és személyek széles kínálatai miatt. A változó árak, az eladásra meghirdetett ingatlanok minőségei és ezek közötti különbségek sokszor megosztják az érdeklődők figyelmét és az erre szánt erőforrásokat.

Dolgozatunk célja egy mesterséges intelligencia alapú, ingatlan kereső webalkalmazás készítése, mely a piacon lévő legújabb és legaktuálisabban elérhető ingatlanokat jeleníti meg felhasználói számára. Mindez egy olyan módszer felhasználásával történik, mely több meglévő online hirdető felületről gyűjti össze és dolgozza fel a szükséges információkat. Emellett innovatív és intelligens megközelítést kínál az ingatlanvásárlás folyamatához. Az alkalmazás segítségével a felhasználók könnyedén és hatékonyan böngészhetnek az ingatlanok között, és az AI-alapú megoldások révén képesek lesznek a számukra legrelevánsabb lehetőségeket azonnal azonosítani és kiemelni a széles kínálatból.

Kulcsszavak: ingatlan, webalkalmazás, mesterséges intelligencia

Tartalomjegyzék

1. Bevezető	6
2. Projekt célja	7
2.1. Az intelligens ingatlankeresés korszaka	7
2.2. A felhasználókért: idő- és pénztakarékosság	8
3. Szakirodalom	8
4. Követelmény specifikáció	11
4.1. Felhasználói követelmények	11
4.2. Rendszerkövetelmények	12
4.2.1. Funkcionális követelmények	12
4.2.2. Nem-funkcionális követelmények	13
5. A rendszer architektúrája	15
6. Felhasznált technológiák	16
6.1. Web scraping	17
6.2. User Service	24
6.3. Mesterséges intelligencia	27
6.4. Docker	38
6.5. Kubernetes	39
7. A BudgetHub alkalmazás	40
8. Összefoglaló	50
9. Továbbfejlesztési lehetőségek	51
Irodalomjegyzék	56

Ábrák jegyzéke

4.1. Use Case Diagram	12
4.2. Szekvencia Diagram - Felhasználó jogai	14
5.1. Rendszer architektúra	15
6.1. Real Estate Service – Séma	22
6.2. Lakások - adat áramlása	23
6.3. User Service – Fájl architektúra	25
6.4. User Service – Séma	26
6.5. YOLOv5 beépített modell felismerései - szoba	29
6.6. YOLOv5 betanított modell felismerései - További tárgyak	30
6.7. YOLOv5 általi felismerések becslése	31
6.8. Átlag időtartam: Saját vs Sentisight.ai	33
6.9. Átlag felismerés: Saját vs Sentisight.ai	34
6.10. Átlag felismerés küszöbérték módosításának függvényében: Saját algoritmus	35
6.11. Átlag időtartam képek mennyiségének módosítása függvényében: Saját algoritmus	36
7.1. Webalkalmazás navigációs diagramja	40
7.2. Kezdőoldal	41
7.3. Szűrt adatok - Filter Page	42
7.4. Üzenet - Bejelentkezés szükséges	43
7.5. Üzenet - Kedvencek listájának módosítása	43
7.6. Ingatlan adatlapja - Detail Page	44
7.7. Képek nagyobb felbontásban való nézete	45
7.8. Regisztráció és bejelentkezés - Register - Login Page	46
7.9. Elfelejtett jelszó - Forgot Password	46
7.10. Felhasználói fiók adatlapja - User Page	47
7.11. Kedvencek listája - Favorites Page	48
7.12. Kedvezőbb ajánlatok listája	49

Kódrészletek jegyzéke

6.1. Példa Stora.ro robots.txt fájlja	17
6.2. Scrapy és Selenium használata projektünkben	19
6.3. Real Estate Service - LocationModel	23
6.4. Real Estate Service - ApartmentModel	24
6.5. User Service - Felhasználó keresése email alapján	26
6.6. A tárgyak nevének, valószínűségének és pozíciójának kinyerése	37
6.7. A felismert tárgyak kategorizálása	37

1. fejezet

Bevezető

A BudgetHub egy kifinomult és forradalmi ingatlankereső alkalmazás, mely a piacon lévő legújabb és legaktuálisabban elérhető ingatlanokat jeleníti meg felhasználói számára, ugyanakkor innovatív és intelligens megközelítést kínál az ingatlanvásárlás folyamatához. Az ingatlanvásárlás kihívásokkal teli és időigényes folyamat lehet, különösen a számos ingatlanhirdető oldal és platform széles kínálata miatt. A BudgetHub ezen a ponton lép be a képbe azáltal, hogy egyedi mesterséges intelligencia alapú megoldást kínál a felhasználóknak az ingatlanok hatékonyabb összehasonlítására és kiválasztására.

Erre a jelenségre Brad Anderson is rávilágít egy 2024-es cikkében, amelyben kihangsúlyozza, a mesterséges intelligenciában rejlő potenciált az ingatlanok vásárlása illetve eladása során [1].

A projekt fő célja az ingatlanok kategorizálása és összehasonlítása a különböző ingatlanhirdető oldalakról. Ezt a célt egy speciálisan fejlesztett web scraper alkalmazás teszi lehetővé, amely automatikusan lekéri és strukturált formában összegyűjti az ingatlanokhoz kapcsolódó információkat. A web scraping módszer alkalmazása az ingatlanok pontos és megbízható adatgyűjtését teszi lehetővé. Ez a kulcsfontosságú elem biztosítja, hogy az ingatlanok összehasonlítása ne csak gazdag információkat, hanem kivételes pontosságot is nyújtson a felhasználóknak.

Az általunk fejlesztett mesterséges intelligencia (AI) rendszer lehetővé teszi számunkra, hogy az adatokat hatékonyan feldolgozzuk és az ingatlanokat automatikusan összehasonlítsuk. Napjainkban nagyon gyakran fordul elő, ha valaki el szeretne adni egy ingatlant, több ezzel foglalkozó személyt/irodát is felkeres abból az okból kifolyólag, hogy

meggyorsítsák ezt a folyamatot. Így előfordulhat az az eset, hogy ugyanaz az eladásra mért ingatlan több hirdetésben, esetleg több oldalon is előfordul majd, különböző ajánlatok társaságában. A mi segítségünkkel gyorsan eloszthatjuk ezen felmerülő problémákat, hisz a felhasználók könnyedén kiválaszthatják az optimális ingatlant a kínálatból, és még pénzt is spórolhatnak meg az általunk kínált kimagasló lehetőségek által.

A BudgetHub nem csupán egy ingatlanokat összehasonlító platform, ez egy olyan eszköz, amely segít a felhasználóknak okosabb és gyorsabb döntéseket hozni, miközben időt és pénzt takarítanak meg az ingatlanvásárlás folyamatában.

2. fejezet

Projekt célja

2.1. Az intelligens ingatlankeresés korszaka

A BudgetHub projekt fő célja az ingatlanok kategorizálása és összehasonlítása a különböző ingatlanhirdető oldalakról. Ezt a célt egy speciálisan fejlesztett web scraper alkalmazás teszi lehetővé, amely automatikusan lekéri és strukturált formában összegyűjti az ingatlanokhoz kapcsolódó információkat.

Minél több ingatlant forgalmazó hirdetőfelületről való információszerzéssel, lehetővé tesszük, hogy az oldalunk amellett, hogy rengeteg információval szolgál az ország minden pontjáról, részletes szűrések és rendezések által mindenki megtalálhatja magának azt, amiben érdekelt és amiért az oldalunkat használja.

Az általunk felhasznált mesterséges intelligencia algoritmusok rendszerének segítségével lehetőség van arra, hogy a kívánt ingatlant képfelismerés, illetve összehasonlítás során olyan értékekben szűrje meg, hogy a felhasználó visszakapja azon lehetséges hirdetéseket, melyek megegyeznek a kiválasztott specifikációkkal, vagyis ugyanazon ingat-

lan más hirdetésben való információival fog gazdagodni, így lehetősége lesz választani és összehasonlítani őket.

2.2. A felhasználókért: idő- és pénztakarékosság

A BudgetHub alapvető küldetése az, hogy a felhasználóknak megkönnyítse az ingatlankeresést, kizárólagosan szem előtt tartva az idő- és pénztakarékosságot. Az általunk kifejlesztett web scraper és AI technológia azért szolgál, hogy az ingatlanok összehasonlítása ne csak egyszerű, de rendkívül hatékony és gazdaságos legyen.

Az intelligens megoldások segítségével a BudgetHub nem csupán egy egyszerű ingatlanportál, hanem egy olyan eszköz, amely lehetővé teszi a felhasználóknak, hogy okosabb döntéseket hozzanak az ingatlanvásárlás terén. Időt és pénzt takarítunk meg számukra azzal, hogy az ingatlanokat gyorsan és hatékonyan hasonlítjuk össze, illetve megszűrve az ugyanazon ingatlanokról szóló hirdetéseket, és a saját érdeklődési szempontjaikat figyelembe véve tesszük ezeket, így segítve őket a leoptimálisabb választás megtalálásában.

3. fejezet

Szakirodalom

Az ingatlanpiac egy olyan terület, amely rengeteg kihívást és problémát jelenthet az ingatlankeresők számára. A vásárlóknak számos nehézséggel kell szembenéznük a keresési folyamat során.

Az egyik ilyen nehézség a keresési ciklus megkezdésekor jelentkezik, amikor az illető nem is tudja, hogyan kezdje el az egész folyamatot. Az ingatlanpiac gyors változásai és az ingatlanok széles skálája miatt az emberek gyakran elveszettnek érzik magukat, és bizonytalanok abban, hogy hol és hogyan keressenek megfelelő ingatlant. Emellett a

piaci trendek, az ingatlanárak ingadozása és az ingatlanok minőségi különbségei további kihívásokat jelentenek az ingatlankeresők számára, és megnehezítik a döntéshozatalt.

Kutatásunk első része arra irányult, hogy felmérjük, hogy mennyire relevánsak a weboldalak az ingatlan piacon, és hogy érdemes-e egy weboldalt indítani ingatlanok keresésére és értékesítésére. A Smith és Zietz [2] által végzett kutatás alapján a weboldalak a legtöbbet használt értékes eszközök az ingatlanügynökségek számára az ügyfelek eléréséhez. A tanulmány eredményei azt mutatják, hogy a vásárlók a weboldalakat tartják a leggyorsabb információs forrásnak, és rendkívül értékelik a 24/7 elérhetőséget, az információk szűrésének a lehetőségét és az interaktivitást.

Amy Shair egy 2017-es cikkben [3] bemutatja az online ingatlankeresés előnyeit és hátrányait. A szerző rámutat arra, hogy az internet és online tér segítségével a vásárlók gyorsan és könnyen hozzáférhetnek a piacon lévő ingatlanok széles kínálatához, ami lehetővé teszi számukra, hogy alaposan tájékozódjanak a vásárlás előtt. Azonban, a cikk második felében felhívja a figyelmet az online keresés árnyoldalaira is, mint például a túlzott információmennyiség okozta zavarodottság és az elavult vagy pontatlan hirdetések jelenléte. A cikk hangsúlyozza, hogy bár az online keresés hatékony eszköz lehet, a vásárlóknak körültekintőnek és alaposnak kell lenniük az ingatlanvásárlás során.

Egy másik cikk [4] szerint az online ingatlankeresésnek számos hátránya van, amelyek jelentősen befolyásolhatják a vásárlók élményét és döntéshozatali folyamatát. Ezek közé tartozik a helytelen adatbevitel, ami hibás helyszín megadásához vagy alaprajz hibákhoz vezethet, valamint a félrevezető fotók használata, amelyek torzítják az ingatlan valós állapotát. Továbbá, ez a cikk is hangsúlyozza az elavult információk jelentőségét. Az ilyen téves információk miatt is előfordulhat, hogy az ingatlanokban rejlő lehetőségeket nehéz felismerni, mivel a vásárlók nem mindig kapnak pontos képet az ingatlan aktuális állapotáról.

A Constellation1 [5] által végzett kutatás alapján az ingatlan weboldalak leggyakoribb problémái, a nem megfelelő platform használata, a keresőoptimalizálás és nem megfelelő szűrési lehetőségek hiánya, az alacsony minőségű képek használata. A nem megfelelő platformok hiányosak lehetnek a szükséges funkciók terén. Ilyen amikor egy általános hirdető platformon próbálunk ingatlanokat böngészni, mint például a Facebook Marketplace. Az alacsony minőségű képek és hiányos információk pedig rontják a fel-

használói élményt és a weboldal megbízhatóságát. Ezek a tényezők mind befolyásolják az ingatlan weboldalak hatékonyságát.

Az ingatlanpiac általános problémáinak megértése és azokra való reagálás képezi az alapját a szakirodalom kutatás második részének. A következőkben áttekintésre kerülnek azok a fontos szakirodalmi források, amelyek segítenek az ingatlankeresőknek megérteni az ingatlanpiac működését, és javaslatokat kínálnak a hatékonyabb és eredményesebb keresési stratégiákhoz.

Egy Romániai, ingatlanokat hirdető weboldal felmérése alapján, melyben több ingatlanokkal foglalkozó specialistát kérdeztek meg, megfigyelhető, melyek azok a legfontosabb szempontok, amelyekre kiemelten figyelniük kell a vásárlóknak [6]. Ezen felül, Chihiro Shimizu egy 2016-os kutatásában kitér az ingatlanok változó áaira és az értékesítés azon szakaszaira, melyek befolyásolják ezeket [7].

A Groupama Biztosítótársaság 2022-ben kiadott cikkében olvashatunk, a 10 legfontosabb szempontot érintő témakör, amelyekre fontos odafigyelni és szem előtt tartani, ha ingatlant vásárolnánk [8]. Először is, döntést kell hozni a lakás típusáról, majd meghatározni a rendelkezésre álló költségvetést. Ezután érdemes megvizsgálni az elhelyezkedést, hozzáférhetőséget, és a lakás emeletét és tájolását. Fontos az ingatlan közműellátottsága és az építési minőség. Az ingatlan megjelenése és felszereltsége is számít, valamint a dokumentumok rendben léte. Ha az ingatlan építés alatt áll, érdemes tudni, mikor készül el. Végül, érdemes tájékozódni a fejlesztő háttéréről és a vásárláshoz szükséges dokumentumokról, valamint figyelembe venni azokat a tanácsokat, amelyek segíthetnek a vásárlás folyamatában.

Ezek után az egyik legfontosabb szempont mérlegelése, az ingatlan megvásárlásának célja. A következő megfogalmazás [9] a vásárlási folyamaton túl, részletesen kifejti, hogy bizonyos döntések szempontjából, mely megközelítések az észszerűek. Olyan témákban fejti ki a feltételeit, melyek részletesen kitérnek arra, hogy milyen sorsra lenne szánva a megvásárolt ingatlan. Esetleg családalapítás céljából, befektetés alapú vagy egyéb elgondolásból.

Néhány kutató egy tanulságos dolgozatban megfogalmazza [10] az ingatlankeresők által gyakran elkövetett általános hibákat a vásárlás folyamatában. Ezek közé tartozik a határozatlanság, illetve a túlzott habozás, a rendelkezésre álló költségvetés zavara. Ugyan-

akkor a hirtelen döntések is könnyen félrevezethetik vagy lassítják a teljes folyamatot, illetve az ingatlanszakértőktől vagy jogi tanácsadóktól való segítségkérés elmulasztása is nagy mértékben befolyásolhatja a vásárló döntését [11].

4. fejezet

Követelmény specifikáció

4.1. Felhasználói követelmények

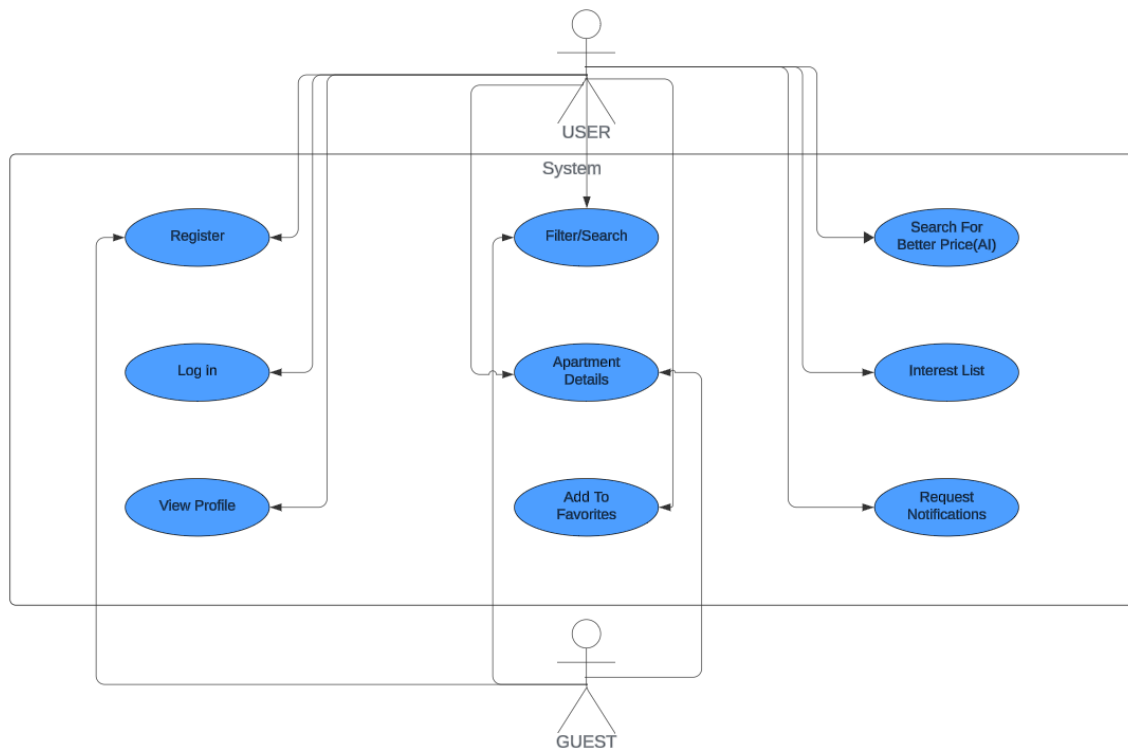
Az alkalmazás használatához nem kötelező a bejelentkezés, hisz aki csak böngészni szeretne vagy kíváncsi az aktuális ingatlanpiaci árakra, az megteheti ezt látogatóként is. Ezen túl, aki regisztrált az oldalra annak rengeteg lehetőségre nyílik alkalma, mint például egy saját kedvencek listájához való jog, ami azt jelenti, hogyha a felhasználónak megtetszik egy ingatlan, lehetősége van ezt bekezdvelni, így hozzáadja ezt a kedvencek listájához. Azonban, ha meggondolta volna magát nyugodtan eltávolíthatja azt, vagy a „Favorites” menü alatt kedvére rendezheti a lista tartalmát.

A felhasználónak lehetősége van a saját adatait megváltoztatni, illetve olyan érdeklőségi köröket állítania, melyek, ha megfelelnek egy bizonyos ingatlan leírásainak, akkor értesítést kap, és így elősegítjük a felhasználót, hogy első kézből értesüljön a legaktuálisabb piacon levő ingatlanokról, melyek iránt érdeklődik.

Az ingatlanok összehasonlítása egy további funkció, mely a felhasználókat olyan lényegi dologban segíti elő, ahol megtekinthetik a kiválasztott ingatlanok közötti különbségeket, lényegében minden szempontból való összehasonlításra nyílik rálátás.

Továbbá egy olyan hirdetőfelület, ahol nem az eladó hirdeti meg lakását, hanem a vásárló jegyzi meg igényeit, mely által a hirdetés tartalmazni fogja azokat az elérhetőségeit és az információkat, melyek iránt érdeklődik a vásárló, így az eladó fél könnyedén elér

őt. Ezzel kedvezni szeretnénk azoknak az eseteknek melyek során az eladó nem szívesen hirdeti eladásra kívánt lakását, de így nyugodtan felveheti a kapcsolatot a vásárlóval, amennyiben megegyeznek a bizonyos feltételek.



4.1. ábra. Use Case Diagram

4.2. Rendszerkövetelmények

4.2.1. Funkcionális követelmények

Ahogy megnyitjuk az alkalmazást rögtön lehetőségünk van a főoldal böngészésére, ahol különféle ajánlatokkal találkozhatunk. Ezeket külön-külön megtekinthetjük és részletesebb képet kaphatunk a hirdetésekről és ezek tartalmáról. Ezen a ponton, ha nem vagyunk bejelentkezve és kedvelni szeretnénk a hirdetést, akkor szükséges a regisztráció vagy bejelentkezés. Regisztráció esetében a rendszer ellenőrzi, és abban az esetben, ha helyesek a beírt információk lementi azokat és egy Refresh Token¹ visszajelzésével elfogadja a regisztrációt. Bejelentkezés esetén ismét figyel a beviteli mezők helyességére, és ha

¹JWT Token: <https://jwt.io/introduction>

megtalálja az adatbázisban a megadott információkat, akkor elfogadja a bejelentkezést és továbblépteti a felhasználót.

A szűrést bármilyen állapotban el lehet érni, viszont a visszkapott listaelemeken ugyanazok a jogosultságok érvényesek akár csak a főoldalon található lista esetében.

A bejelentkezett felhasználó megtekintheti saját adatait, módosíthatja ezeket, illetve törölheti is fiókját. A kedvencek listáját is eléri, melyet kedvére módosíthat. Ugyanakkor lehetősége van az ingatlanok összehasonlítására, melyet ugyancsak bejelentkezve érhet el. A rendszer 60 perc inaktivitást enged meg, azt követően a Token ideje lejár és a felhasználó kijelentkeztetésre kerül.

4.2.2. Nem-funkcionális követelmények

Teljesítmény szempontjából az alkalmazásnak képesnek kell lenni gyorsan kiszolgálni a felhasználókat nagy terhelés esetén is.

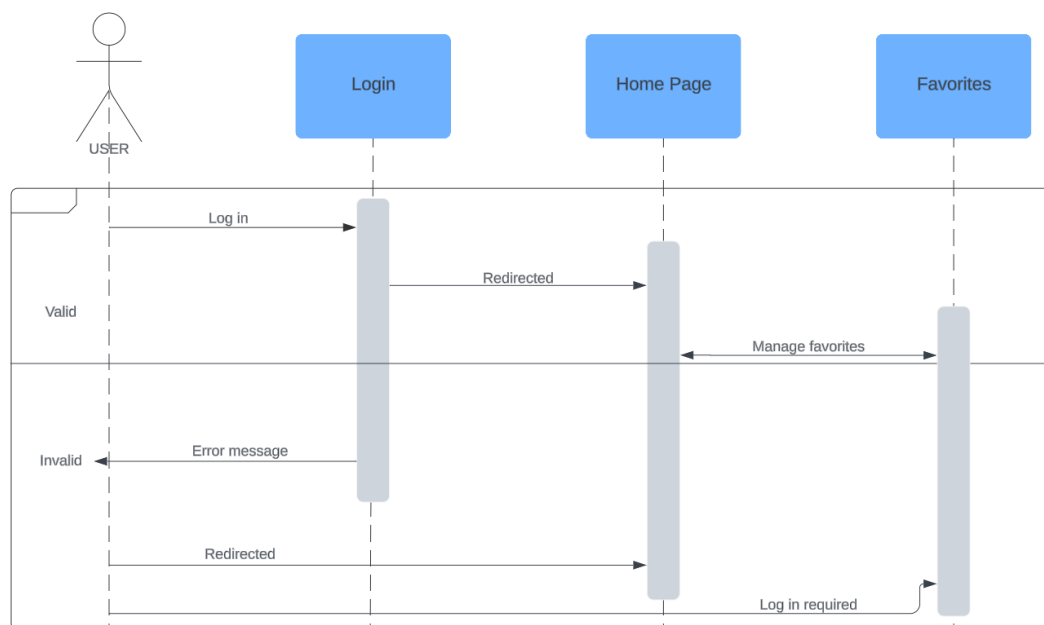
A felhasználói adatokat biztonságosan kell tárolni, jelszavakat megfelelő titkosítással elmenteni és ugyanakkor biztosítani a megfelelő azonosítást és hozzáférést a személyes fiókhoz.

A felhasználói felületnek átláthatónak és egyértelműnek kell lennie, hogy bárki számára könnyen használható legyen.

A rendszernek képesnek kell lennie a folyamatos skálázhatóságra, hogy a növekvő ügyfélkör esetén is képes legyen a gyors és pontos működésre. Az alkalmazásnak folyamatosan elérhetőnek kell lennie, mindig a legfrissebb ingatlan adatokkal.

A mesterséges intelligencia rendszernek nagy pontossággal kell működnie, az azonos ingatlanok meghatározásában, mivel ennek az a célja, hogy biztosítsuk a felhasználók számára a megbízható és pontos összehasonlítási eredményeket az ingatlanok keresése során.

Az alkalmazás használatához olyan eszközre van szükségünk, amelyik képes böngészőt futtatni, mint például számítógép, laptop vagy okostelefon. Emellett stabil internetkapcsolat szükséges az alkalmazás eléréséhez. Böngészőkompatibilitás szempontjából ajánlott a Google Chrome, a Mozilla Firefox vagy az Opera böngésző használata.

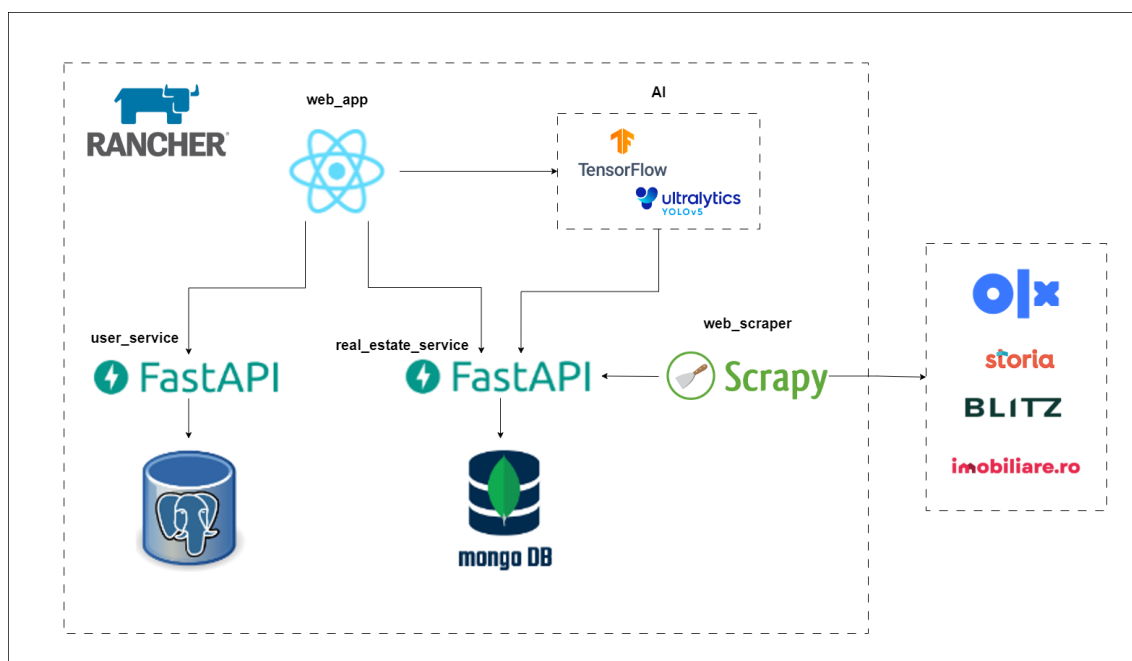


4.2. ábra. Szekvencia Diagram - Felhasználó jogai

A fenti 4.2 diagrammon a felhasználó jogaira vonatkozó leírás látható. A bejelentkezést követően lehetőség adódik a felhasználó saját adatainak megtekintésére, illetve olyan hirdetések mentésére is, amelyek megtetszenek.

5. fejezet

A rendszer architektúrája



5.1. ábra. Rendszer architektúra

Az alkalmazás fejlesztéséhez mikroszerviz¹ architektúra alapú tervezést alkalmaztunk, mert ez a megközelítés számos előnnyel jár a hagyományos monolitikus architektúrával szemben. Mivel a projekt több részből tevődik össze, ezeket külön kell tudnunk választani egymástól a hatékonyabb fejlesztés és a stabil működés érdekében. Ezek a

¹Mikroszerviz architektúra: <https://learn.microsoft.com/hu-hu/azure/architecture/microservices/>

service-ek nagyban eltérnek és fejlesztés szempontjából jobb volt, hogy egymástól független lehetett fejleszteni a részeket.

Az alkalmazásban található 5 service:

- Web scraper
- Real Estate Service
- User Service
- AI
- Web App

A fejlesztés során négy service-t Python nyelven implementáltunk, az ötödiket, a web appot JavaScript nyelv felhasználásával.

Az alkalmazásunk Rancher² platformon fut, amely egy nyílt forráskódú konténerkezelő platform. A Rancher lehetővé teszi a felhasználók számára, hogy egyszerűen kezeljék és futtassák konténereiket, valamint a mikroszolgáltatások közötti kommunikációt is. Ez a platform könnyen skálázható és nagyobb rugalmasságot biztosít az alkalmazások kezelésében és üzemeltetésében egyaránt.

6. fejezet

Felhasznált technológiák

A szoftver fejlesztése során számos különböző technológiát alkalmaztunk, hogy biztosítsuk a rendszer hatékonyságát, skálázhatóságát és megbízhatóságát. A következőkben bemutatjuk a legfontosabb technológiákat a projektre vonatkozóan.

²Rancher: <https://www.rancher.com/learn-the-basics>

6.1. Web scraping

A web scraping egy olyan automatizált adatgyűjtési technika, amely során számítógépes programok (botok) böngésznek az interneten, és strukturált adatokat gyűjtenek össze különböző weboldalakról. Ezen adatokat általában további feldolgozásra, elemzésre vagy tárolásra használják. A web scraping lehetővé teszi a nagy mennyiségű adat hatékony és gyors begyűjtését, és ezt számos területen lehet alkalmazni, mint például piackutatás, ármonitorizálás, hírek és statisztikai kimutatások elemzése. Fontos, hogy a web scraping-et etikusan és törvényesen alkalmazzuk, figyelembe véve a weboldalak használati feltételeit és az adatvédelmi szabályokat.

Egy fontos szabvány, amit követni kell, a robots.txt. Ez egy szabványos fájl, amelyet a weboldalak tulajdonosai készítenek el, hogy elmondják a keresőmotoroknak és más botoknak, hogy mely területek látogathatók vagy nem látogathatók az oldalon. A robots.txt fájl tartalmazza azokat a szabályokat, amelyeket a web scraper botnak követniük kell. Az etikus web scraping során érdemes megvizsgálni a célzott weboldalak robots.txt fájlját, és betartani az ott leírtakat.

```
User-agent: *

Disallow: /ajax/
Disallow: /adminpanel/
Disallow: /ro/login
Disallow: /adprint/
Disallow: /oferta/contact/
Disallow: /plata/
Disallow: /anunt-nou/confirm/
Disallow: /anunt-nou/confirmpage/
Disallow: /api/query

Allow: /api/query?crawl=true
Allow: /
```

6.1. kódrészlet. Példa Storia.ro robots.txt fájlja

Ez volt az alkalmazásunk egyik alapja mivel nekünk nincsenek saját adataink, hanem azokkal az adatokkal dolgozunk, amelyet össze tudunk gyűjteni különböző ingatlan értékesítéssel foglalkozó weboldalakról, ezért ez volt a legelső, amit leimplementáltunk.

A Web scraper szervizünk (szolgáltatásunk) a Scrapy¹ nevű keretrendszerre épül, és az ingatlanok automatizált gyűjtését és feldolgozását végzi különböző oldalakról. Ilyen oldal a Storia és a Blitz.ro. Az alkalmazásunkban kulcsszerepe van a pontos és naprakész ingatlaninformációk biztosításának a felhasználóink számára.

A Scrapy egy nyílt forráskódú, közösségi összefogáson alapuló Python nyelvben íródott keretrendszer, amely lehetőséget nyújt a webes tartalmak strukturált és rendszerezett módon történő gyűjtésére és utó feldolgozására. Főként web scraping célokra alkalmazzák, de felhasználható API hívásokból történő adatok kinyerésére is. Gyors, egyszerű, mégis bővíthető módon képes hatalmas mennyiségű adatot összegyűjteni és feldolgozni.

Léteznek más alternatívák is web scrapingre, például a BeautifulSoup-eszköz, de amíg az csak egy könyvtár, amely egyszerű HTML elemeknél hasznos, addig a Scrapy egy teljes keretrendszer, komplexebb oldalaknál is felhasználható, gyorsabb és jobban skálázható.

Nagy előnyei közé tartozik, hogy egymagában képes nagyon sok extra funkciót beállítani, amivel etikusak maradhatunk web scraping közben, és támogatja az aszinkron működést.

Az egyik hátránya, amivel mi is szembe találoztunk használat közben, hogy nem támogatja a JavaScript elemek feldolgozását, és ezért nem tud minden oldalt teljesen értelmezni, amelyik dinamikusan tölti be a tartalmát. Ezért a Scrapy mellett a Selenium² keretrendszert is használtuk annak érdekében, hogy hatékonyan kezeljük a weboldalak dinamikus tartalmát és JavaScript elemeit.

A Selenium egy nyílt forráskódú keretrendszer és eszközkészlet, amelyet webes alkalmazások tesztelésére és automatizálására terveztek. Fő felhasználási módja automatizált tesztek futtatása a böngészőn keresztül, és felhasználói interakciók szimulálása, amellyel tesztelhető a legtöbb webes alkalmazás. Ez a keretrendszer nyelvfüggetlen, így lehetőséget nyújt sok magas szintű programozási nyelv használatára is. Továbbá kiválóan használható

¹Scrapy: <https://scrapy.org>

²Selenium: <https://www.selenium.dev>

web scraping célokra is, amely úgy működik, hogy létrehoz egy Selenium WebDriver-t, ez szimulál egy böngésző motort, és lehetővé teszi az oldalak tartalmának lekérdezését, és betöltését.

```
import scrapy
from scrapy.selector import Selector
from selenium import webdriver

class BlitzSpider(scrapy.Spider):
    name = "blitz"
    allowed_domains = ["blitz.ro"]
    start_urls = ["https://www.blitz.ro/targu-mures/vanzari-apartamente"]

    def __init__(self):
        self.driver = webdriver.Chrome()

    def parse(self, response):
        for link in response.css('li.card__el div.card__content--head
            a::attr(href)').getall():
            link = f"https://www.blitz.ro{link}"

            yield response.follow(link, callback=self.parse_page)
```

6.2. kódrészlet. Scrapy és Selenium használata projektünkben

Alkalmazásunk web scraping részének hatékonyságát és teljesítményét számos statisztikai adattal mérhetjük. A legutóbbi adatgyűjtés során a Blitz³ és a Storia⁴ weboldalakról csak Marosvásárhelyről összesen 1005, illetve 879 lakás adatát gyűjtöttük össze. Ezek alapján el tudjuk mondani, hogy a mi oldalunkon összesen 1884 ingatlan található meg, ezzel jócskán túlszárnyalva más hasonló profilú oldalakat. Az adatok összegyűjtésének és feldolgozásának időigényét illetően fontos megemlíteni, hogy az adott weboldalak

³Blitz: <https://www.blitz.ro>

⁴Storia: <https://www.storia.ro>

szerkezetétől és teljesítményétől függően változik az összegyűjtési idő, de az adatgyűjtés minden esetben hatékonyan és elfogadható időkereten belül valósult meg.

Az Scrapy keretrendszer lehetővé teszi számunkra, hogy beállítsunk egy időzítést az adatgyűjtés elindításához. Ennek köszönhetően a web scraperünk automatikusan elkezd a működését azokban az időpontokban, amikor a legnagyobb valószínűséggel érkeznek friss ingatlanhirdetések az online platformokra.

Az ingatlanpiac dinamikájának megfigyelése alapján észrevettük, hogy a legtöbb ingatlanhirdetés a délutáni órákban kerül fel az online platformokra. Ezt a tapasztalatot erősíti meg az a két cikk, amelyek az ideális időpontot vizsgálják az ingatlanhirdetések posztolására. Ezen cikkek kiemelik, hogy a délután 5 óra a csúcsideőpont, mivel ekkor az emberek jellemzően hazatérnek a munkából és elkezdnek az ingatlanok keresésével foglalkozni [12], [13]. Diana Olick 2018-as cikkében kifejti, hogy az ingatlanhirdetések kora reggel történő posztolása esetén azok estére általában már a második oldalra kerülnek, ami jelentősen csökkenti az elérésüket.

Ezen információknak a tudatában állítottuk be a web scraperünk, hogy délután 6 órakor induljon el automatikusan az adatgyűjtés, és így délután 7 óra előtt lehetőségünk van a legfrissebb hirdetéseket szolgáltatni a felhasználóinknak.

Az alkalmazás backend részeit két szolgáltatásra osztottuk fel, az egyik a Real Estate Service, amely felelős a web scraper által összegyűjtött ingatlanok eltárolásáért és az ingatlanokkal kapcsolatos műveletekért, a másik a User Service, amely a felhasználók adatainak biztonságos tárolására, a bejelentkezett felhasználók kedvenceinek listában történő tárolására és további különböző felhasználóhoz kapcsolódó műveletek biztosítására használjuk.

A Real Estate Service Python nyelven a FastAPI keretrendszerben készült, annak előnyeit kihasználva. A FastAPI⁵ egy Python nyelvben készült gyors és nagy teljesítményű modern web keretrendszer, amely aktívan tartja a lépést a legújabb Python verziókkal. Ahogyan a neve is sugallja gyorsan fejleszthető API végpontok létrehozására alkalmas.

A FastAPI kiemelkedik a Python webfejlesztési kategóriában, ezért is választottuk ezt a Djangoval és Flaskal szemben, mert a Starlette⁶-re és a Pydantic⁷-ra épülő alapok-

⁵FastAPI: <https://fastapi.tiangolo.com/tutorial/>

⁶Starlette: <https://www.starlette.io>

⁷Pydantic: <https://docs.pydantic.dev/latest/>

nak köszönhetően biztosít masszív programot. Ezt a keretrendszert úgy tervezték, hogy gyors fejlesztést és lenyűgöző sebességet biztosítson. Az egyik jellemzője, hogy hatékonyan minimalizálja az ember által okozott hibákat. Ez nem csak időt képes megspórolni, hanem a kód megbízhatóságának növelésével hiba tűrőbbé teszi az alkalmazásokat. Az egyszerűség jegyében tervezték, így egyszerű szintaxisa, tömör, olvasható kódja és magas szerkesztői támogatása fejlesztőbaráttá teszi, és ez az egyszerűség teszi lehetővé a kevesebb hibát fejlesztés közben. Az FastAPI automatikusan generálja a dokumentációt a kód alapján, ami megkönnyíti a fejlesztést és megnöveli a karbantarthatóságot. A FastAPI betartja az web fejlesztési szabványokat, teljes mértékben kompatibilis az OpenAPI⁸-val (korábbi nevén Swagger) és a JSON Schema⁹-val. Ez a megfelelés biztosítja, hogy a fejlesztők zökkenőmentesen integrálhassák a FastAPI-t meglévő munkafolyamataikba, és kihasználhassák e széles körben elfogadott API-szabványok előnyeit.

A FastAPI alkalmazás elindításához szükség van egy ASGI¹⁰ (Asynchronous Server Gateway Interface) szerverre is, jelen esetben használjuk a Uvicorn¹¹-t. Az Uvicorn egy ASGI webszerver implementáció Pythonhoz amely képes nagyon sok kérést fogadni egyszerre. Pythonban sokáig nem állt rendelkezésre minimális, alacsony szintű szerver interfész az aszinkron keretrendszerekhez. Az ASGI specifikáció pótolja ezt a hiányt, és azt jelenti, hogy most már képesek vagyunk elkezdni egy közös eszközkészlet létrehozását, amely minden aszinkron keretrendszerben használható. Az Uvicorn jelenleg a HTTP protokollt támogatja.

A web scraper által összegyűjtött adatokat egy MongoDB adatbázisban tároljuk. A MongoDB¹² egy nyílt forráskódú dokumentumorientált adatbázis szoftver, amely a NoSQL^[14] adatbázisok közé tartozik, amely rugalmas és skálázható megoldást kínál strukturálatlan adatok tárolására, mivel a dokumentumokat egy cluster-ben JSON-szerű formátumban tárolja. A MongoDB lehetővé teszi különböző struktúrájú és felépítésű adatoknak a könnyű tárolását és indexelését, és ennek az indexelésnek köszönhetően nagyon gyorsan tudjuk elérni az adatokat nagy mennyiségű adathalmazban is.

⁸OpenAPI: <https://swagger.io/specification/>

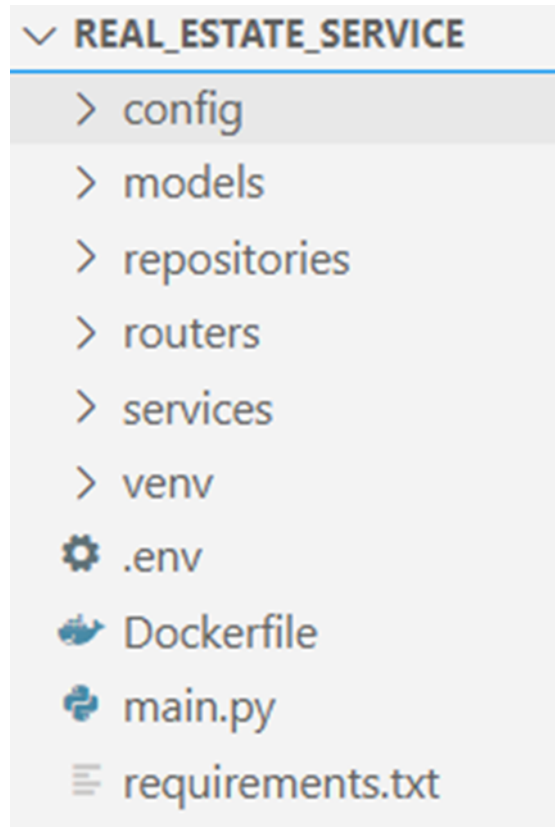
⁹JSON Schema: <https://json-schema.org/learn/getting-started-step-by-step>

¹⁰ASGI Documentation: <https://asgi.readthedocs.io/en/latest/>

¹¹Uvicorn: <https://www.uvicorn.org/>

¹²MongoDB: <https://www.mongodb.com>

A megoldáshoz Clean Architecture-t [15] használtam annak érdekében, hogy a szoftver architektúrája tiszta és jól strukturált legyen. A Clean Architecture olyan tervezési elveket alkalmaz, amelyek segítik az alkalmazás könnyű karbantarthatóságát, bővíthetőségét és tesztelhetőségét. Alapelveit követve a szolgáltatást három fő komponensre osztottam fel: routers, services és repositories.



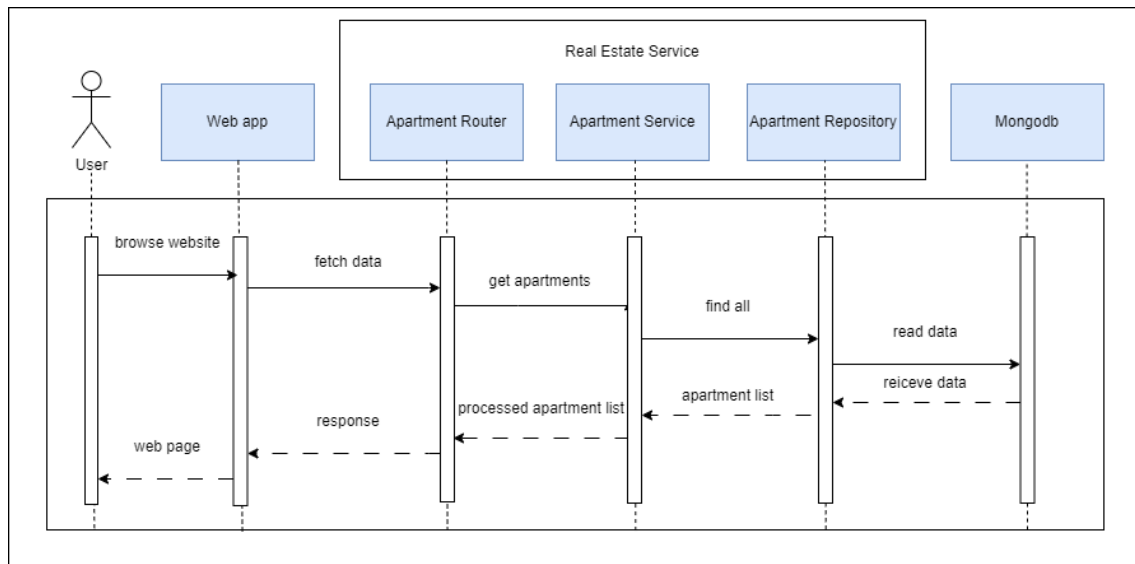
6.1. ábra. Real Estate Service – Séma

A routers réteg a legfelső, ebben készítem a végpontokat, amelyek fogadják a HTTP kéréseket és irányítják azokat a megfelelő szolgáltatásokhoz. Ezek a routerek kezelik a különböző HTTP metódusokat (GET, POST, PUT, DELETE) és biztosítják az URL útvonalak megfelelő kezelését, ezáltal kiszolgálva az alkalmazás frontend részét.

A középső a service réteg, amely felelős az alkalmazás üzleti logikájának végrehajtásáért és a web scraper által összegyűjtött adatok feldolgozásáért. Továbbá ez a réteg létesít kapcsolatot a mesterséges intelligencia projektrésszel, és itt történik meg az adatok előszűrése a lehetséges összehasonlításokhoz.

A legelső szinten a repositories réteg található, amely tartalmazza az adatbázisműveleteket. Itt történik az adatok lekérdezése, mentése, frissítése és törlése a MongoDB adatbázisból. Ennek elkülönítése biztosítja, hogy az adatbázis hozzáférési logikája független legyen az üzleti logikától, ezáltal könnyebb karbantartani és tesztelni az adatbázis műveleteket.

A továbbiakban figyelemmel kísérhetjük az adatok áramlását a 6.2 ábra segítségével.



6.2. ábra. Lakások - adat áramlása

A szolgáltatásban használt modelleket is a Clean Architecture elveinek megfelelően hoztuk létre, biztosítva a különböző adatstruktúrák tiszta és jól definiált reprezentációját. Az egyik modell a helyszínnel kapcsolatos adatokat strukturálja, amiben eltároljuk a megyét, a várost és a zónát.

```

class LocationModel(BaseModel):
    county : str | None = None
    city : str | None = None
    zone : str | None = None
    
```

6.3. kódrészlet. Real Estate Service - LocationModel

A másik modell az ingatlanok különböző tulajdonságait tartalmazza.

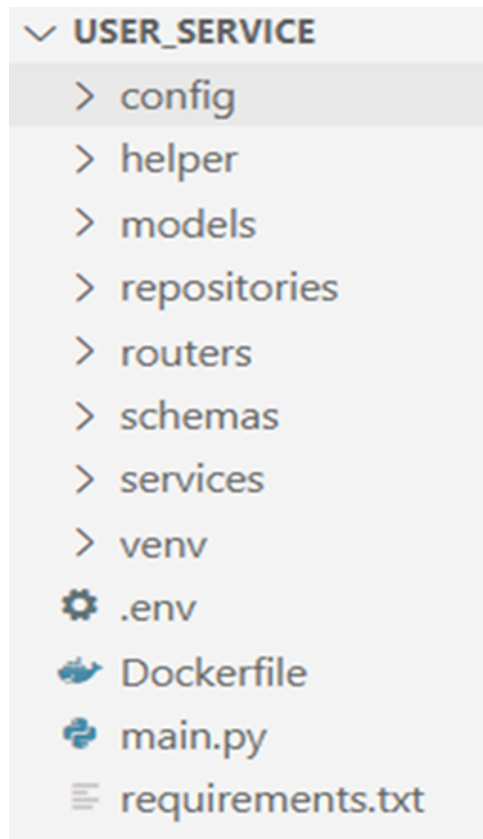
```
class ApartmentModel(BaseModel):  
    id: str | None = None  
    name: str | None = None  
    location: LocationModel  
    price: float | None = None  
    ...  
    surface: float | None = None  
    floor: int | None = None  
    build_year: int | None = None  
    description: str | None = None  
    images: List[str]
```

6.4. kódrészlet. Real Estate Service - ApartmentModel

Egy ingatlannal kapcsolatban sokféle jellemzőt eltárolunk az ingatlan típusától függően. Például egy lakás esetében tároljuk a hirdetés nevét, a leírását, az elhelyezkedését, az árát, a szobák számát, az emeletet, az építési évet és az összes fényképet, amely megtalálható a hirdetésben. Ezek a tulajdonságok relevánsak a lakásokkal kapcsolatban, de előfordulhat, hogy más típusú ingatlanok esetében (pl. iroda épület, ház, telek) más jellegű információkat kell tárolni és feldolgozni.

6.2. User Service

User Service-t Pythonban, FastAPI keretrendszerrel valósítottuk meg, annak pozitívumait hasznosítva. Ez a szolgáltatás felelős a felhasználók adatainak biztonságos tárolásáért. Az alkalmazás továbbá kezeli a bejelentkezett felhasználók kedvenceinek listában történő tárolását, valamint számos egyéb felhasználóhoz kapcsolódó műveletet biztosít. Ezen funkciókat Clean Architecture elv alapján implementáltuk annak érdekében, hogy a szoftver architektúrája átlátható és rendezett legyen, ezáltal javítva a programkód karbantarthatóságát és a tesztelhetőségét.



6.3. ábra. User Service – Fájl architektúra

A három fő komponensre való bontás a 6.3 ábrán is látható.

A routers réteg szolgál a végpontok elkészítésére, amelyek fogadják és kiszolgálják a HTTP kéréseket. Ez a réteg kezeli az útvonalakat és irányítja a bejövő kéréseket a megfelelő Service réteg felé.

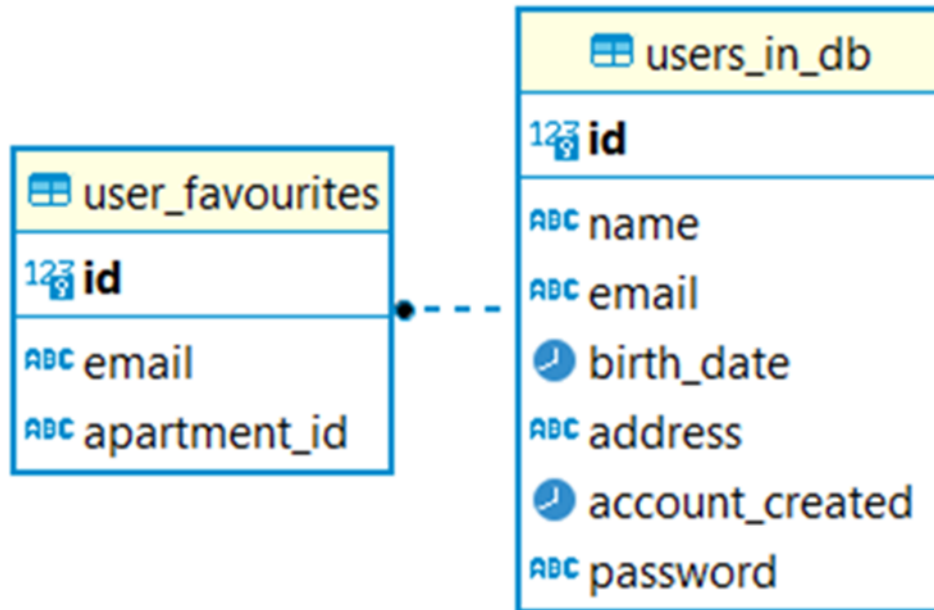
A services réteg központi szerepet tölt be a projektrész működésében. Itt találhatóak azok az osztályok és függvények, amelyek végrehajtják az alkalmazás üzleti logikáját. Ez magában foglalja a felhasználók regisztrációját, bejelentkezését, valamint a kedvenc ingatlanok kezelését. A jelszavakat ebben a rétegben titkosítjuk egy biztonságos encryptáló algoritmus segítségével, amelyeket ezután az oldal validál és tárol. A felhasználók biztonságos azonosításához Token alapú azonosítást alkalmaztunk, ami biztosítja a felhasználói fiókok integritását és a személyes adatok védelmét.

Az repositories réteg felelős az adatbázis műveletek végrehajtásáért. A felhasználói adatokat egy PostgreSQL adatbázisban tároljuk (6.4 ábra) A Postgres¹³ egy nagy

¹³Postgres: <https://www.postgresql.org/docs/>

teljesítményű nyílt forráskódú relációs adatbázis kezelő rendszer, amely az SQL[16] nyelvet használja és bővíti olyan funkciókkal, amelyek elősegítik a biztonságos tárolást és a skálázhatóságot.

A Postgres segítségével könnyen kezelhetőek a felhasználói adatok, és interakciók ezekkel az adatokkal optimalizálhatjuk az adatelérési műveleteket a hatékony működés érdekében.



6.4. ábra. User Service – Séma

Az adatbázis műveletek végrehajtásához SQLAlchemy ORM ¹⁴-t használtunk. Ez az ORM rendszer lehetővé teszi, hogy Python nyelvi objektumok formájában dolgozzunk az adatbázisban található adatokkal, és így egyszerűbb módon kezeljük azokat.

```

def get_user(self, email: str):
    logger.info(f"{TAG} = get_user() - called")
    user = self.db.query(UserInDB).filter(UserInDB.email == email).first()
    return user
  
```

6.5. kódrészlet. User Service - Felhasználó keresése email alapján

¹⁴SQLAlchemy ORM: <https://docs.sqlalchemy.org/en/20/orm/>

A mikroszerviz architektúra előnye, hogy a User Service könnyen cserélhető más technológiával vagy programozási nyelven írt szolgáltatással. Például, ha a jövőben szükség lenne egy alternatív implementációra, például C# .NET technológiában, akkor az architektúrának köszönhetően elég lenne csak ezt kicserélni, anélkül, hogy a többi részt érintené és ugyanúgy működne tovább az alkalmazás.

6.3. Mesterséges intelligencia

Az alkalmazásra való tekintettel, illetve a funkcionalitások alapos megvizsgálását követően, arra véltünk következtetni, hogy ez a fajta megközelítése a képfelismerésnek, objektum-detektálásnak és ezek összehasonlításának egy kifejezetten komplex és kivételes megoldást kíván megvalósításra. Ezen funkciók elérésének érdekében, áttekintettük a rendelkezésre álló lehetőségeket és alternatívákat, melyeket felhasználhatnánk a célunk eléréséhez. A kutatás során olyan problémákba ütköztünk, amelyek nagy részben nem feleltek meg bizonyos feltételeknek, hisz túlságosan általánosak voltak vagy nem biztosítottak elegendő rugalmasságot és testreszabhatóságot, lassú és pontatlan eredménnyel térítettek vissza megoldást, kevés bemeneti adatot fogadtak, de mindegyik esetben a limitált használat vagy az előfizetést igénylő használati feltétel volt a meggátoló tényező. Ennek következtében arra jutottunk, hogy az esetleges függőségek okozta problémák elkerülése miatt stabilabb és megbízhatóbb rendszer kialakítása érdekében saját rendszert hoztunk létre.

Így az általunk különösen erre a célra fejlesztett mesterséges intelligencia rendszer két különálló szerepet betöltő részből áll. Erre azért van szükség, mert az adott probléma hatékony kezelésére és megoldására két technológia kombinációja biztosított kielégítő megoldást.

Az első része a YOLOv5 könyvtár felhasználásával történik, amely kiválóan alkalmas különböző tárgyak felismerésére. Egy mélytanulási modell, mely gyors és pontos objektumfelismerést tesz lehetővé, így ideális választás a feladatunk szempontjából. Ezen felül olyan funkcióval ellátott rendszerrel dolgoztunk, amely lehetővé teszi a tanítást és finomhangolást. Ez azt jelenti, hogy a rendszert további adatokkal taníthatjuk és finomíthatjuk,

optimalizálva annak teljesítményét az aktuális környezethez és feladathoz. A Roboflow¹⁵ platformon található, előre előkészített és tesztelt adathalmazok felhasználásával lehetőségünk volt kiterjeszteni az egység felismerő képességét, mely a kezdetekben nem adott számunkra kielégítő eredményeket.

Az algoritmus egy konvolúciós neuronhálót¹⁶ (CNN) használ az objektumfelismerésre és pozicionálásra a képeken. A konvolúciós neuronhálók hatékonyak a képfelismerésben és más képfeldolgozási feladatokban, mivel képesek a képek jellemzőinek hierarchikus kinyerésére és azoknak az objektumokhoz történő hozzárendelésére. Ezt követően a Yolov5 algoritmus további rétegeket és technikákat alkalmaz az objektumok pontos osztályozására és pozíciójának meghatározására a képen [17].

A bemeneti képek előfeldolgozása előtt az algoritmusnak meg kell változtatnia a képek méretét és formátumát a hálózat által könnyen feldolgozható formátumra. Ez a folyamat az adatok normalizálását és átméretezését foglalja magába.

Az előfeldolgozott képek azután egy sor konvolúciós rétegen mennek keresztül a kép jellemzőinek kinyerése érdekében. Ezek a rétegek a képek különböző aspektusait, például színeket, textúrákat és formákat analizálják.

A konvolúciós rétegek eredményeként létrejönnek a „feature mapok”, amelyek a képek különböző jellemzőit reprezentálják térbeli hierarchiában. Ezek a rétegek végzik az objektumok detektálását. A Yolov5 hálózat az objektumok koordinátáit, osztályát és bizonyossági értékét határozza meg ezeken a rétegeken.

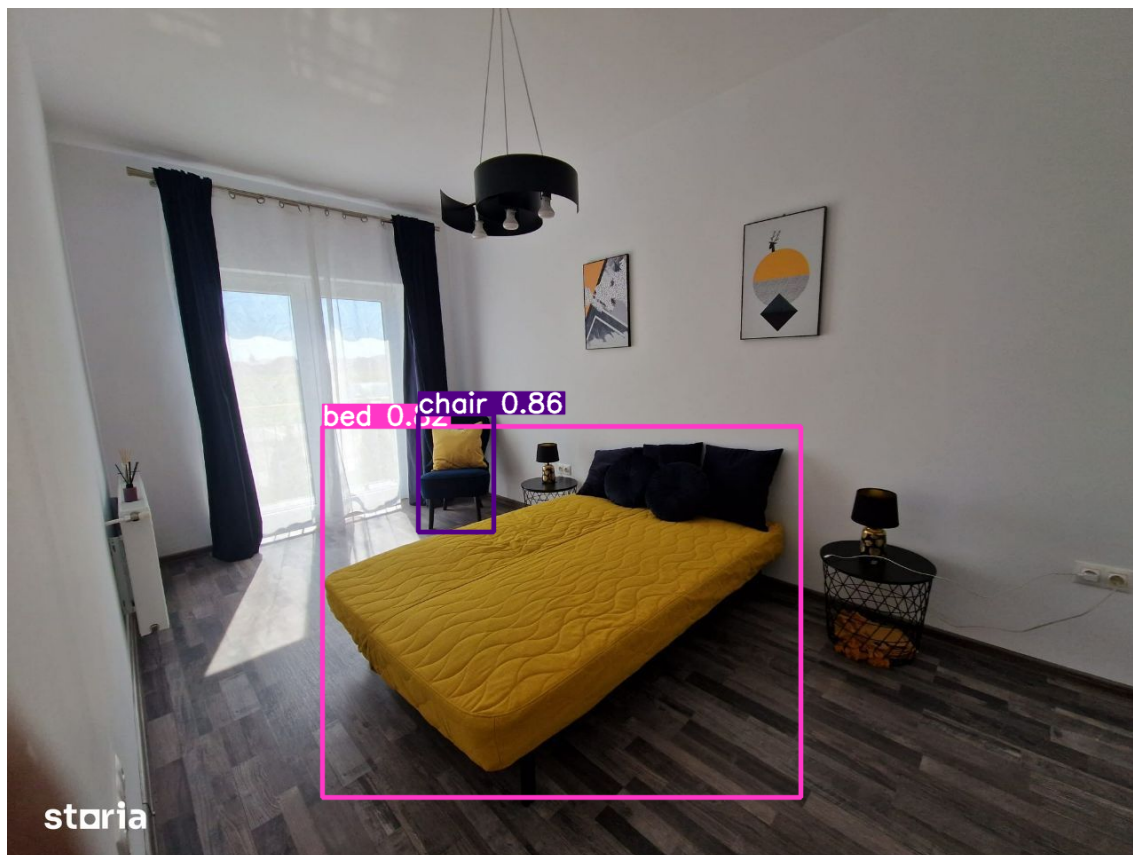
A detektált objektumok közötti redundancia csökkentése érdekében a Yolov5 alkalmazza a „non-maximum suppression” algoritmust. Ez a technika eltávolítja az olyan detektált téglalapokat, amelyek túl sok átfedésben vannak egymással.

Ezen lépések kombinációjával a Yolov5 képes hatékonyan és pontosan detektálni a különböző objektumokat a képeken, és visszaadni azok pozícióját és osztályát. A konvolúciós rétegeknek és az objektumfelismerő rétegeknek köszönhetően az algoritmus képes megbízhatóan kezelni különböző képfelismerési feladatokat ([18],[19],[20]).

¹⁵Roboflow platform: <https://docs.roboflow.com/>

¹⁶Konvolúciós neuronháló: <https://www.ibm.com/topics/convolutional-neural-networks>

Ahogy az alábbi 6.5 ábrán is látható, a beépített¹⁷, tanítás előtti fázisban a rendszer objektumfelismerő képességei nem teljesen alkalmasak az általunk vizsgált környezetben lévő, specifikusan megkülönböztethető tárgyak esetében. Ezért a modell ezen része javításra szorult, a következőkben elvégzett további vizsgálatok és optimálisabb eredmények korrigálásának érdekében.



6.5. ábra. YOLOv5 beépített modell felismerései - szoba

Ezt követően további két modell tanításával igyekeztünk javítani a felismert tárgyak számát és ezek pontosítását. Először egy az ajtókat és azok különböző helyzeteit felismerő adathalmaz¹⁸ segítségével tanítottunk egy modellt, amely ezen tanításnak megfelelően ismeri fel és szolgáltatja vissza az információkat.

¹⁷YOLOv5 beépített modell, "Coco adathalmaz" - <https://www.kaggle.com/code/venkatkumar001/object-recognition-yolov5-coco-dataset>

¹⁸Tanító adathalmaz ajtókra: <https://universe.roboflow.com/mohammed-naji/doors-6g8eb/dataset/1>

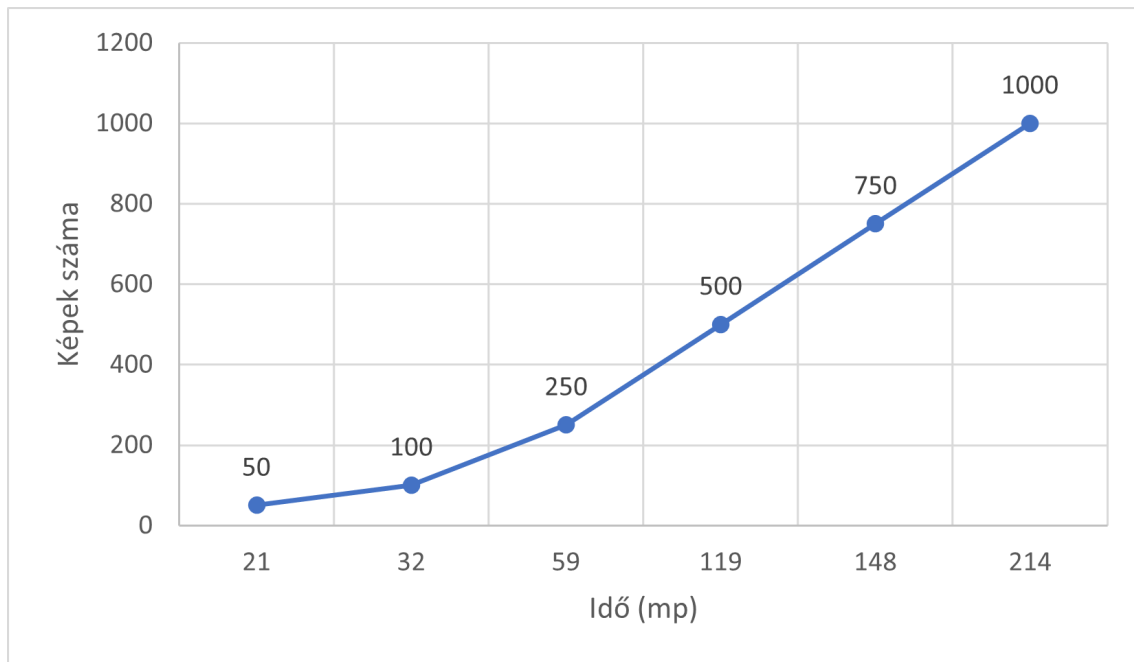
Ezen felül, szükség volt egy olyan modell¹⁹ tanítására is, mely a különböző kellékek és tárgyakat képes felismerni, melyek nem mindegyike található meg a már fent említett, beépített modell osztályai között.



6.6. ábra. YOLOv5 betanított modell felismerései - További tárgyak

A következő számítások a program gyorsasági teljesítményeire vonatkozó eredményeket mutatják be, ahol megfigyelhető, hogy egy képen való vizsgálat mennyi ideig tart és milyen pontossággal számíthatunk. Ugyanakkor a felismert tárgyak száma mindig pontosan megegyező, ami arra a következtetésre utal, hogy a rendszer felismerő képessége megfelelően és kiegyensúlyozottan tanulta meg a kijelölt objektumokat.

¹⁹Tanító adathalmaz különböző tárgyakra: <https://universe.roboflow.com/um-nlhmr/living-room-items/dataset/1>



6.7. ábra. Yolov5 általi felismerések becslése

A rendszer második része a TensorFlow könyvtár, ez a nyílt forráskódú mélytanulási platform lehetővé teszi az objektumok hasonlóságainak precíz ellenőrzését, amely elősegíti a teljes funkció megbízható működését. A TensorFlow Hub kiterjesztett könyvtárának segítségével, különösen a KerasLayer²⁰ formájában, lehetőségünk van a képek különböző alkotóelemeinek és tulajdonságainak megvizsgálására. Ezen egységbe való betöltés során könnyen integrálhatjuk ezeket az előre tanított modulokat a saját Keras modellünkbe, ami nagy rugalmasságot és hatékonyságot biztosít a modellezési folyamat során.

Ennek alapjául a MobileNetV2²¹ tervezési modulhálózat szolgál, amely a TensorFlow Hub része, és kiválóan alkalmas a képek reprezentációjára. Ez egy könnyű és hatékony konvolúciós neurális hálózat (CNN), amelyet az alacsony erőforrású eszközökön való futtatásra terveztek, például mobiltelefonokon vagy beágyazott eszközökön. A MobileNetV2 célja a gyors és pontos képfelismerés, miközben minimalizálja a modell méretét és a számítási erőforrások igényét. A fent említett technológiák alkalmazásával lehetőségünk van a képek vektorokká alakítására, utána pedig könnyedén további feldolgozásnak és skálázásnak vetjük alá őket. Ezen műveletek során a képek olyan formátumot öltenek, amelyek

²⁰KerasLayer mint objektum: https://www.tensorflow.org/hub/api_docs/python/hub/KerasLayer

²¹MobileNetV2: <https://paperswithcode.com/method/mobilenetv2>

könnyen összehasonlíthatóak és hasonlóságukat megfelelően értelmezhetjük. A különböző műveletek alatt, olyan algoritmusokat értünk, melyek segítenek a képek közötti hasonlóság mérésében és értékelésében, mint például az SSE, AVX2, FMA, SSIM és MSE. Ezek az egyszerű és párhuzamos vektorok számításaiban és súlyozási mértékeiben, illetve az objektumok összevetésében játszanak kulcsfontosságú szerepet.

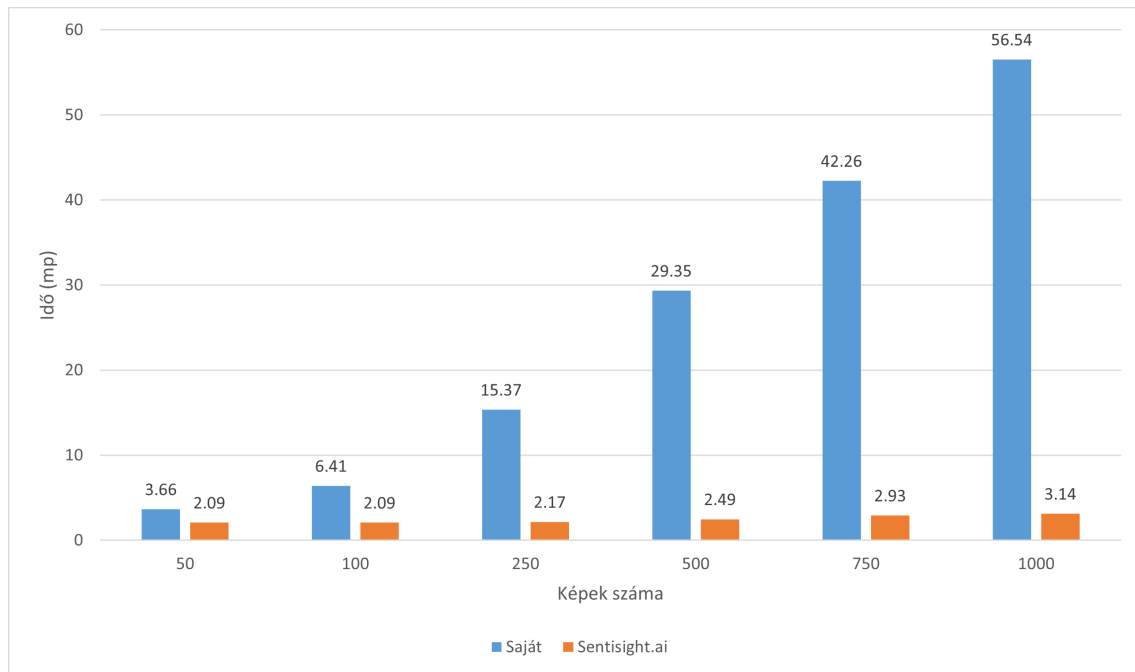
Mivel a rendszer egésze nagyon komplex és kritikus jelentőségű az alkalmazás szempontjából, kiemelkedő fontosságú az ellenőrzési idő javítása és a hatékonyság növelése. Egy olyan rendszerben, ahol nagy mennyiségű adatot kell feldolgozni és összehasonlítani, minden egyes másodperc számít. A rendszer gyorsasága és megbízhatósága közvetlen hatással van az algoritmus működésére és műveletek elvégzésére. Ezért elengedhetetlen, hogy a rendszer képes legyen hatékonyan és gyorsan kezelni a feladatokat anélkül, hogy kompromisszumot kellene kötni a pontosság vagy a megbízhatóság terén.

A következőkben az objektumok összehasonlításában mért számításokat szeretnénk szemléltetni, melyek megmutatják a rendszer ezen pontjának a működési elveit. Ahogyan már említettük, igyekeztünk olyan, már rendelkezésre álló alternatívák bevezetésére, mint például a Sentisight.ai²² platform. Ez a megközelítés lehetőséget nyújt az objektumok összehasonlításának megoldására, viszont korlátozott mennyiségű adat bevitele, illetve a limitált számú ellenőrzés miatt elvetettük ezt a próbálkozást is. Azonban egy olyan kiindulási szempontból megfelelő eszköznek bizonyult, miszerint a platform által szolgáltatott eredményeket összevethetnénk a saját rendszerünk adataival, így átfogóbb képet kapunk további megközelítések komplexitásáról, pontosságáról és futási idejéről, a miénkkel szemben.

A számítások során, egyenként 50, 100, 250, 500, 750 és 1000 képfájl közötti hasonlítást ellenőriztünk, ezen adatok mindegyikében, általunk választott 15 darab objektum állandó volt, a többi pedig egy átlagos háztartáson belüli specifikus tárgyakat tartalmazta. Az állandó adatokra azért volt szükség, hogy kifejezetten egyedi, szélsőséges és különleges esetek megvizsgálását tudjuk biztosítani. Ezeket három egységre bontottuk, 5-5 specifikus kép esetére. Az első egy falióra több szögből való megvizsgálása volt, itt az 5 megegyező kép egyikét ellenőriztük rendre a többivel. A második egy karóra volt, ez

²²Sentisight.ai platform: <https://www.sentisight.ai/solutions/image-similarity-search/>

alkalommal is más-más helyzetből rögzítve, 5 megegyező kép összehasonlítását, viszont alátámasztva azt a tényt, hogy hasonló, de nem megegyező tárgyakat képes megkülönböztetni a rendszerünk. Végül az utolsó 5 tárgy esetén egy vázát vettünk modellnek, amely 2 különböző színben szerepelt, de formájában teljesen megegyeztek, itt az algoritmus színek közötti megkülönböztetésének helyességére szeretnénk volna rávilágítást nyerni, amely során teljesen jó eredményekkel szolgált mindkét modell.



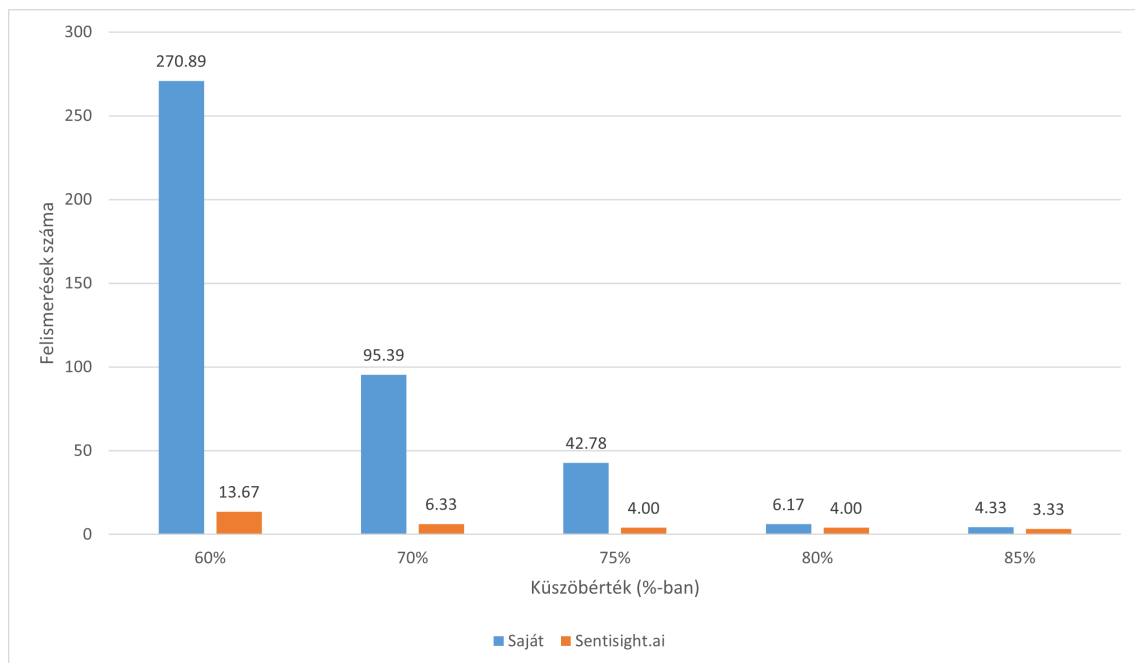
6.8. ábra. Átlag időtartam: Saját vs Sentsight.ai

A fenti 6.8 ábrán megfigyelhető a három egységnyi mérésre vonatkozó átlagok. A saját összehasonlító egység, nagy adatmennyiség mellett gyengébben teljesít az összehasonlított platformmal szemben, azonban az 50-100 darab képek esetében elfogadható eltérés jelenik meg. Ez a küszöbérték lehetőséget ad arra, hogy finomhangoljuk az objektumok hasonlóságának érzékenységét az alkalmazás specifikus igényei szerint, és határozzuk meg azt a határt, amely felett az objektumokat már azonosnak tekintjük. A megfelelő küszöbérték kiválasztása kulcsfontosságú az alkalmazás teljesítményének és megbízhatóságának szempontjából.

Az objektumok hasonlóságának meghatározásában kimagaslóan fontos szerepet játszik a küszöbérték meghatározása, amely egy olyan paraméter, ami meghatározza, hogy

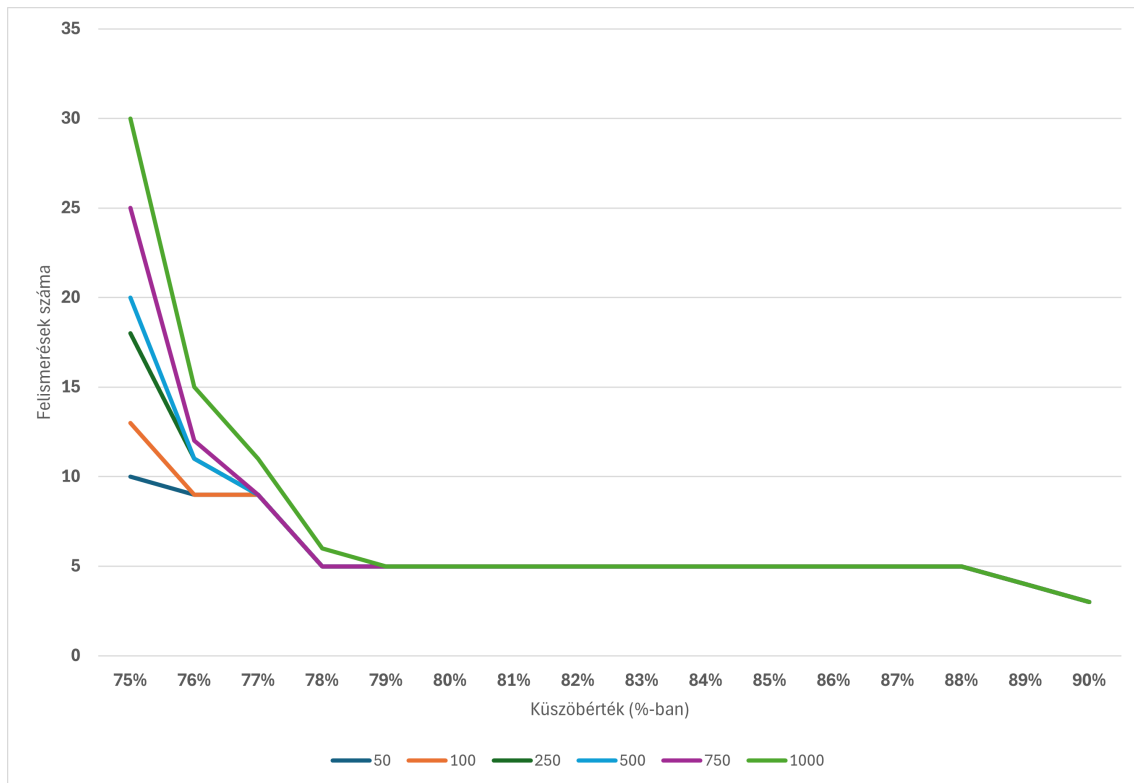
az objektumok milyen mértékben tekintendők hasonlóknak vagy különbözőnek egymáshoz képest. Ez a küszöbérték lehetőséget ad arra, hogy finomhangoljuk az objektumok hasonlóságának érzékenységét az alkalmazás specifikus igényei szerint határozzuk meg azt a határt, amely felett az objektumokat már azonosnak tekintjük. A megfelelő küszöbérték kiválasztása kulcsfontosságú az alkalmazás teljesítményének és megbízhatóságának szempontjából.

A alábbi 6.9 diagrammon látható, hogy a mérések során, mindegyiket 60, 70, 75, 80 illetve 85 százalékos pontossággal mértünk. Ez az alacsonyabb küszöbérték esetén észrevehetően nagy eltérést eredményez a két szemlélet között, azonban ezen érték szigorításával az eredmények is javuló irányt vesznek.



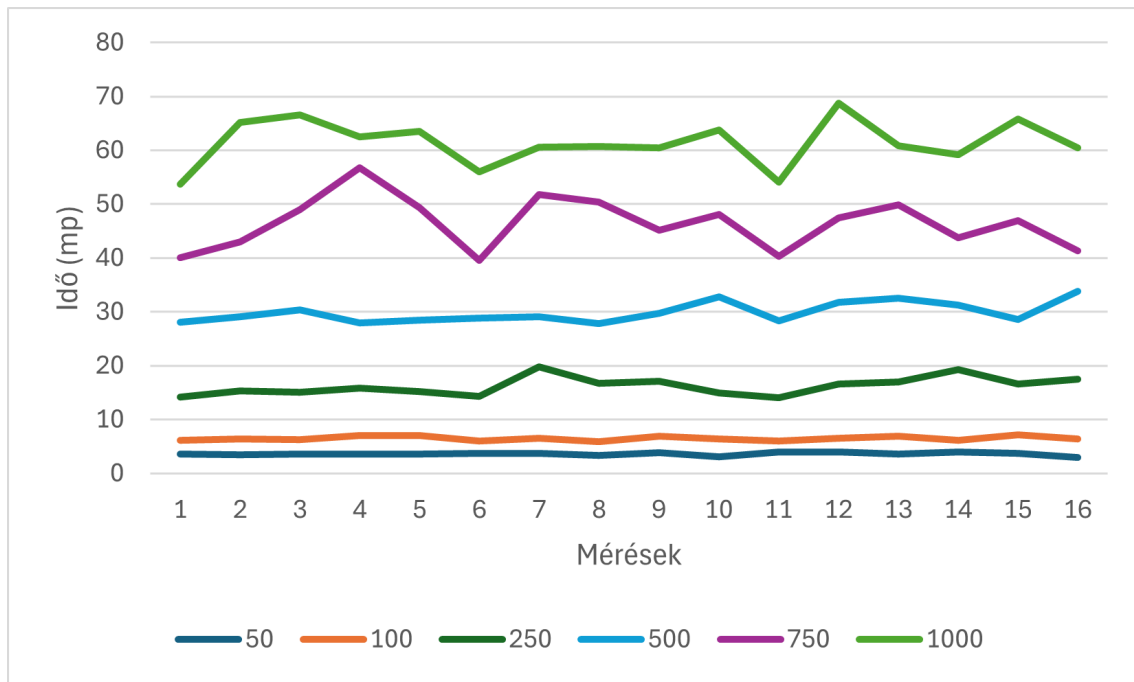
6.9. ábra. Átlag felismerés: Saját vs Sentisight.ai

A továbbiakban egy a küszöbértékekre vonatkozó mérést, a 6.10 ábrán szemléltetünk, amely a túlságosan magas értékek esetén történhet a saját algoritmus méréseiben. A határérték 70-90% közötti eredményeit figyelhetők meg a képek mennyiségének változásai függvényében. Tehát a mérésekből való következtetés alapján, az egység a kisebb méretű adathalmazokat, illetve a 80-87%-os küszöbértékkel ellátott tartományban lévő adatokat képes a leghatékonyabban és legpontosabban meghatározni, illetve felismerni.



6.10. ábra. Átlag felismerés küszöbérték módosításának függvényében: Saját algoritmus

Az algoritmus futási ideje a 6.11 ábrán megfigyelhető, hogy jelentősen nő, ahogy a képek száma növekszik. Ez a növekedés egyértelműen az objektumdetektálás komplexitásának és az adatfeldolgozás terhelésének növekedéséből ered.



6.11. ábra. Átlag időtartam képek mennyiségének módosítása függvényében: Saját algoritmus

Ezen technológiák és módszerek együttes felhasználásával igyekeztünk egy olyan egységes rendszert alkotni, amely négy meghatározó részből áll. Először az előfeldolgozás során az ingatlanokat nagy mértékű szűrés alá vetjük, ami sok időt és műveletet takarít meg a következő lépésekben. Ennek célja, hogy azonos kerületeken belüli, adott árkategórián belül rendelkező és szobák számában megegyező ingatlanokat vessük alá detektálás és összehasonlítás műveleteinek.

Ezt az előszűrt ingatlanok képeinek feldolgozása követi, amely az objektumok detektálására fókuszál a YOLOv5 modellek felhasználásával.

Majd a felismert tárgyak kategorizálása négy csoportba: szoba, konyha, fürdőszoba és közös helyiségek. Ennek a lépésnek a célja, hogy a következőkben ne minden tárgyat hasonlítsunk össze minden másikkal, csak azonos kategóriában lévőkkel, csökkentve az időt és a műveletek számát. Ezenkívül a kategóriákon belül is csak azonos identitású tárgyakat hasonlítunk össze, amivel ugyancsak a hatékonyság növelése volt a cél.

A felismert tárgyak kategorizálása a következő módon történik. Először a YOLOv5 modell által felismert tárgyak nevét és valószínűségét kinyerjük az alábbi, 6.6 kódrészlet segítségével. Látható, hogy minden sor (azaz minden felismert tárgy) esetében egy

szótár (*object_info*) jön létre, amely tartalmazza a tárgy nevét(*name*), a felismerés valószínűségét (*confidence*) és a tárgy pozícióját a képen (*position*). Ezeket az információkat hozzáadjuk az (*objects_with_confidence*) listához, amely tartalmazza az összes felismert tárgyat.

6.6. kódrészlet. A tárgyak nevének, valószínűségének és pozíciójának kinyerése

```
for row in results.pandas().xyxy[0].values:
    object_info = {
        'name': row[-1],
        'confidence': row[4],
        'position': (int(row[0]), int(row[1]), int(row[2]), int(row[3]))
    }
    objects_with_confidence.append(object_info)
```

Ezt követően az összes felismert tárgyat végigjárva kategóriákba rendezzük őket:

6.7. kódrészlet. A felismert tárgyak kategorizálása

```
for obj in all_objects:
    class_name, confidence = obj['name'], obj['confidence']
    for category, class_list in categories.items():
        if class_name in class_list and confidence >= 0.5:
            categorized_objects[category].append(obj)
```

Ebben a kódrészletben minden felismert tárgy esetében ellenőrizzük, hogy a tárgy neve szerepel-e az adott kategóriához tartozó listában, és hogy a felismerés valószínűsége eléri-e vagy meghaladja-e a 0.5-öt. Ha mindkét feltétel teljesül, a tárgyat hozzáadjuk a megfelelő kategóriához a *categorized_objects* szótárban.

Végül az összehasonlítás során visszaszolgáltatót eredményekből kiszámítunk egy valószínűségi értéket, amely az ingatlanok közötti hasonlóságot hordozza magában, ha ez az általunk választott küszöbérték fölött található, akkor megbizonyosodhatunk, hogy egyezés történt.

6.4. Docker

A Docker²³ egy olyan platform és technológia, amely lehetővé teszi alkalmazások konténerizálását és futtatását. A konténerizálás egy módszer, amelyben az alkalmazásokat konténerekbe csomagolják, amelyek minden szükséges szoftvert, függőséget és konfigurációt tartalmaznak az alkalmazás futtatásához. Ezek a konténerek környezetfüggetlenek és könnyen hordozhatóak, ami lehetővé teszi az alkalmazások számára, hogy konzisztensen működjenek többféle környezetben, függetlenül attól, hogy az fejlesztői gép, tesztelő környezet vagy valós felhasználási környezet.

A projektünk esetében hat konténerrel dolgozunk. Ezek közül kettő a FastAPI szolgáltatásokat futtatja, amelyek felelősek az üzleti logika és az adatok kezelésének szempontjából. Egy további konténer a React webalkalmazást futtatja, ami a felhasználói felületet biztosítja. Újabb kettő foglalkozik az adatbázisokkal, amely egy PostgreSQL és egy MongoDB tárolóból áll, végül pedig az AI modellt futtató konténer, ami felelős a képek felismeréséért és azok összehasonlításáért.

Minden szolgáltatásnak készítettünk egy Dockerfile²⁴-t, amelyben leírtuk az utasításokat, hogyan készüljön el a Docker kép²⁵. A Dockerfile egy szöveges fájl, amely szabványokat betartva tartalmaz utasításokat, amelyekkel egy Docker képet hozunk létre. Ez az alapja minden Docker konténernek. Egy Docker kép egy állókép az alkalmazásról, amely tartalmazza az alkalmazáshoz szükséges elemeket, mint például az operációs rendszert, a könyvtárakat, a függőségeket és magát az alkalmazást. A képet használva a Docker konténerek az alkalmazások futtatására szolgálnak.

A Docker konténer²⁶ egy futó példány a Docker képből. A konténer izolált környezetben fut, ami azt jelenti, hogy saját fájlrendszere, hálózata és folyamatai vannak, függetlenül a gazda operációs rendszertől.

A Docker technológiáját a konténerek menedzselésére és kezelésére használják, melyek nagyon könnyűek és gyorsan indíthatók, így minimalizálják az alkalmazások tele-

²³Docker: <https://www.docker.com/>, https://ithub.hu/blog/post/Docker_bevezetes_a_containererek_vilagaba

²⁴Dockerfile: <https://docs.docker.com/reference/dockerfile/>

²⁵Docker kép: <https://docs.docker.com/guides/docker-concepts/the-basics/what-is-an-image/>

²⁶Docker konténer: <https://docs.docker.com/guides/docker-concepts/the-basics/what-is-a-container/>

pítési és futtatási idejét. Emellett a technológia lehetővé teszi az alkalmazások egyszerű skálázását, ami azt jelenti, hogy könnyen növelhető vagy csökkenthető az alkalmazás példányainak száma a terhelés, vagy a felhasználói kereslet alapján. A konténerekben történő futtatás célszerű és biztonságos megoldás, mivel a konténerek teljesen elkülönültek egymástól, ami megakadályozza a biztonsági sérüléseket vagy támadásokat a konténerek közötti áttérjedéstől.

Ezután készítettünk egy docker compose²⁷ fájlt, amely segítségével egyetlen parancs futtatásával létrehozhatjuk és elindíthatjuk az összes konténerem. A Docker compose egy olyan YAML fájl, amely lehetővé teszi több Docker konténer egyszerű definiálását és beállítását. Ebben a fájlban meghatározhatjuk az összes szolgáltatás konténereit, beállítjuk azok függőségeit, hálózati beállításait és egyéb konfigurációit.

6.5. Kubernetes

A Kubernetes²⁸ egy nyílt forráskódú konténeres orchestrációs platform, amelyet elsősorban skálázható, megbízható és rugalmas konténeres alkalmazások kezelésére terveztek. Ez a rendszer lehetővé teszi az alkalmazások hatékony és automatizált elosztását, skálázását, terheléelosztását, valamint a konténerek folyamatos üzemeltetését és felügyeletét.

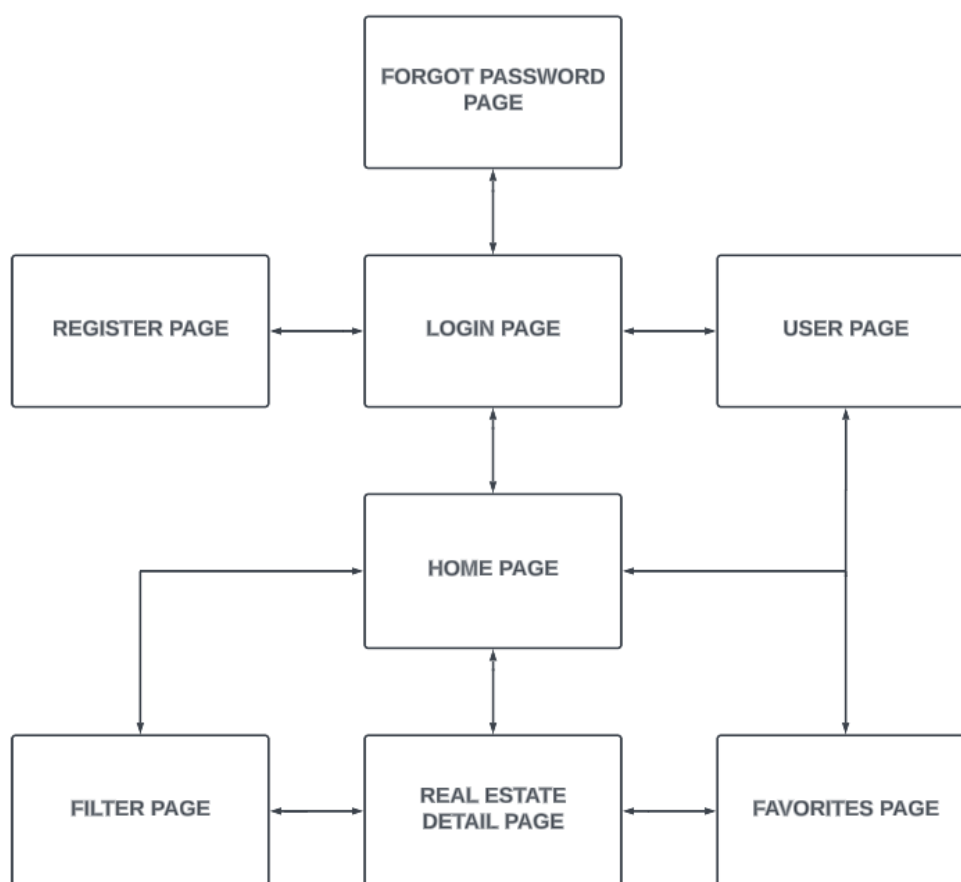
A projektünk üzembe helyezéséhez megalapozottan ezt a technológiát választottuk, mely segítségünkre volt az alkalmazásaink menedzselésére és üzemeltetésére. Ennek oka, hogy a Kubernetes egy olyan erőteljes eszköz, amely lehetővé teszi a különálló egységek gyors és hatékony skálázását a felhasználói kereslet növekedése esetén, emellett automatizált üzemeltetési funkciói révén minimalizálja az emberi beavatkozásra szükséges időt és erőforrásokat, ami hatékonyabbá teszi az alkalmazások menedzselését. Ezen túl lehetővé teszi a platformunk magas rendelkezésre állását, rugalmas skálázását a terhelés függvényében, valamint egyszerű és hatékony módon kezeli az alkalmazások telepítését és frissítését.

²⁷Docker compose: <https://docs.docker.com/compose/>

²⁸Kubernetes: <https://kubernetes.io/>, <https://fotc.com/hu/blog/kubernetes-tutorial/>

7. fejezet

A BudgetHub alkalmazás

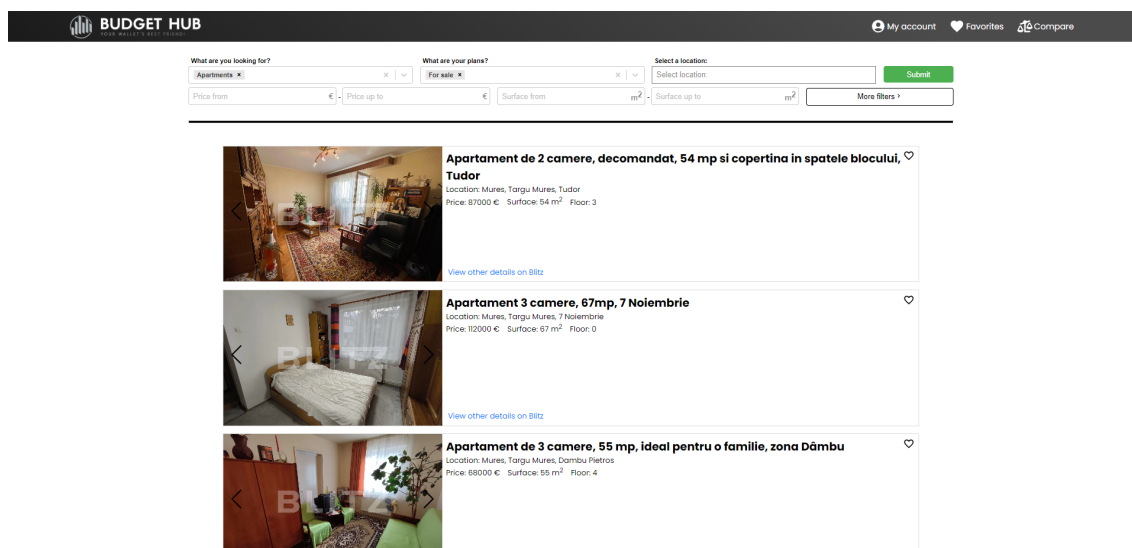


7.1. ábra. Webalkalmazás navigációs diagramja

Az alkalmazás felhasználói felületének navigációs diagramja a 7.1 ábrán látható. Az interfész célja, hogy intuitív és könnyen használható felületet biztosítson a felhasználók

száma, miközben igyekeznek megfelelni ezek igényeinek és a piaci elvárásainak. A diagram bemutatja a szoftver jelenlegi verziójának funkcionalitásait, amelyek folyamatos fejlesztés alatt állnak. Az ábra alapján látható, hogy az alkalmazás különböző oldalai között hogyan történik az elérhetőség és a navigáció. Ennek része a regisztrációs, bejelentkezési, felhasználói profil, kedvencek és az ingatlan részletes információira kiterő oldalak.

Jelenleg a felhasználói felület angol nyelvű, ami számos előnnyel jár. Az angol nyelv széles körben elterjedt és a legtöbb felhasználó számára érthető, ami elősegíti az alkalmazás nemzetközi közönség általi elfogadását és kedvelését. Ezáltal növelhető a potenciális felhasználói bázis és lehetővé teszi az alkalmazás számára, hogy globális piacokon is megjelenjen.

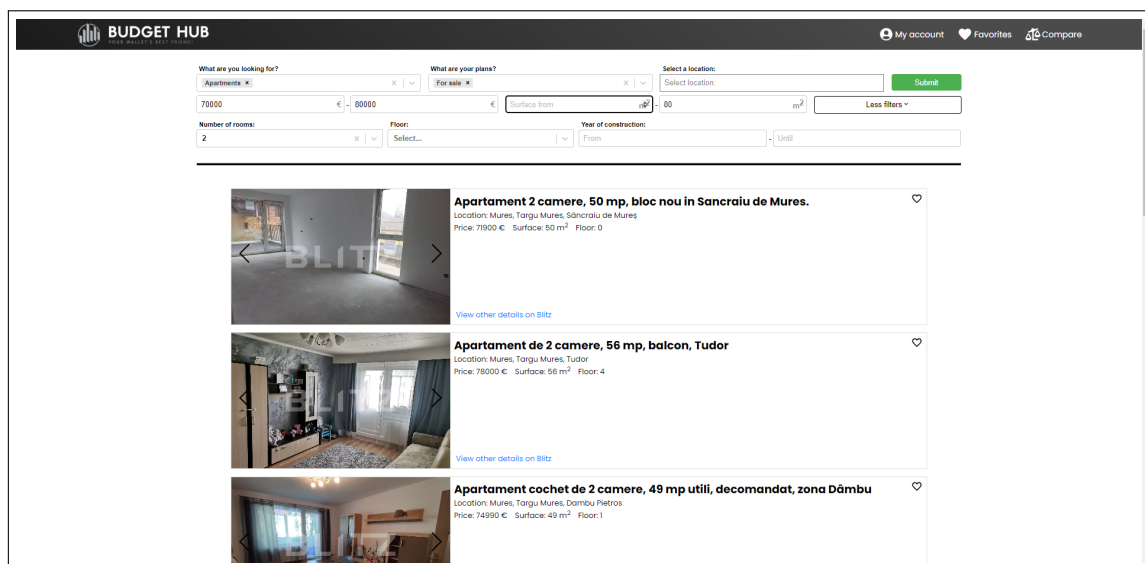


7.2. ábra. Kezdőoldal

A webhely meglátogatása során a 7.2 ábrán látható kezdőoldal - *Home Page* - fogadja a látogatókat, lehetővé téve az azonnali hozzáférést a legfrissebb ingatlanhirdetésekhöz. Az oldal letisztult és átlátható kialakítása segíti a felhasználókat abban, hogy gyorsan és hatékonyan találják meg az érdeklődésüknek megfelelő ingatlanokat.

Az oldal tetején találunk egy navigációs felületet, melynek bal felén az alkalmazás logója kapott helyet, erre bárholnan kattintva a főoldalra navigálja a felhasználót. A jobb felén három lehetőség jelenik meg: a felhasználói fiók adatainak megtekintése, kedvelt hirdetések listája, illetve a fejlesztés alatt álló összehasonlító felület.

Ezt követően a főoldalon egy rendezésre, valamint szűrésre szolgáló felület található, ahol a látogató egyaránt tud majd kedvére keresni és különböző szűréseket végezni. A „Submit” gombra kattintva, az alkalmazás átnavigál arra a felületre, ahol a beállított szűréseket alkalmazva visszakapjuk a megfelelő hirdetéseket - *Filter Page*. A továbbiakban az oldal hasonlóan használható, mint a főoldal.



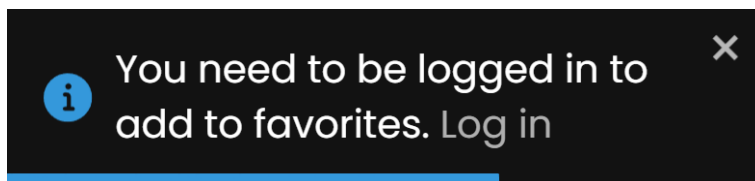
7.3. ábra. Szűrt adatok - Filter Page

A 7.3 ábrán látható példa egy esetleges szűrésre, amelyben kíváncsiak voltunk olyan eladó apartmanok hirdetéseire, amelyek 70.000-80.000 € közötti árban mozognak, méretük legfeljebb 80 m² és két szobával rendelkeznek.

A szűrő felületet követően, megtekinthető az ingatlanhirdetések listája, mely most az összes elérhető ingatlan megjelenítésére szolgál, azonban fontos szerepet játszik majd a dedikált és kiemelt hirdetések helyszínéül is.

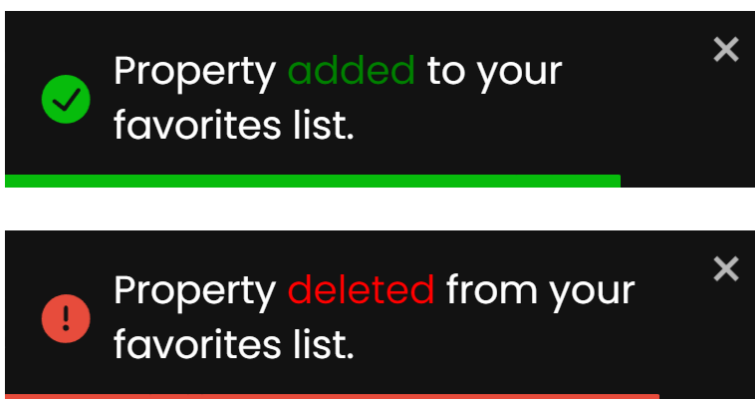
Minden hirdetés egyenként tartalmazza az ingatlan fő leírását, részletes tartózkodási helyét, árát, méretét (m²-ben), illetve az emelet számát amelyen található. Továbbá a hirdetések végén lévő kék megjegyzésre kattintva az oldal átnavigál az ingatlanhirdetés eredeti forrására, ahonnan az származik. Végül pedig a kedvelési funkció, amelyet felhasználói fiókkal rendelkező személy vehet igénybe. Abban az esetben, ha az illető nincs bejelentkezve és menteni szeretné a hirdetést, a 7.4 ábrán látható figyelmeztetés jelenik

meg számára, amely 5 másodpercig látható. Ezen az információs ablakon a „Log in” kiírásra kattintva, átkerül a felhasználó a bejelentkezési oldalra - *Login Page*.



7.4. ábra. Üzenet - Bejelentkezés szükséges

Bejelentkezés után lehetőség nyílik a kedvencek listájához hozzáadni és abból eltávolítani hirdetéseket. Ekkor a következő üzeneteket kapjuk, amelyek ugyancsak 5 másodpercig láthatóak az oldal jobb felső sarkában: (7.5)



7.5. ábra. Üzenet - Kedvencek listájának módosítása

A hirdetések külön-külön kiválaszthatóak és megtekinthetőek, így egy kattintásra az ingatlan adatlapján - *Detail Page* - találjuk magunkat, ahol az ingatlanról egy részletesebb ismertetőt találunk. Itt további tájékoztatást kapunk az ingatlanról, annak elhelyezkedéséről, felszereltségéről és adottságairól. A 7.6 ábrán részletezve az ingatlanról, megtekinthetőek az erről készült képek, ezzel kapcsolatban lehetőség van ezek nagyobb felbontásban való szemügyre vételére is (7.7 ábra). Az ingatlanról készült képek részletes megtekintését követően a nézetből a bal felső sarokban található „X” lenyomásával tudunk visszalépni az adatlapra.




Apartament 3 camere, 89mp, Dambu Pietros

Location: Mures, Targu Mures, Dambu Pietros

Price: 129000 € 1449.44 €/m²

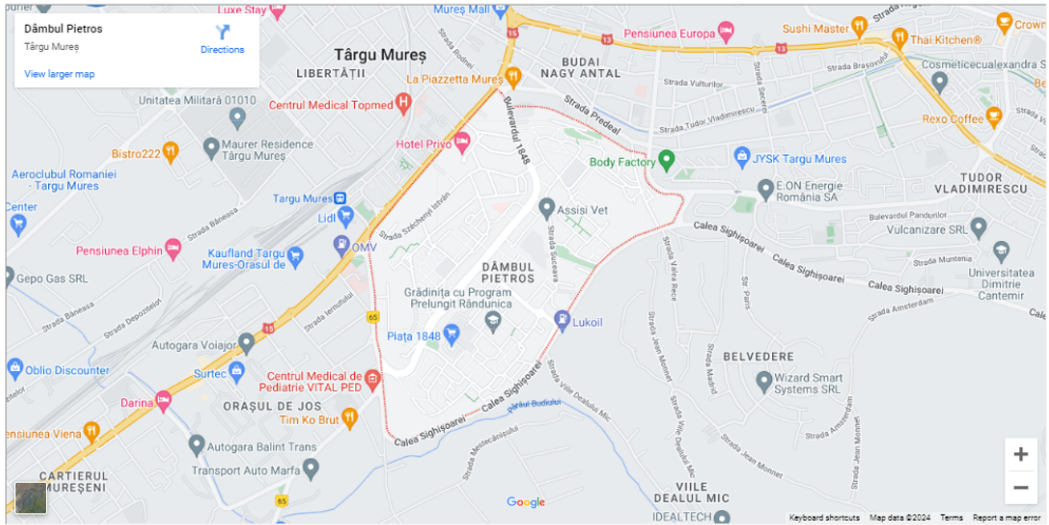
[SEARCH FOR BETTER PRICE](#)

Description

Blitz ofera spre vanzare: apartament cu 3 camere, in zona Dambu Pietros. O zona foarte frumoasa si linistita cu multe facilitati in imediata apropiere precum supermarket, piata de legume-fructe, scoala si gradinita, etc.

General informations

Surface:	89
Number of rooms:	3
Status:	Modern/lux
Floor:	6
Balcony/garden/terrace:	balcon
Year of construction:	2016



7.6. ábra. Ingatlan adatlapja - Detail Page

Egy sokkal átfogóbb és kifejtettebb összefoglaló leírás mutatja be az ingatlan tulajdonságait, ezek között az ingatlan €/m² ára, hány szobával rendelkezik, melyik évben épült stb. Lehetőségünk van ezen a ponton is kedvelni egy hirdetést, a viselkedés meg- egyezik az előzőben leírtakkal. Az információk között található Google Térkép segítségével könnyen beazonosíthatjuk a lakás környezetét, így felmérve az elhelyezkedési adottságo- kat, könnyedén lehet mérlegelni a döntés során.

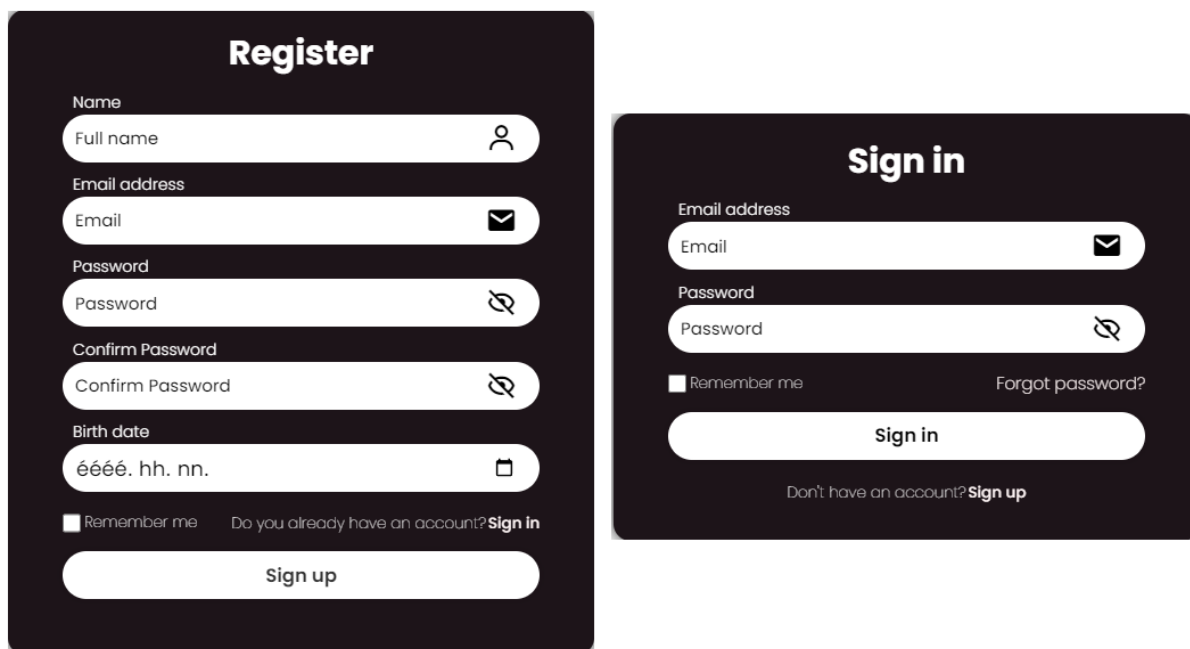


7.7. ábra. Képek nagyobb felbontásban való nézete

A 7.8 ábrán látható regisztráció és bejelentkezés felület segítségével a felhasználók könnyedén létrehozhatják saját fiókjukat és bejelentkezhetnek az oldalra. A regisztrációs űrlap kitöltésekor a felhasználónak először meg kell adnia a teljes nevét, majd egy ér- vényes e-mail címet, amelyet a rendszer ellenőriz a formátum helyességének biztosítása érdekében. A jelszó megadásakor fontos, hogy az megfelelően bizonyos biztonsági követel- ményeknek: legalább nyolc karakter hosszú legyen, tartalmazzon legalább egy nagybetűt és kisbetűt, egy számjegyet illetve egy speciális karaktert, emellett ne tartalmazzon szókö- zöket. A jelszót ezt követően újra be kell írni a megerősítéshez, hogy elkerüljük a gépelési hibákat. Végül a születési dátumot kell megadni év, hónap és nap formátumban, emellett a regisztrálónak legalább 16 évesnek kell lennie.

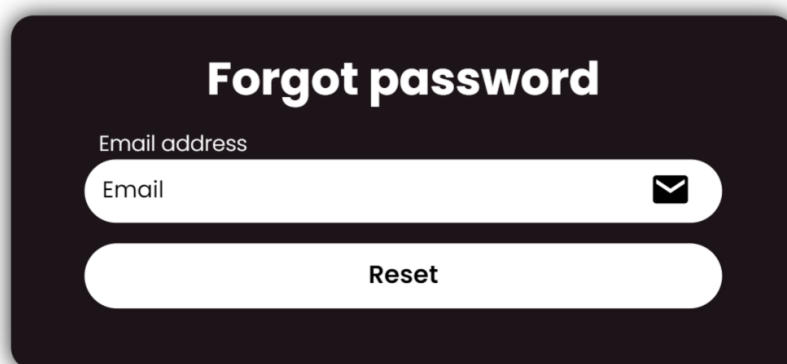
A bejelentkezési űrlap kitöltésekor a felhasználónak meg kell adnia a regisztráció során megadott érvényes e-mail címet és jelszót. Az űrlapon található „Remember me” opcióval a felhasználó elérheti, hogy a rendszer megjegyezze őt a következő bejelentkezés-

kor, így nem kell újra megadnia az e-mail címet és a jelszót. Ha a felhasználó elfelejtette a jelszavát, a „Forgot password” link segítségével könnyedén visszaállíthatja azt egy e-mail segítségével. Az e-mail cím megadását követően a „Reset” gombra kattintva a rendszer elküldi a jelszó visszaállításához szükséges információkat.



The image displays two side-by-side login and registration forms. The 'Register' form on the left includes fields for Name (Full name), Email address (Email), Password, Confirm Password, and Birth date (formatted as éé.éé. hh. nn.). It also features a 'Remember me' checkbox and a 'Sign in' link. The 'Sign in' form on the right includes fields for Email address (Email) and Password, a 'Remember me' checkbox, a 'Forgot password?' link, and a 'Sign in' button. Both forms have a 'Sign up' link at the bottom.

7.8. ábra. Regisztráció és bejelentkezés - Register - Login Page

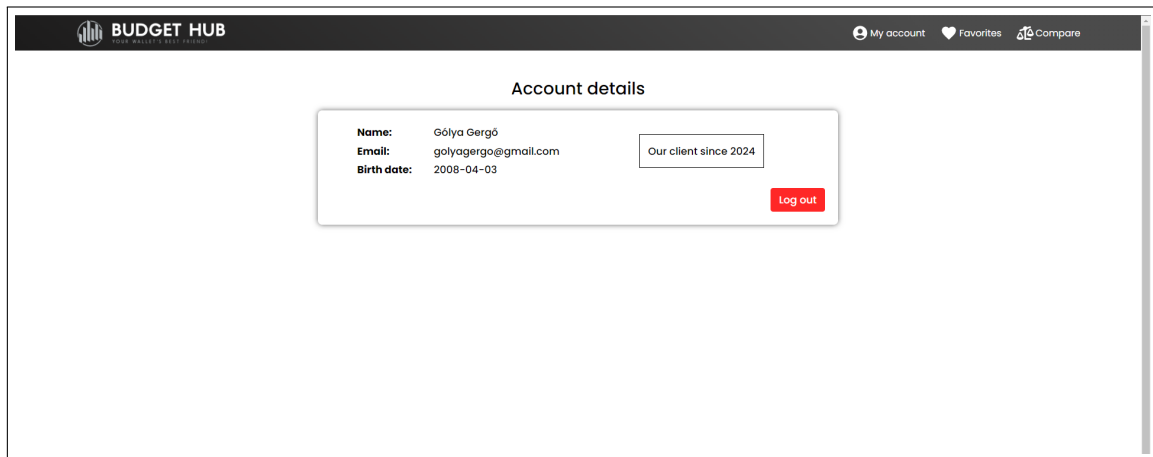


The image shows a 'Forgot password' form. It features a title 'Forgot password' and a single input field for 'Email address' (Email). Below the input field is a 'Reset' button.

7.9. ábra. Elfelejtett jelszó - Forgot Password

A felhasználói fiók létrehozásával az oldalon számos hasznos funkció válik elérhetővé. Például, a kedvencek listájának használatával elmenthetőek az érdekelt hirdetések,

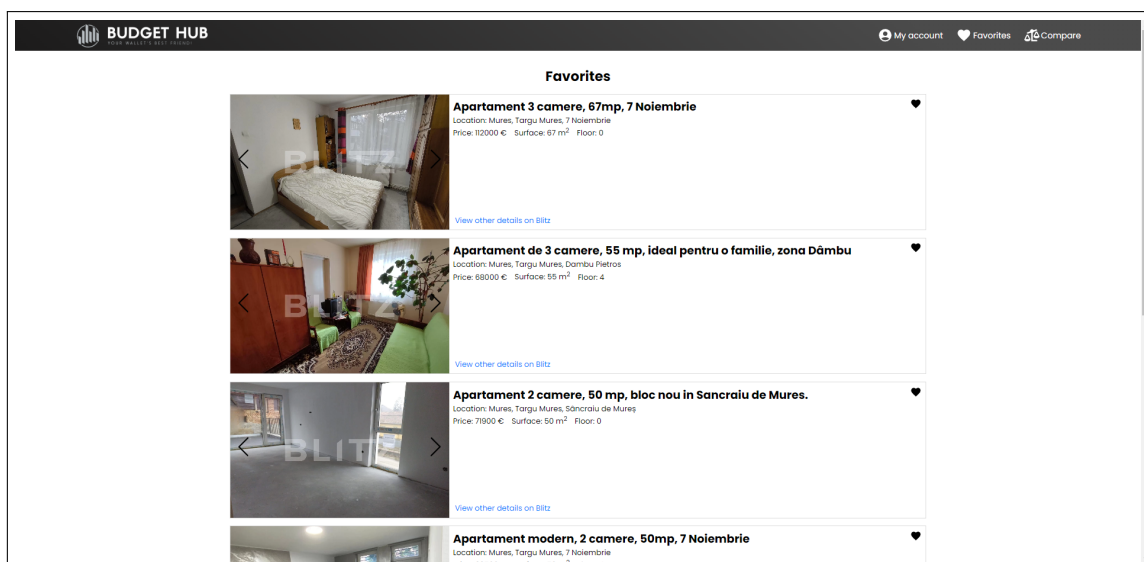
megtekinthetők a felhasználói adatok, beállíthatóak a saját érdekeltségi körök és preferenciák. Igénybe vehető a platform által kínált kiemelkedő mesterséges intelligencia felismerő rendszere is. A felhasználói fiók adatlapjának jelenleg elérhető verziója a 7.10 ábrán figyelhető meg.



7.10. ábra. Felhasználói fiók adatlapja - User Page

A kedvencek oldala - *Favorites Page* - a felhasználók által kedvelt hirdetéseket jeleníti meg. Ezen az oldalon a felhasználók könnyedén hozzáférhetnek azokhoz a hirdetésekhez, amelyeket korábban megjelöltek kedvencekként. Az oldal hasonló funkciókkal rendelkezik, mint a *Home Page* és a *Filter Page*, így a felhasználók itt is egyszerűen navigálhatnak.

Az oldal fő funkciói közé tartozik a hirdetések megjelenítése, ahol a listázott kedvelt hirdetések tartalmazzák a képeket, a hirdetés címét, leírását és egyéb fontos információkat. A hirdetések kezelése is egyszerű, hiszen a felhasználók könnyedén eltávolíthatják a kedvencek közül azokat a bejegyzéseket, amelyek már nem érdeklik őket. Ehhez csupán a hirdetés mellett található szív ikonra kell kattintaniuk, ez azonnal eltávolítja a hirdetést a listából.

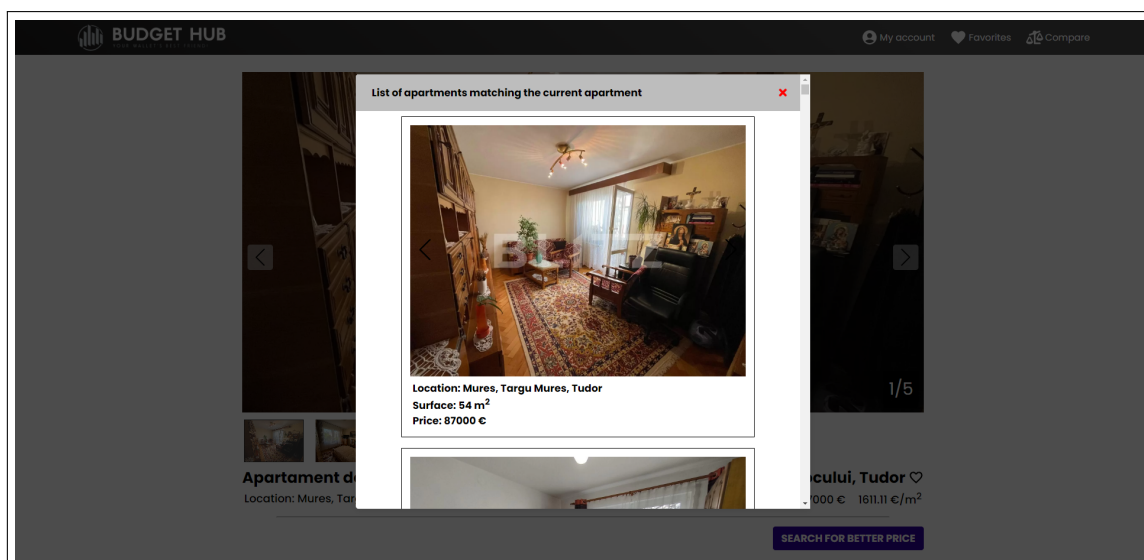


7.11. ábra. Kedvencek listája - Favorites Page

Az alkalmazás fő funkcionalitását figyelembe véve, a mesterséges intelligencia rendszer hasznossága kiemelkedő. A felhasználók számára rendkívül kényelmes és hatékony eszközt biztosít, amely segíti őket a legjobb döntések meghozatalában. Amikor a felhasználó egy hirdetés melletti gombra kattint, egy új ablak jelenik meg, amely kilistázza a potenciálisan azonos lakásról szóló hirdetéseket, ez a 7.12 ábrán látható.

Ez a funkció lehetővé teszi, hogy könnyedén összehasonlítsák az adott ingatlan különböző hirdetéseit, amelyeket különböző forrásokból származhatnak. Az MI rendszer automatikusan elemzi a hirdetések tartalmát, képeit és egyéb fontos jellemzőit, hogy pontosan azonosítsa azokat a hirdetéseket, amelyek ugyanarra a lakásra vonatkoznak.

Ezáltal a felhasználók időt takaríthatnak meg, mivel nem kell manuálisan keresgélniük és összehasonlítaniuk a hirdetéseket. Az MI rendszer segítségével gyorsan és hatékonyan találhatják meg a legjobb ajánlatokat, és biztosak lehetnek abban, hogy nem hagynak ki egyetlen releváns hirdetést sem. A rendszer intelligenciája és pontossága növeli a felhasználói élményt, és segíti a vásárlási folyamat egyszerűbbé és átláthatóbbá tételét.



7.12. ábra. Kedvezőbb ajánlatok listája

Az alkalmazás bemutatása során részletesen megismerkedtünk annak számos hasznos funkciójával és tulajdonságával, amelyek mind a felhasználói élményt hivatottak javítani. Az egyszerű és intuitív felhasználói felület lehetővé teszi a hirdetések könnyed böngészését és szűrését, míg a kedvencek listája segít a legfontosabb hirdetések nyomon követésében és kezelésében.

Kiemelkedő szerepet játszik a mesterséges intelligencia rendszer, amely automatikusan felismeri és összegyűjti az ugyanarról az ingatlanról szóló hirdetéseket, ezzel megkönnyítve a felhasználók számára a legjobb ajánlatok megtalálását. Ez a funkció nemcsak időt takarít meg, hanem biztosítja, hogy a felhasználók ne maradjanak le egyetlen releváns lehetőségről sem.

Összességében az alkalmazás átfogó megoldást nyújt az ingatlanhirdetések kezelésére, összehasonlítására és nyomon követésére. Innovatív funkciói és felhasználóbarát kialakítása révén hatékonyan támogatja a felhasználókat a döntéshozatali folyamatban, és hozzájárul ahhoz, hogy a lakáskeresés folyamata gördülékenyebb és eredményesebb legyen. Az alkalmazás bemutatása rávilágíthat arra, hogy a modern technológia és a mesterséges intelligencia hogyan képes forradalmasítani és megkönnyíteni a mindennapi élet különböző területeinek nehézségeit.

8. fejezet

Összefoglaló

A dolgozatunkban bemutattuk az ingatlanpiac rejtelseit és problémáit, amelyek főként az ingatlankeresés időigényességéből és a platformok széles választékából adódnak. Ezekre a kihívásokra igyekeztünk megoldást nyújtani, amely az idő és pénz megtakarítás jegyében indult, és egy egyezményes platform létrehozásában valósult meg.

Célkitűzéseink közül sikerült megvalósítani az adatok összegyűjtését különböző oldalakról, sikeresen létrehoztuk a saját weboldalunkat, és a felhasznált két mesterséges intelligencia technológiák segítségével elkészítettünk egy algoritmust, amely valós adatokból képes kiszűrni és megtalálni az azonos ingatlanokat.

Úgy gondoljuk, hogy az általunk kidolgozott megoldás forradalmi a hatékony ingatlankeresésben, mivel lehetővé teszi a felhasználók számára, hogy könnyen és gyorsan böngésszenek a rendelkezésre álló ajánlatok között, továbbá az alkalmazás használatával időt és pénzt is képesek megspórolni.

A projekt rengeteg további lehetőségre nyitott. Az ingatlanpiac mint piaci rés, mint társadalmi szempontból véve nagyon komplex és bonyolult fogalom, rengetek további funkcióval, fejlesztéssel és kiegészítéssel fog bővülni, mely a jelenlegi hirdetőoldali piacon megállhatja a helyét.

9. fejezet

Továbbfejlesztési lehetőségek

A *Social Page* ötlete, megtervezése és kivitelezése egy olyan részét képezi a projektnek, amely jelentős mértékben javíthatja a felhasználói élményt, és megoldást nyújthat az ingatlan adásvétel területén felmerülő problémákra. Ez a funkció az ingatlant eladók és vásárlók igényeit egyaránt igyekszik kielégíteni. A folyamat a vásárló kezdeményezésével kezdődik, aki közzétesz egy bejegyzést a felületen. Ez a bejegyzés tartalmazza a vásárló igényeit és preferenciáit, mint például az érdeklődési területet vagy régiót, az ingatlan típusát, az árkategóriát, a méretet és egyéb fontos paramétereket. Emellett a vásárló megadja az elérhetőségeit is, hogy az eladók felvehessék vele a kapcsolatot, amennyiben az általuk kínált ingatlan megfelel a keresett feltételeknek. Az eladók számára lehetőség van a vásárlók bejegyzéseinek szűrésére, így könnyen megtalálhatják azokat a potenciális vevőket, kiknek igényei egyeznek az eladásra kínált ingatlanéval. Ez a rendszer hatékonyabbá és célzottabbá teszi az ingatlanpiaci kommunikációt, és elősegíti az adásvétel gyorsabb lebonyolítását.

Egy jelentős továbbfejlesztési lehetőség az adatgyűjtés kiterjesztése további oldalakra, ezáltal jelentősen növekedne az oldalunkon található ingatlanok száma. Hasznos fejlesztés lenne, az adatgyűjtés kibővítése nem csak lakásokra, hanem házakra és telkekre is. Ez lehetővé tenné, hogy az alkalmazás szélesebb kínálatot nyújtson, lefedve a különböző ingatlantípusok iránti keresletet.

További fejlesztési lehetőségek közé tartozik a szűrési lehetőségek bővítése. Eredményként kapott ingatlanok rendezése. Egy másik MongoDB cluster-ben a felhasználó aktivitásának nyomon követése. A felhasználók interakcióinak rögzítése, továbbá egy rend-

szer kialakítása, amely olyan jellegű ingatlanok ajánlana, melyek relevánsak lehetnek a felhasználó számára.

Account Page-n további funkciók bevezetése, adatok módosítása, előző keresések megjelenítése hirdetések formájában, olyan jellegű ingatlanok ajánlása, melyek felkelthetik a felhasználó figyelmét. Illetve az említett érdeklődési felület kialakítása, implementálása, mely lehetőséget kínál arra az esetre, hogy ha az elérhető hirdetések között akad olyan ingatlan, mely paraméterei nagyban megegyeznek az érdeklődési felületen bejegyzett paraméterekkel, a felhasználó azonnali értesítést kapjon erről. Ez a funkció jelentős időmegtakarítást eredményez, mivel a felhasználónak nem kell folyamatosan keresgélnie, hanem azonnali értesítést kap a számára releváns hirdetésekről.

A *Comparison Page* folytatása, implementálása, megfelelő interakció meghatározása az összehasonlítani kívánt ingatlanokhoz. Itt lesz lehetősége a felhasználónak az előzőleg kiválasztott ingatlanok összehasonlítására.

Detail Page további hirdetéseknek adhat helyet, melyek az aktuálisan kiválasztott ingatlanhoz hasonló leírással vagy tulajdonságaiban véve megegyező kategóriába sorolható.

Az alkalmazás felhasználói felülete több nyelven is elérhető legyen. Ezáltal a rendszer még szélesebb felhasználói kör számára válik hozzáférhetővé, és jobban igazodik a különböző nyelvi háttérrel rendelkező felhasználók igényeihez. A többnyelvű támogatás bevezetésével a célunk, hogy a szoftver globálisan is versenyképes legyen, és minden felhasználó számára kényelmes és érthető felületet biztosítson.

Egy fontos fejlesztési irány az AI rész kiegészítése az üres lakások azonosításában és összehasonlításában, mivel a mostani rendszer a lakásban található tárgyakra összpontosít. Ennek a megoldására bevezetnénk egy másik modellt, amely képes lenne az ingatlanokban található üres terek automatikus felismerésére és elemzésére. Még egy hasznos fejlesztés lenne a lakások leírásából történő információkinyerés, amely tovább mélyítené a rendszer pontosságát.

A rendszer hatékonyságának korszerűsítése érdekében, kiemelkedő fontossággal bír a detektálást és összehasonlítást végző algoritmusok továbbfejlesztése. Ennek részeként tervezzük a detektálás folyamatának kibővítését több különböző modell bevonásával, melyek képesek lesznek még pontosabban azonosítani és kategorizálni az ingatlanokat. Emellett

intenzívebb figyelmet fordítani a tanítási folyamatra, hogy a modellek még jobban illeszkedjenek a konkrét igényeinkhez és a rendelkezésre álló adatokhoz, így magasabb pontosságot érve el az eredményekben.

Az összehasonlítás terén a rendszer finomhangolása és a megerősítő elemek bevezetése elengedhetetlenek a működés hatékonyságának javítása érdekében. Ez magába foglalja az algoritmusok paramétereinek korrigálását, valamint az újabb jellemzők bevezetését a rendszer ezen részébe, hogy még részletesebben és átfogóbban tudja elemezni az ingatlanokat. Ezek a fejlesztések jelentős mértékben hozzájárulnak a rendszer hatékonyságához és teljesítményéhez, így még pontosabb és megbízhatóbb eredményeket biztosíthatok a felhasználók számára.

Továbbiakban igyekszünk olyan személyek véleményét is kikérni a témában, akik több ingatlanértékesítési tapasztalattal rendelkeznek, így lehetőségünk lesz olyan szemszögből rálátni a kisebb, részletesebb dolgokra, melyek mérvadóak lehetnek a következőkben. Illetve megpróbálunk olyan személyekkel is kapcsolatba lépni, akik már találkoztak olyan ingatlanvásárlási problémákkal, melyek megnehezítették, vagy esetleg eltántorították őket a vásárlástól. Ezáltal a felhasználók igényeit és lehetőségeit az oldal minél jobban képes legyen kielégíteni.

Irodalomjegyzék

- [1] Brad Anderson. The impact of ai on buying or selling a house. *ReadWrite*, 2024. <https://readwrite.com/ai-buying-or-selling-house/>.
- [2] Vicky Seiler Michael Bond, Michael Seiler and Ben Blake. Uses of websites for effective real estate marketing. *Journal of Real Estate Portfolio Management*, 6(2):203–211, 2000. <https://doi.org/10.1080/10835547.2000.12089601>.
- [3] Amy Shair. The good, the bad, and the ugly of online house hunting, 2017. <https://www.linkedin.com/pulse/good-bad-ugly-house-hunting-online-amy-shair/>.
- [4] Stacks Property Search. The disadvantages of house-hunting online. 2021. <https://www.stacks.co.uk/difficult-to-find-property/drawbacks-house-hunting-online/>.
- [5] 5 most common problems with real estate websites. *Constellation1*, 2021. <https://www.constellation1.net/5-most-common-problems-with-real-estate-websites/>.
- [6] Hogyan készülj fel a vásárlásra, mik az első lépések. https://www.imobiliare.ro/imoexpert/cum-te-pregatesti-pentru-cumpararea-unui-apartament-3_db/?cq_src=google_ads&cq_cmp=19498160125&cq_term=&cq_plac=&cq_net=x&cq_plt=gp&gad_source=1&gclid=CjwKCAjw_e2wBhAEEiwAyFFFo8NtK9JcaJn5XU4YQo2ey0VfGa97MW9uFTfvawXfV86JseAZqiiG1hoCU_gQAvD_BwE.

- [7] Chihiro Shimizu, Kiyohiko G Nishimura, and Tsutomu Watanabe. House prices at different stages of the buying/selling process. *Regional Science and Urban Economics*, 59:37–53, 2016.
- [8] 10 szempont, ami biztosít arról, hogy helyesen választasz. <https://www.groupama.ro/ghiduri/cumparare-apartament>.
- [9] Ingatlanvásárlás menete, otthonteremtés lépései. <https://tudastar.ingatlan.com/tippek/az-ingatlanvasarlas-menete/>.
- [10] Gary W Eldred. *The 106 Common Mistakes Homebuyers Make (and how to avoid them)*. John Wiley & Sons, 2002.
- [11] Általános hibák, melyeket a vásárlók elkövettek. <https://blog.remax.ro/5-greseli-comune-pe-care-le-fac-cumparatorii-de-case>.
- [12] Diana Olick. This is the best day — and time — to list your home for sale. *cnn*, 2018. <https://www.cnn.com/2018/05/10/this-is-the-best-day--and-time--to-list-your-home-for-sale.html>.
- [13] Dineth Lankaloka. Best time to list a house: When to sell for the most money. *serviceform*, 2023. <https://www.serviceform.com/blog/best-time-to-list-a-house>.
- [14] Nosql adatbázis. <https://cloud.google.com/discover/what-is-nosql>.
- [15] Clean architecture fastapi-ban. <https://theprodev.medium.com/desginig-apis-with-clean-architecture-2fedff1b79cb>, 2022.
- [16] Sql adatbázis. <https://aws.amazon.com/what-is/sql/>.
- [17] Qisong Song, Shaobo Li, Qiang Bai, Jing Yang, Xingxing Zhang, Zhiang Li, and Zhongjing Duan. Object detection method for grasping robot based on improved yolov5. *Micromachines*, 12(11):1273, 2021.
- [18] Yolov5 architektúra - működés. https://docs.ultralytics.com/yolov5/tutorials/architecture_description/.

- [19] Ravpreet Kaur and Sarbjeet Singh. A comprehensive review of object detection with deep learning. *Digital Signal Processing*, 132:103812, 2023.
- [20] YOLOv5 működésének áttekintése. <https://sh-tsang.medium.com/brief-review-yolov5-for-object-detection-84cc6c6a0e3a/>.