

# BudgetHub: Modern és Intelligens Ingatlankeresés

## XXVII. Erdélyi Tudományos Diákköri Konferencia

### Informatika II.: innovatív számítástechnikai termékek, alkalmazások szekció

GÓLYA Gergő, III. év, Sapientia Erdélyi Magyar Tudományegyetem, Marosvásárhelyi Kar, Matematika-Informatika Tanszék

KONCZ Hunor, III. év, Sapientia Erdélyi Magyar Tudományegyetem, Marosvásárhelyi Kar, Matematika-Informatika Tanszék

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

2024

# 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 web-alkalmazás 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, web-alkalmazás, mesterséges intelligencia

# Tartalomjegyzék

<b>1. Bevezető</b>	<b>5</b>
<b>2. Projekt célja</b>	<b>6</b>
2.1. Az intelligens ingatlankeresés korszaka . . . . .	6
2.2. A felhasználókért: idő- és pénztakarékosság . . . . .	7
<b>3. Szakirodalom</b>	<b>7</b>
<b>4. Követelmény specifikáció</b>	<b>9</b>
4.1. Felhasználói követelmények . . . . .	9
4.2. Rendszerkövetelmények . . . . .	10
4.2.1. Funkcionális követelmények . . . . .	10
4.2.2. Nem-funkcionális követelmények . . . . .	11
<b>5. A rendszer architektúrája</b>	<b>13</b>
<b>6. Felhasznált technológiák</b>	<b>14</b>
6.1. Web scraping . . . . .	15
6.2. Real Estate Service . . . . .	18
6.3. User Service . . . . .	20
6.4. React . . . . .	21
6.5. Mesterséges intelligencia . . . . .	22
6.6. Docker . . . . .	31
6.7. Kubernetes . . . . .	32
<b>7. A BudgetHub alkalmazás</b>	<b>33</b>
<b>8. Továbbfejlesztési lehetőségek</b>	<b>37</b>
<b>9. Összefoglaló</b>	<b>38</b>
<b>Irodalomjegyzék</b>	<b>40</b>

# Ábrák jegyzéke

4.1. Use Case Diagram . . . . .	10
4.2. Szekvencia Diagram - Felhasználó jogai . . . . .	12
5.1. Rendszer architektúra . . . . .	13
6.1. Lakások - adat áramlása . . . . .	20
6.2. User Service – Séma . . . . .	21
6.3. Yolov5 beépített modell felismerései - szoba . . . . .	25
6.4. Yolov5 betanított modell felismerései - További tárgyak . . . . .	26
6.5. Yolov5 általi felismerések becslése . . . . .	27
6.6. Átlag időtartam: Saját vs Sentisight.ai . . . . .	29
6.7. Átlag felismerés: Saját vs Sentisight.ai . . . . .	30
7.1. Wireframe diagram . . . . .	33
7.2. Kezdőoldal . . . . .	34
7.3. Oldal az ingatlan részletesebb leírásairól . . . . .	35
7.4. Regisztráció és bejelentkezés oldal . . . . .	36

# Kódrészletek jegyzéke

6.1. Példa Storia.ro robots.txt fájlja . . . . .	15
6.2. Scrapy és Selenium használata projektünkben . . . . .	16

# 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 amellet, 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 hirdetések, 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 felhasználóért: idő- és pénztakarékosság

A BudgetHub alapvető küldetése az, hogy a felhasználóknak megkönnyítse az ingatlan keresést, 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 legoptimálisabb választás megtalálásában.

# 3. fejezet

## Szakirodalom

Az ingatlanpiac egy olyan terület, amely rengeteg kihívást és problémát jelenthet az ingatlan keresők 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 elveszítettnek 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.

Az ingatlanpiac általános problémáinak megértése és azokra való reagálás képezi az alapját a szakirodalom kutatás ezen 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 [2]. Ezen felül, Chihiro Shimizu egy 2016-os kutatásában kitér az ingatlanok változó áaira és az értékesítés azon szakaszaira, melyek befolyásolják ezeket [3].

A Groupama Biztosítótársaság 2022-ben kiadott cikkében olvashatunk, a 10 legfontosabb szempontot érintő témakör, amelyekre fontos odafigyelni és szem előtt tartani, ha ingatlant vásárolnánk [4]. 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 [5] 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 [6] 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 [7].

## 4. fejezet

# Követelmény specifikáció

### 4.1. Felhasználói követelmények

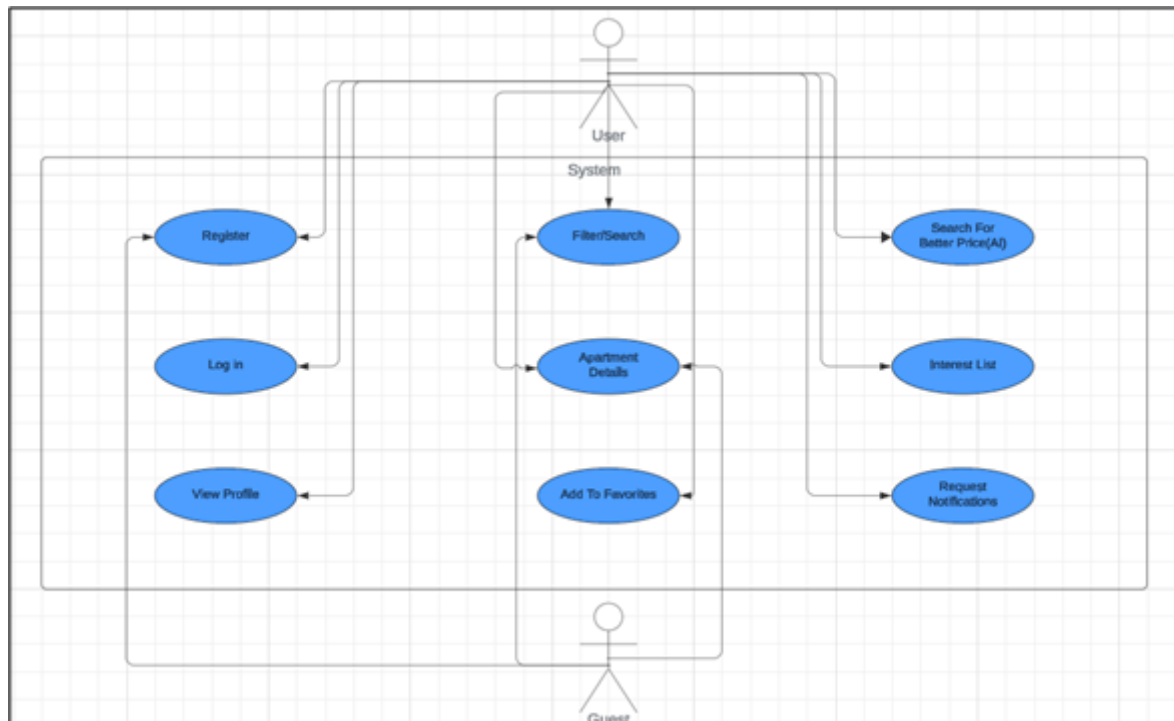
Az alkalmazás használatához nem kötelező a bejelentkezés, hisz aki csak böngészni szeretne vagy kíváncsi az aktuális ingatlanpiaci árakra, az megteheti ezt látogatóként is. Ezen túl, aki regisztrált az oldalra annak rengeteg lehetőségre nyílik alkalma, mint például egy saját kedvencek listájához való 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 eléri

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 hirdetéséről 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 Token<sup>1</sup> visszajelzésével elfogadja a regisztrációt. Bejelentkezés esetén ismét figyel a beviteli mezők helyességére, és ha

<sup>1</sup>JWT 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 visszakapott 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.

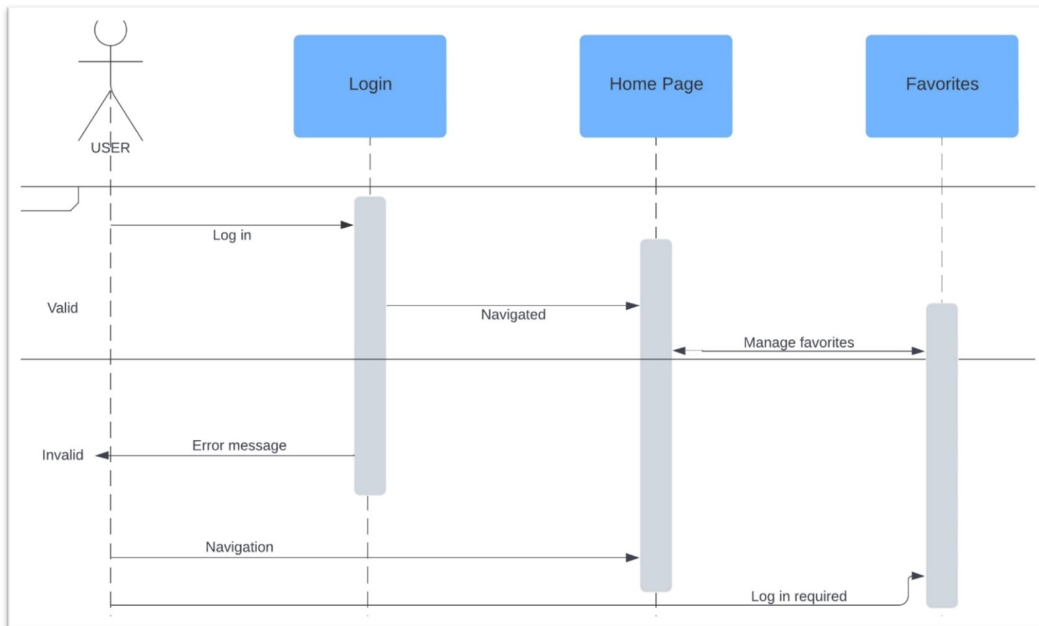
A felhasználói adatokat biztonságosan kell tárolni, jelszavakat megfelelő titkosítással elmenteni és ugyanakkor biztosítani a megfelelő azonosítást és hozzáférést a személyes fiókhoz.

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.

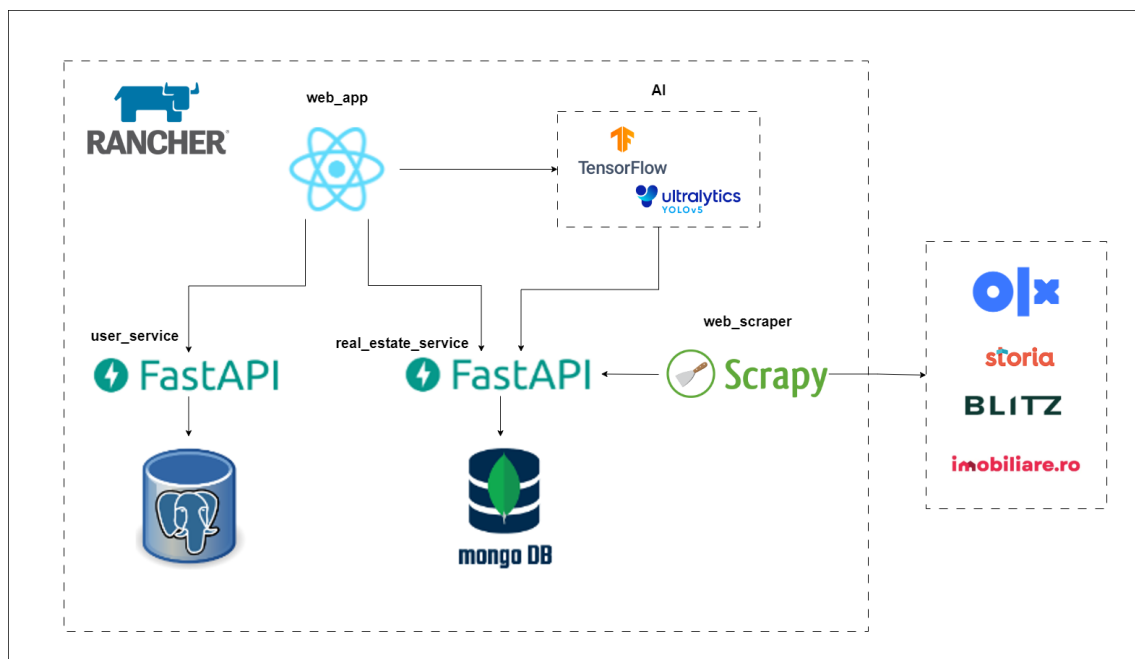


**4.2. ábra.** Szekvencia Diagram - Felhasználó jogai

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 mikroszerviz<sup>1</sup> 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

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

service-ek nagyban eltérnek és fejlesztés szempontjából jobb volt, hogy egymástól függetlenül 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 Rancher<sup>2</sup> 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.

---

<sup>2</sup>Rancher: <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 Scrapy<sup>1</sup> 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 alkalmazták, de felhasználható API hívásokból történő adatok kinyerésére is. Gyors, egyszerű, mégis bővíthető módon képes hatalmas mennyiségű adatot összegyűjteni és feldolgozni.

Léteznek más alternatívák is, 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. Van egy hátránya, amivel mi is szembesültünk használat közben, hogy nem támogatja a JavaScript elemek feldolgozását. Ezért a Scrapy mellett a Selenium<sup>2</sup> 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 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
```

---

<sup>1</sup>Scrapy: <https://scrapy.org>

<sup>2</sup>Selenium: <https://www.selenium.dev>



```

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 Blitz<sup>3</sup> és a Storia<sup>4</sup> 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 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 elkezdi a működését azokban az időpontokban, amikor a legnagyobb valószínűséggel érkeznek friss ingatlanhirdetések az online platformokra.

<sup>3</sup>Blitz: <https://www.blitz.ro>

<sup>4</sup>Storia: <https://www.storia.ro>

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 csúcspont, mivel ekkor az emberek jellemzően hazatérnek a munkából és elkezdnek az ingatlanok keresésével foglalkozni [8], [9]. 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.

## 6.2. Real Estate Service

Az alkalmazás backend részeit két service-re 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<sup>5</sup> keretrendszerben készült, annak előnyeit kihasználva biztosítja a gyors és hatékony API végpontokat, amelyeken keresztül a frontend eléri az adatokat. Ez a szolgáltatás felelős az ingatlanokkal kapcsolatos lekérések és műveletek végrehajtásáért, az ingatlanok eltárolásáért, biztosítva a pontos és gyors válaszokat a webes alkalmazás számára.

Azért választottuk a FastAPI-t a Djangoval és Flaskal szemben, mert a FastAPI kiemelkedő teljesítményt, aszinkron működést és modern Python szintaxist kínál, beleértve a beépített típusellenőrzést és naprakészen a legújabb Python funkciókat.

---

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

Mindkét service alkalmazását az Unicorn<sup>6</sup> szerverrel indítjuk el, amely támogatja a FastAPI alkalmazásokat, és lehetővé teszi a hatékony és aszinkron működést, és képes nagyon sok kérést fogadni egyszerre.

Clean Architektúrában<sup>7</sup> implementáltuk annak érdekében, hogy a szoftver architektúrája tiszta és jól strukturált legyen. A Clean Architektúra olyan tervezési elveket alkalmaz, amelyek segítik az alkalmazás könnyű karbantarthatóságát, bővíthetőségét és tesztelhetőségét.

A web scraper által összegyűjtött adatokat egy MongoDB adatbázisban tároljuk, vagyis pontosabban egy clusterben mert a MongoDB<sup>8</sup> egy NoSQL adatbázis, amely rugalmas és skálázható megoldást kínál strukturálatlan adatok tárolására. A web scraper által begyűjtött ingatlaninformációk változó szerkezetűek, és a MongoDB lehetővé teszi ezeknek az adatoknak a könnyű tárolását és indexelését, és ennek az indexelésnek köszönhetően nagyon gyorsan tudjuk elérni az adatokat nagy mennyiségű adathalmazban is.

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 más típusú ingatlanok esetében más jellegű információkat kell tárolnunk és feldolgoznunk.

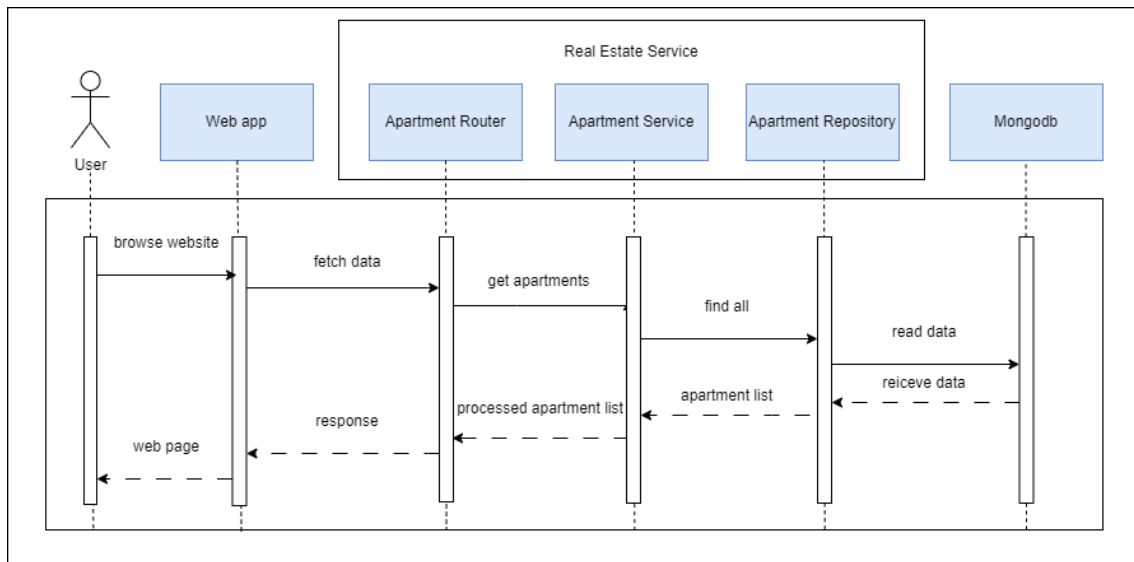
Továbbá figyelemmel kísérhetjük az adatok áramlását a 6.1 ábra segítségével.

---

<sup>6</sup>Uvicorn: <https://www.uvicorn.org/>

<sup>7</sup>Clean Architektúra FastAPI-ban: <https://theprodev.medium.com/desgining-apis-with-clean-architecture-2fedff1b79cb>

<sup>8</sup>MongoDB: <https://www.mongodb.com>



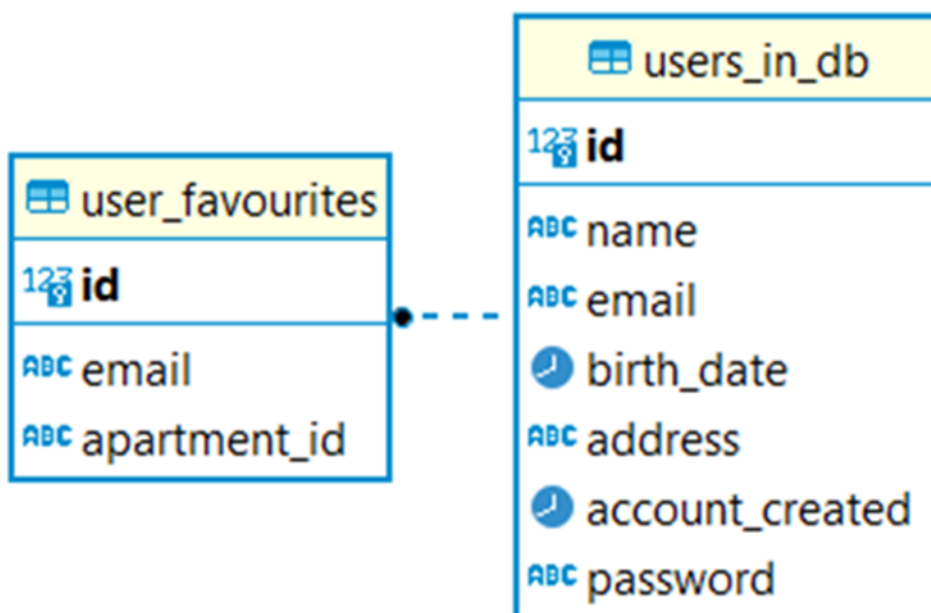
6.1. ábra. Lakások - adat áramlása

### 6.3. User Service

User Service-t Pythonban, FastAPI keretrendszerrel valósítottuk meg, annak előnyeit kihasználva biztosítja a gyors és hatékony API végpontokat a frontend számára. Ez a szerviz 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 Architektúrában implementáltuk annak érdekében, hogy a szoftverarchitektúra tiszta és jól strukturált legyen, ezáltal javítva a karbantarthatóságot és a tesztelhetőséget.

Az alkalmazás a felhasználói adatokat egy Postgres<sup>9</sup> adatbázisban tárolja, ami egy hatékony, megbízható és skálázható relációs adatbázis. A Postgres segítségével könnyen kezelhetjük a felhasználói adatokat, és optimalizálhatjuk az adatelérési műveleteket a hatékony működés érdekében. A felhasználók biztonságos azonosításához és a jelszavak védelméhez a bejelentkezés és regisztráció folyamata során Token alapú azonosítást alkalmazunk. A jelszavakat hashelve tároljuk el, biztosítva ezzel a felhasználói fiókok biztonságát.

<sup>9</sup>Postgres: <https://www.postgresql.org/docs/>



6.2. ábra. User Service – Séma

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

## 6.4. React

A React<sup>10</sup> egy nyílt forráskódú JavaScript könyvtár, amelyet a felhasználói felületek készítésére terveztek. Fő előnye a komponens alapú architektúra, amely lehetővé teszi az alkalmazás felépítését úgy, hogy az különálló, újrahasznosítható komponensekre van bontva. Ezáltal a fejlesztők könnyen tudnak komplex felhasználói felületeket készíteni, miközben könnyen karbantartható és kiterjeszthető marad az alkalmazás. Ezen túl, egy hatékony virtuális DOM alapú megközelítést alkalmaz, amely lehetővé teszi az alkalmazások gyorsabb frissítését és jobb teljesítményét a változások hatékony kezelése révén.

<sup>10</sup>React: <https://legacy.reactjs.org/docs/getting-started.html>

Az alkalmazás felületét JavaScript<sup>11</sup> programozási nyelv segítségével fejlesztettük, mivel ez egy általánosan elterjedt és széles körben támogatott nyelv a webfejlesztés területén. A JavaScript alkalmas a kliensoldali fejlesztésre, és kiválóan együttműködik más webes technológiákkal, például a HTML és CSS, lehetővé téve az egyszerű és hatékony fejlesztést.

A fejlesztés folyamatában a Vite<sup>12</sup> fejlesztői eszköz kiemelkedő szerepet töltött be, melyet a hatékony és gyors fejlesztési folyamat elősegítésére terveztek. Ugyanakkor rendelkezik egy beépített szerverrel, amely a forrásfájlokat natív ES modulokon keresztül szolgáltatja ki, és rendkívül gyors Hot Module Replacement (HMR) funkciót biztosít a modulok dinamikus frissítéséhez futásidőben. Ez azt jelenti, hogy amikor a fejlesztők módosítják a kódot, csak az érintett modulokat frissítik, és azonnal láthatják a változtatásokat az alkalmazásban, anélkül, hogy újra kellene tölteni az egész alkalmazást. Ez a gyors visszajelzés jelentősen felgyorsítja a fejlesztési folyamatot és lehetővé teszi a hatékonyabb munkavégzést.

Ezen felül számos további kiegészítő könyvtárat használtunk fel a felhasználói felület tervezéséhez, amelyek integrációja és működése hatékonyan megvalósítható volt a React rugalmasságának köszönhetően.

## 6.5. 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 esetében a

---

<sup>11</sup>JavaScript programozási nyelv: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

<sup>12</sup>Vite fejlesztői eszköz: <https://vitejs.dev/guide/>

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 Roboflow<sup>13</sup> 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 neuronhálót<sup>14</sup> (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 [10].

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.

---

<sup>13</sup>Roboflow platform: <https://docs.roboflow.com/>

<sup>14</sup>Konvolúciós neuronháló: <https://www.ibm.com/topics/convolutional-neural-networks>

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.

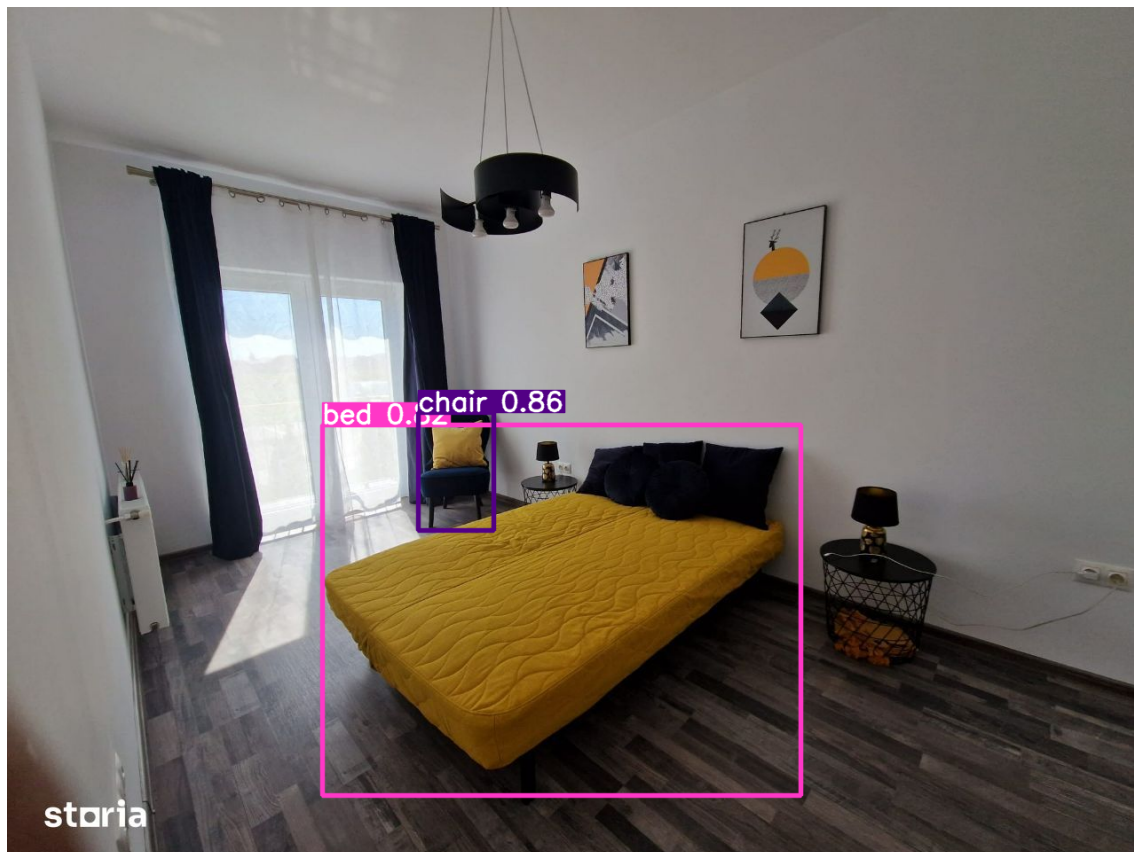
Ezen lépések kombinációjával a YOLOv5 képes hatékonyan és pontosan detektálni a különböző objektumokat a képeken, és visszaadni azok pozícióját és osztályát. A konvolúciós rétegeknek és az objektumfelismerő rétegeknek köszönhetően az algoritmus képes megbízhatóan kezelni különböző képfelismerési feladatokat ([11],[12],[13]).

Ahogy az alábbi 6.3 ábrán is látható, a beépített<sup>15</sup>, 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.

---

<sup>15</sup>YOLOv5 beépített modell, "Coco adathalmaz" - <https://www.kaggle.com/code/venkatkumar001/object-recognition-yolov5-coco-dataset>





6.3. 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 adathalmaz<sup>16</sup> segítségével tanítottunk egy modellt, amely ezen tanításnak megfelelően ismeri fel és szolgáltatja vissza az információkat.

Ezen felül, szükség volt egy olyan modell<sup>17</sup> tanítására is, mely a különböző kellékek és tárgyakat képes felismerni, melyek nem mindegyike található meg a már fent említett, beépített modell osztályai között.

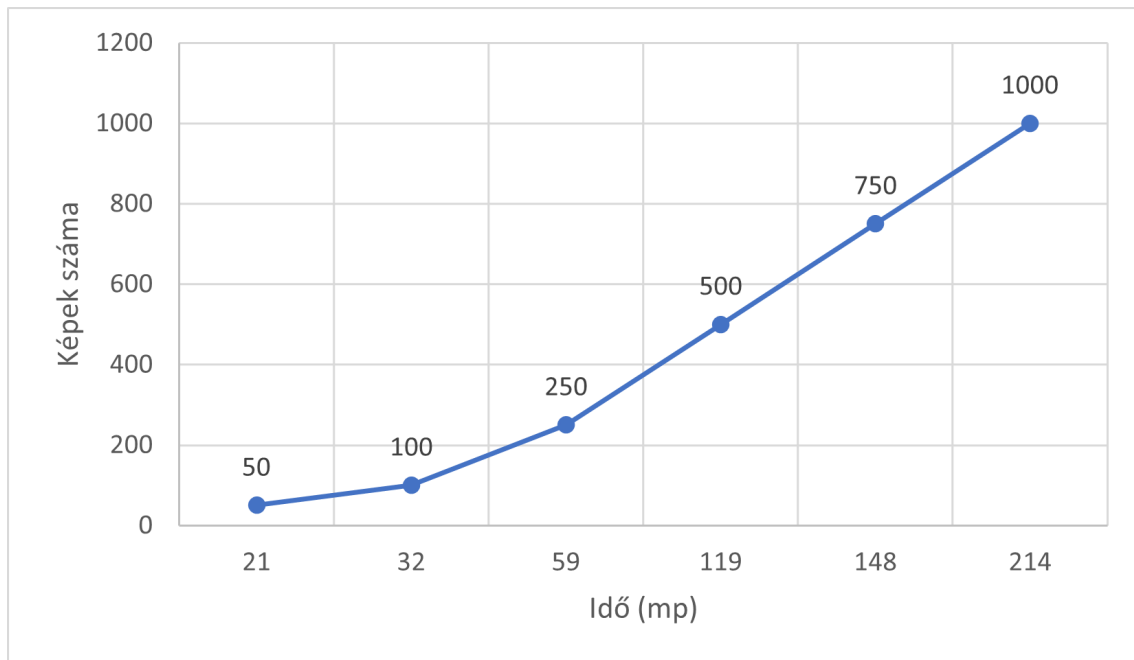
<sup>16</sup>Tanító adathalmaz ajtókra: <https://universe.roboflow.com/mohammed-naji/doors-6g8eb/dataset/1>

<sup>17</sup>Tanító adathalmaz különböző tárgyakra: <https://universe.roboflow.com/um-nlhmr/living-room-items/dataset/1>



**6.4. 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.



**6.5. 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 KerasLayer<sup>18</sup> 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 MobileNetV2<sup>19</sup> 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

<sup>18</sup>KerasLayer mint objektum: [https://www.tensorflow.org/hub/api\\_docs/python/hub/KerasLayer](https://www.tensorflow.org/hub/api_docs/python/hub/KerasLayer)

<sup>19</sup>MobileNetV2: <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.

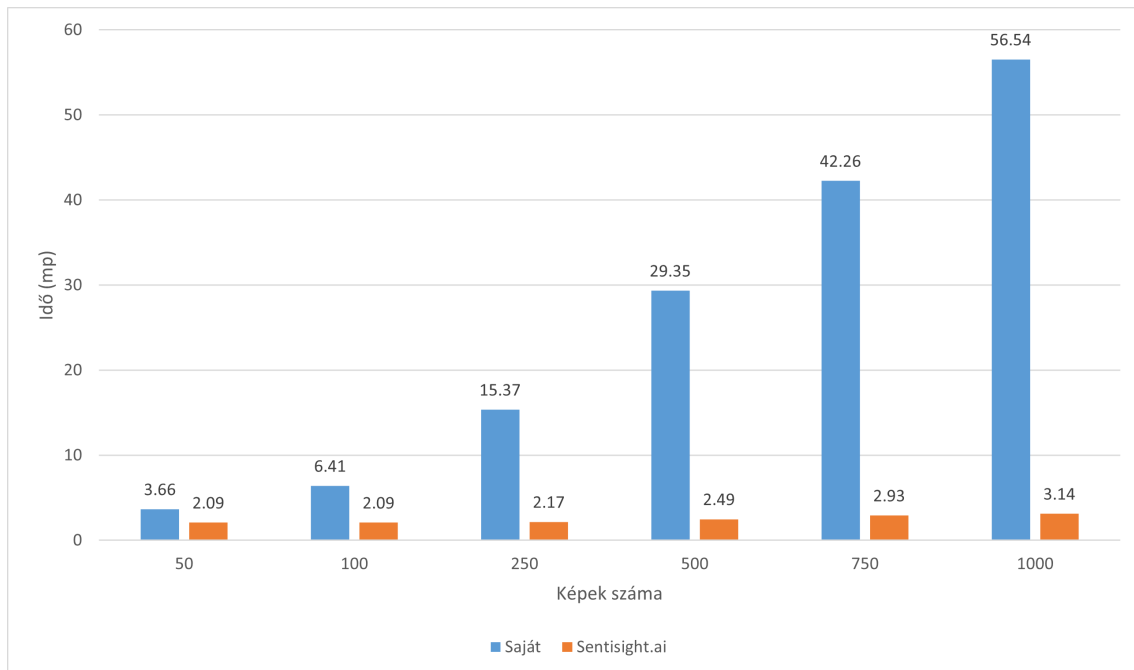
A következőkben az objektumok összehasonlításában mért számításokat szeretnénk szemléltetni, melyek megmutatják a rendszer ezen pontjának a működési elveit. Ahogyan már említettük, igyekeztünk olyan, már rendelkezésre álló alternatívák bevezetésére, mint például a Sentsight.ai<sup>20</sup> platform. 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 falóra több szögből való megvizsgálása volt, itt az 5 megegyező kép egyikét ellenőriztük rendre a többivel. A második egy karóra volt, ez

---

<sup>20</sup>Sentsight.ai platform: <https://www.sentsight.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.



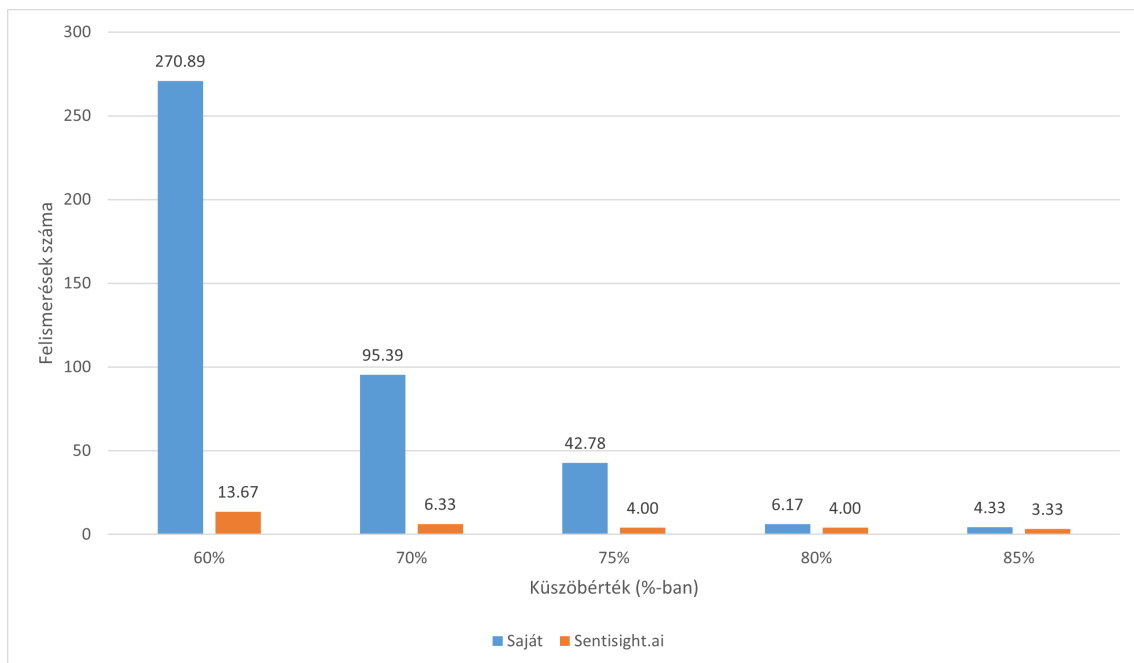
**6.6. ábra.** Átlag időtartam: Saját vs Sentisight.ai

A fenti 6.6 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 6.7 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.



**6.7. ábra.** Átlag felismerés: Saját vs Sentsight.ai

A továbbiakban egy a küszöbértékekre vonatkozó mérést (melléklet 1. oldal táblázat) szemléltettük, amely a túlságosan magas értékek esetén történhet. Tehát a mérésekből való következtetés alapján, az egység a kisebb méretű adathalmazokat, illetve a 80-87 százalékos küszöbértékkel ellátott tartományban lévő adatokat képes a leghatékonyabban és legpontosabban meghatározni, illetve felismerni. Az algoritmus futási ideje (melléklet 2. oldal táblázat) jelentősen nő, ahogy a képek száma növekszik. Ez a növekedés

egyértelműen az objektumdetektálás komplexitásának és az adatfeldolgozás terhelésének növekedéséből ered.

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.

Végül az összehasonlítás során visszaszolgáltató 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.

## 6.6. Docker

A Docker<sup>21</sup> 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.

---

<sup>21</sup>Docker: <https://www.docker.com/>, [https://ithub.hu/blog/post/Docker\\_bevezetes\\_a\\_containerek\\_vilagaba](https://ithub.hu/blog/post/Docker_bevezetes_a_containerek_vilagaba)



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

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.

## 6.7. Kubernetes

A Kubernetes<sup>22</sup> 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 konténeres folyamatok 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.

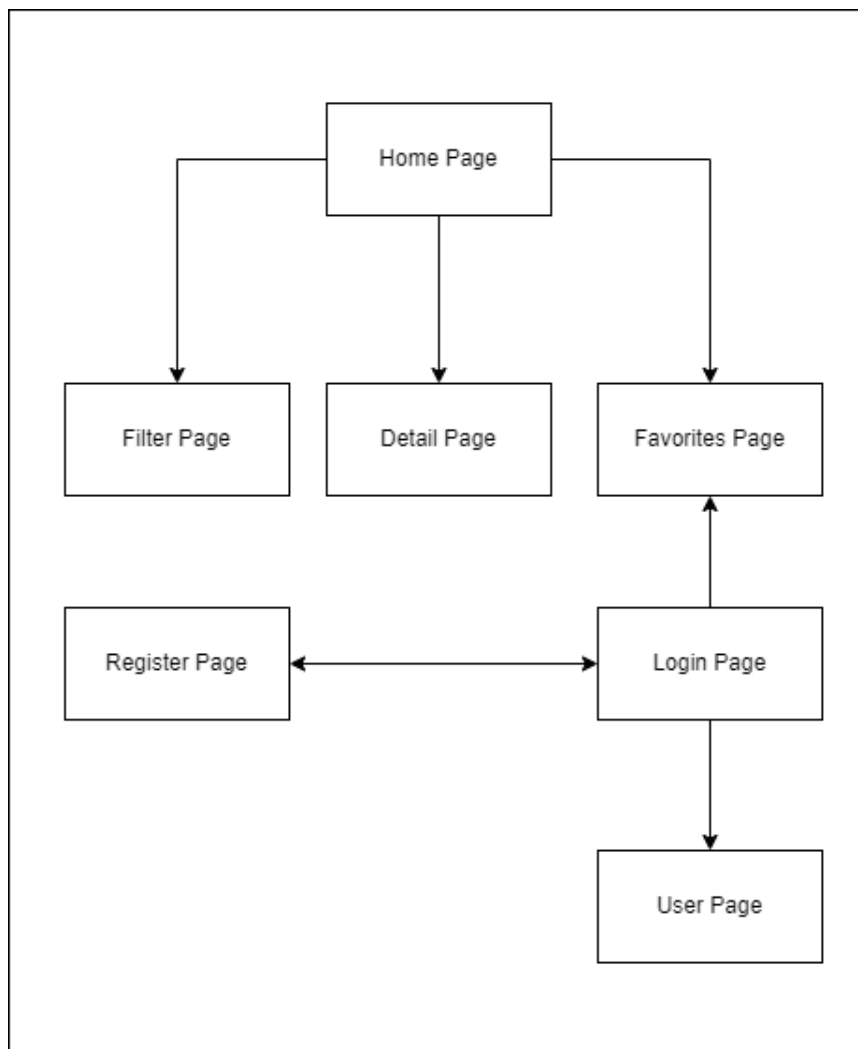
---

<sup>22</sup>Kubernetes: <https://kubernetes.io/>, <https://fotc.com/hu/blog/kubernetes-tutorial/>



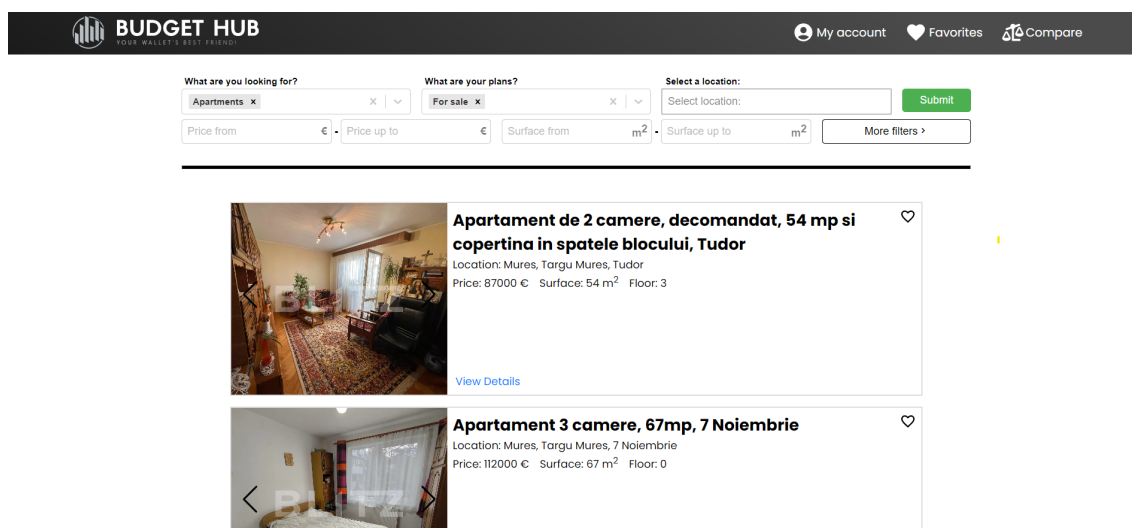
## 7. fejezet

### A BudgetHub alkalmazás



7.1. ábra. Wireframe diagram

Ahogy a fenti Wireframe 7.1 diagrammon láthatjuk, az oldal látogatása során a kezdőoldal, a „Home Page” fogad, ahonnan minden más oldalt elérhetünk a maga módján és idejében. Először is a főoldalon találunk egy navigációs felületet, ahol az alkalmazás logója is helyet kapott, erre bárholonnan kattintva a főoldalra fog navigálni bennünket. Emellett található három navigációs menü, mely a későbbiekben bővíteni fog egy „Social” opcióval, ezekre egyenként térünk ki a következőkben.




7.2. ábra. Kezdőoldal

Található egy rendezésre, valamint szűrésre szolgáló felület, ahol a felhasználó/látogató egyaránt tud majd kedvére keresni és különböző szűréseket végezni. A „Submit” gombra kattintva, az alkalmazás átnavigál arra az oldalra, ahol a beállított szűréseket alkalmazva visszakapjuk a megfelelő eredményt. A továbbiakban hasonlóan reagál, mint a főoldalunk.

Végül az ingatlanokról szóló hirdetések listája, melyek most az adatbázisban levő összes adat megjelenítésére szolgál, viszont itt olyan különleges, illetve legkedvezőbb típusú ingatlanok fognak helyet kapni, melyek a felhasználók számára kecsegtető és legkedvezőbb eseteket nyújthatják. Ezek a lista elemek/hirdetések egyenként kiválaszthatóak és megtekinthetőek, így egy kattintásra a „Detail Page”-n találjuk magunkat, ahol az ingatlanról egy részletesebb leírást kapunk, további információk, szolgáltatások vannak megjelenítve, ami magába foglalja az ingatlan tartózkodási helyét egy Google Térkép keretein belül, ami a környékről is szemléletes képet mutat az érdeklődőnek.



**Apartment de 2 camere, decomandat, 54 mp si copertina in spatele blocului, Tudor**   
 Location: Mures, Targu Mures, Tudor Price: 87000 € 161111 €/m<sup>2</sup>

[SEARCH FOR BETTER PRICE](#)

### Description

Alina te asteapta spre achizitionare un apartament de 2 camere, zona Tudor. Este situat intr-o zona foarte buna, in special pentru o familie deoarece, in imediata apropiere regasim multiple linii de transport in comun, scoli, magazine, supermarketuri, parcuri de joaca pentru copii, etc.

### General informations

Surface:	54
Number of rooms:	2
Status:	gata de utilizare
Floor:	3
Balcony/garden/terrace:	balcon
Year of construction:	1990



7.3. ábra. Oldal az ingatlan részletesebb leírásairól

Erről a pontról, ha nincs bejelentkezve az illető, és kedvelni szeretne egy ingatlant, egy üzenetet fog kapni, mely javasolja a bejelentkezést ehhez a művelethez, ugyanakkor, ha a navigációs felületen a három lehetőség közül bármelyikre kattint, automatikusan a „Login Page”-n találja magát, hisz ezek a funkciók csak a bejelentkezett felhasználók számára érhetőek el. Innen már egészen egyszerű a döntés, ha nincs még felhasználói

fiókja az illetőnek, akkor lehetősége van átnavigálni magát a „Register Page”-re, ahol regisztrálhat az oldalunkra, továbbá, ha elfelejtette volna a jelszavát, akkor kicserélheti azt a „Forgot Password Page”-n, végül ha már van fiókja, csak nincs bejelentkezve, ennek elvégzését követően, már teljeskörűen minden funkciót elérhet és használatba is vehet.

The image shows two side-by-side screenshots of a user interface. The left screenshot is titled "Register" and contains the following fields: "Name" (Full name), "Email address" (Email), "Password", "Confirm Password", and "Birth date" (formatted as "éééé. hh. nn."). There are also checkboxes for "Remember me" and "Do you already have an account?" with a "Sign in" link. A "Sign up" button is at the bottom. The right screenshot is titled "Sign in" and contains the following fields: "Email address" (Email) and "Password". There are checkboxes for "Remember me" and a "Forgot password?" link. A "Sign in" button is in the center, and a "Don't have an account? Sign up" link is at the bottom.

7.4. ábra. Regisztráció és bejelentkezés oldal

Visszatérve a navigációs felületen található három opcióra, a legelső ezek közül a „My account” feliratot kapta, erre kattintva a „User Page” oldalra navigál az alkalmazás, ahol a bejelentkezett felhasználó saját adatait tudja megtekinteni, kijelentkezni, és a továbbiakban fejlesztett érdeklődési körét is be tudja majd állítani, mely során olyan értesítéseket fog kapni, amelyek megfelelő ingatlanokat fognak tartalmazni a beállított opciók figyelembevételével. A második a „Favorites”, mely a bejelentkezett felhasználó kedvelt ingatlanait fogja tartalmazni, itt lehetőség lesz ezek eltávolítására, illetve rendezésére.

A harmadik a „Comparison”, ahol a kiválasztott ingatlanok összehasonlítására kerül a sor, itt megtekinthetjük táblázat-összehasonlító formában, hogy miben egyeznek meg illetve miben különböznek az előzőleg kiválasztott ingatlanok.

A kitelepített alkalmazásunk elérhető ezen a linken: <http://188.213.217.231:8020>

## 8. fejezet

# Továbbfejlesztési lehetőségek

A fentiekben említett „Social Page” kigondolása, megtervezése, kivitelezése. Ez a része a projektnek egy teljesen tovább gondolt és kiterjesztett problémát oldhat meg a felhasználók számára. Ez a funkció mind az ingatlan eladó és vásárló igényeit igyekszik kielégíteni, mely során a vásárló teszi meg az első lépéseket, ami azt jelenti, hogy egy bejegyzést tesz közzé a felületen, mely magába foglalja, hogy mi iránt érdeklődik, vagyis a helységet/régiót, ingatlan típusát, árát, méretet stb. Továbbá az elérhetőségeit, mely alapján az eladó fél felveheti ezzel a kapcsolatot, amennyiben megegyeznek az említett feltételek. Itt természetesen az eladónak lehetősége van olyan bejegyzések opcióira szűrni, amelyek megegyeznek az eladásra kívánt ingatlanával.

Egy jelentős továbbfejlesztési lehetőség az adatgyűjtés kiterjesztése további oldalakra, ezáltal jelentősen növekedne az oldalunkon található ingatlanok száma.

További szűréssel kapcsolatos lehetőségek bevezetése. Eredményként kapott ingatlanok rendezése. UX javítása, felhasználó aktivitásának nyomon követése.

„Account Page”-n további funkciók bevezetése, adatok módosítása, előző keresések megjelenítése hirdetés formában, olyan jellegű ingatlanok ajánlása, melyek felkelthetik a felhasználó figyelmét. Illetve az említett érdeklődési felület kialakítása, implementálása.

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.

Egy fontos fejlesztési irány az AI rész kiegészítése az üres lakások azonosításában és összehasonlításában, mivel a mostani rendszerünk 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 modellek pontosságát.

## 9. fejezet

# Összefoglaló

A dolgozatunkban bemutattuk az ingatlanpiac rejtjelmeit 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.

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 problémákba, melyek megnehezítették, vagy esetleg eltántorították őket a vásárlástól. Ezáltal a felhasználóink igényeit és lehetőségeit minél jobban ki tudjuk elégíteni.

## 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] 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](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).
- [3] 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.
- [4] 10 szempont, ami biztosít arról, hogy helyesen választasz. <https://www.groupama>.

[ro/ghiduri/cumparare-apartament](https://ro/ghiduri/cumparare-apartament).

- [5] Ingatlanvásárlás menete, otthonteremtés lépései. <https://tudastar.ingatlan.com/tippek/az-ingatlanvasarlas-menete/>.
- [6] Gary W Eldred. *The 106 Common Mistakes Homebuyers Make (and how to avoid them)*. John Wiley & Sons, 2002.
- [7] Általános hibák, melyeket a vásárlók elkövettek. <https://blog.remax.ro/5-greseli-comune-pe-care-le-fac-cumparatorii-de-case>.
- [8] Diana Olick. This is the best day — and time — to list your home for sale. *cncb*, 2018. <https://www.cncb.com/2018/05/10/this-is-the-best-day--and-time--to-list-your-home-for-sale.html>.
- [9] 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>.
- [10] 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.
- [11] Yolov5 architektúra - működés. [https://docs.ultralytics.com/yolov5/tutorials/architecture\\_description/](https://docs.ultralytics.com/yolov5/tutorials/architecture_description/).
- [12] Ravpreet Kaur and Sarbjeet Singh. A comprehensive review of object detection with deep learning. *Digital Signal Processing*, 132:103812, 2023.
- [13] Yolov5 működésének áttekintése. <https://sh-tsang.medium.com/brief-review-yolov5-for-object-detection-84cc6c6a0e3a/>.