DOKUMENTÁCIÓ

Szoftverfejlesztő- és tesztelő képzésen



Rentify

/Ingatlanbérlési platform/

Készítette: Jancsurák Napsugár,

Varga Antónia,

Katona Alexandra

Konzulens: Németh Bence

Tartalom

[Fejlesztői környezet 4](#_Toc195079356)

[Frontend fejlesztői környezete: 4](#_Toc195079357)

[**a.** A HTML5 4](#_Toc195079358)

[**b.** CSS3: 4](#_Toc195079359)

[**c.** HTML5 és CSS3 együttműködése: 4](#_Toc195079360)

[**d.** Bootstrap: 5](#_Toc195079361)

[**e.** WPF (Windows Presentation Foundation): 5](#_Toc195079362)

[Backend fejlesztői környezete: 6](#_Toc195079363)

[**a.** C#: 6](#_Toc195079364)

[**b.** .NET (vagy .NET Core/5/6/7): 6](#_Toc195079365)

[**c.** ASP.NET Core: 6](#_Toc195079366)

[**d.** SQL Server / MySQL / PostgreSQL / SQLite: 7](#_Toc195079367)

[**e.** Entity Framework (EF): 7](#_Toc195079368)

[**f.** REST API: 8](#_Toc195079369)

[Felhasználói dokumentáció 9](#_Toc195079370)

[Motiváció és tervezés 9](#_Toc195079371)

[Adatbázis és kapcsolatok 10](#_Toc195079372)

[Első elképzelések 10](#_Toc195079373)

[React alapú weboldal 11](#_Toc195079374)

[Bejelentkezés és regisztráció 12](#_Toc195079375)

[Profiloldal 13](#_Toc195079376)

[Pontosabb szűrés elérése 14](#_Toc195079377)

[WPF karbantartó app 15](#_Toc195079378)

[React Native mobilapplikáció 16](#_Toc195079379)

[Tesztelés és minőségbiztosítás 17](#_Toc195079380)

[Szerverre való kihelyezés 17](#_Toc195079381)

[Irodalomjegyzék 20](#_Toc195079382)

# Fejlesztői környezet

## Frontend fejlesztői környezete:

1. A HTML5 (Hypertext Markup Language 5) az internetes tartalom egyik alapja, amely a weboldalak struktúráját és tartalmát határozza meg. Az előző verziókhoz képest a HTML5 számos új elemmel és funkcióval bővült, amelyek jelentősen javítják a weboldalak működését, hozzáférhetőségét és interaktivitását.

Az egyik legfontosabb újítás, hogy a HTML5 már támogatja a multimédiás elemek (például audio és video) közvetlen lejátszását HTML kódból, anélkül, hogy külső pluginokra lenne szükség, így egyszerűbbé válik a tartalmak integrálása. Emellett a HTML5 lehetőséget biztosít a fejlettebb grafikai megjelenítésre is a és az SVG (Scalable Vector Graphics) technológiák bevezetésével. Ezek segítségével dinamikus grafikák és animációk hozhatók létre közvetlenül a böngészőben, amely korábban csak harmadik féltől származó bővítményekkel volt lehetséges.

A HTML5 által bevezetett szemantikus HTML elemek, mint például a <header>, <footer>, <article>, <section>, <nav> nemcsak a weboldal szerkezetének tisztábbá tételét segítik, hanem hozzájárulnak a SEO (keresőoptimalizálás) javításához is. Ezek az elemek segítik a keresőmotorokat abban, hogy jobban megértsék és indexeljék az oldal tartalmát, így elősegítik a weboldalak jobb helyezését a keresési eredmények között.

Összességében a HTML5 egy jelentős előrelépés a webfejlesztés területén, amely számos új lehetőséget biztosít a modern, interaktív és reszponzív weboldalak fejlesztéséhez. A szemantikus elemek, a multimédiás támogatás, az új API-k és az offline működési lehetőségek mind hozzájárulnak ahhoz, hogy a webfejlesztők hatékonyabb és felhasználó barátibb weboldalakat készíthessenek.

1. CSS3: A **CSS3** (Cascading Style Sheets, Level 3) a HTML5 mellett a weboldalak megjelenésének és dizájnjának alapvető szabályait határozza meg. A CSS3 az előző verziókhoz képest számos új funkcióval bővült, amelyek lehetővé teszik a fejlesztők számára, hogy dinamikus, interaktív és reszponzív felületeket készítsenek. A **reszponzív dizájn** azt jelenti, hogy az alkalmazás automatikusan alkalmazkodik a különböző képernyőméretekhez és eszközökhöz, így a felhasználók élménye minden eszközön (asztali gépen, mobilon, tableten) optimalizált marad. A **flexbox** és **CSS grid layout** technológiák a CSS3-ban lehetővé teszik az összetett elrendezések könnyed kezelését, miközben javítják a weboldalak teljesítményét is. A CSS3 segítségével a fejlesztők finomhangolhatják a **vizuális elemeket**, mint például a gombokat, navigációs sávokat, képeket, háttérszíneket és animációkat. Az új CSS funkciók, mint a **@keyframes** és **transitions**, lehetővé teszik a weboldal animációinak, áttűnéseinek és egyéb dinamikus hatásainak implementálását, amelyek az oldalon belépő és eltűnő elemek mozgásának és vizuális változásainak kezelését is biztosítják.
2. HTML5 és CSS3 együttműködése: A **HTML5** és **CSS3** együtt biztosítják a weboldalak alapvető felépítését és megjelenését. A HTML5 biztosítja az oldal tartalmát és szerkezetét, míg a CSS3 a megjelenést, a vizuális stílusokat és az elrendezéseket szabályozza. A CSS3 lehetőséget biztosít az oldalak **reszponzív** megjelenítésére, így az alkalmazások automatikusan alkalmazkodnak a különböző kijelzők és eszközök méreteihez. A HTML5 új elemzései és struktúrái együttműködnek a CSS3 által kínált vizuális funkciókkal, lehetővé téve komplex felhasználói felületek és interaktív weboldalak gyors létrehozását. Emellett a CSS3-es animációk és a **flexbox** vagy **grid** alapú elrendezések segítenek a fejlesztőknek a dinamikus és vizuálisan gazdag felületek megvalósításában.
3. Bootstrap: A Bootstrap egy nyílt forráskódú front-end keretrendszer, amely célja, hogy megkönnyítse a webalkalmazások és weboldalak fejlesztését, különösen a gyors és egyszerű felépítésük terén. A Bootstrap legnagyobb előnye, hogy előre definiált dizájn elemeket és egy rugalmas grid rendszert biztosít, amely jelentősen felgyorsítja a fejlesztési folyamatot. Ennek köszönhetően a fejlesztők könnyedén létrehozhatnak reszponzív, szép és jól karbantartható weboldalakat, anélkül, hogy az alapvető dizájn és elrendezés részletezésére kellene koncentrálniuk.

A **grid rendszer** az egyik alapvető komponens, amely segít az oldal elrendezésében. Ez lehetővé teszi a fejlesztők számára, hogy egyszerűen hozzanak létre rugalmas, több oszlopos elrendezéseket, miközben biztosítja, hogy az oldalak jól nézzenek ki különböző képernyőméreteken. A grid rendszernek köszönhetően nem szükséges bonyolult CSS kódot írniuk, mivel az előre megadott oszlopok és sávok segítenek az elrendezés gyors kialakításában.

A **Bootstrap** emellett számos előre megtervezett és testreszabható **UI komponenset** tartalmaz. Ezek közé tartoznak például a gombok, navigációs menük, űrlapok, kártyák, értesítési rendszerek és sok más hasznos elem, amelyek már beépített stílusokkal és funkciókkal rendelkeznek. Ez jelentős időmegtakarítást eredményez, mivel nem kell minden egyes komponens stílusait és interakcióit külön-külön megírni.

1. WPF (Windows Presentation Foundation): WPF egy Microsoft által kifejlesztett technológia, amely Windows alapú asztali alkalmazások fejlesztésére szolgál. A WPF az asztali alkalmazások felhasználói felületét kezeli, és számos fejlett grafikai elemet, animációt és interaktivitást kínál. A WPF lehetővé teszi a fejlesztők számára, hogy dekoratív, dinamikus és gazdag felhasználói élményt biztosítsanak a felhasználók számára, miközben teljes mértékben kihasználja a modern grafikus kártyák és a Windows operációs rendszer képességeit.

A WPF XAML (Extensible Application Markup Language) szintaxist használ a felhasználói felületek deklaratív leírására, amely segíti a logika és a megjelenítés elválasztását, ezáltal tiszta és karbantartható kódot biztosít. A WPF támogatja az MVVM (Model-View-ViewModel) architektúrát, amely a kód tisztán tartására és a tesztelhetőség növelésére szolgál. A WPF erőteljes adatbinding mechanizmusa segíti a felhasználói felületek dinamikus frissítését, amikor az alkalmazás adatmodellje változik.

A WPF fejlett grafikai lehetőségeket biztosít, beleértve a vektorgrafikát, 3D megjelenítést és hardveresen gyorsított renderelést, amely lehetővé teszi a nagy teljesítményű vizuális elemek és animációk használatát. A WPF támogatja az eseményvezérelt programozást és a stílusok, sablonok alkalmazását, amelyek segítségével testreszabható és újrafelhasználható UI-komponenseket lehet létrehozni.

A WPF további előnyei közé tartozik az adatvizualizáció támogatása különböző vezérlők, például diagramok és táblázatok segítségével, valamint a multi-touch és a stylus eszközök kezelése. Az elosztott rendszerekkel való kompatibilitás érdekében a WPF képes kommunikálni WCF (Windows Communication Foundation) szolgáltatásokkal, lehetővé téve az adatok hatékony feldolgozását és megjelenítését komplex üzleti alkalmazások esetén.

## Backend fejlesztői környezete:

1. C#: A **C#** egy objektum-orientált, **erőteljes programozási nyelv**, amelyet a **Microsoft** fejlesztett ki, és a **.NET** platformon való fejlesztéshez használják. A C# modern programozási nyelvként számos fejlett funkcióval rendelkezik, amelyek segítenek a fejlesztőknek tiszta, karbantartható és hibamentes kódot írni. A C# erős típusellenőrzést, **garbage collection**-t (automatikus memória kezelés), valamint **LINQ** (Language Integrated Query) támogatást biztosít, amely megkönnyíti az adatkezelést és a komplex lekérdezések írását. A C# szoros integrációban áll a **.NET** platformmal, így a backend logika, adatkezelés és API fejlesztés ideális eszköze. A C# nyújtja az eszközöket, amelyek lehetővé teszik a fejlesztők számára a komplex üzleti logikák megvalósítását, miközben biztosítják a kód könnyű karbantarthatóságát.
2. .NET (vagy .NET Core/5/6/7): A .NET egy fejlett keretrendszer, amely lehetővé teszi a fejlesztők számára, hogy különböző típusú alkalmazásokat hozzanak létre, mint például webalkalmazások, asztali alkalmazások, mobil alkalmazások, és szolgáltatások. A .NET-et a Microsoft fejlesztette ki, és az évek során jelentős fejlődésen ment keresztül, hogy megfeleljen a modern fejlesztési követelményeknek.

A .NET Core egy nyílt forráskódú és cross-platform verziója a hagyományos .NET Frameworknek, amely támogatja a Windows, Linux és macOS rendszereket, így rugalmasan alkalmazkodik a fejlesztési igényekhez. A .NET Core 3.1 volt az utolsó verziója a .NET Core sorozatnak, ezt követően a Microsoft egyesítette a különböző .NET implementációkat, és bevezette a .NET 5-öt, amely az egységes platformként szolgál a további fejlesztések számára.

A .NET platform gyors és skálázható alkalmazások fejlesztésére lett optimalizálva, amely különösen előnyös nagy terhelésű és adatintenzív alkalmazások esetén. Az újabb verziók, mint a .NET 5, 6 és 7, számos új funkciót és fejlesztési lehetőséget kínálnak, így a fejlesztők még gyorsabban és hatékonyabban tudják elkészíteni a kívánt alkalmazásokat.

1. ASP.NET Core: Az ASP.NET Core egy rendkívül erőteljes és rugalmas webfejlesztési keretrendszer, amely a .NET Core környezetre épül, és számos előnnyel rendelkezik. A legfontosabb jellemzője, hogy cross-platform, így az alkalmazások futtathatók Windows, Linux és macOS rendszereken is, biztosítva a fejlesztők számára a különböző platformokon való munkavégzést.

Az ASP.NET Core az MVC (Model-View-Controller) architektúrát követi, amely segít a fejlesztőknek abban, hogy jól szervezett, könnyen érthető és karbantartható kódot írjanak. Az MVC elválasztja az adatokat (Model), a felhasználói felületet (View), és az üzleti logikát (Controller), ezáltal javítva a kód tisztaságát és skálázhatóságát.

Az Entity Framework Core (EF Core) egy ORM (Object-Relational Mapping) eszköz, amely lehetővé teszi az adatbázisok kezelését közvetlenül a C# nyelven keresztül. Ezzel az eszközzel a fejlesztők könnyebben dolgozhatnak az adatbázisokkal, mivel nem szükséges SQL kódot írniuk, és az adatkezelés is egyszerűbbé válik.

Az ASP.NET Core kiemelkedik a teljesítményével és skálázhatóságával is. A keretrendszer optimalizált, így ideális választás a nagy teljesítményű, skálázható alkalmazások számára. Ez különösen fontos lehet olyan projektek esetén, ahol nagy adatforgalom vagy gyors válaszidő szükséges.

Továbbá, az ASP.NET Core teljes mértékben támogatja a modern webtechnológiákat, például a RESTful API-kat, WebSocketeket és SignalR-t, valamint a legújabb JavaScript és CSS könyvtárakat. Ez biztosítja a fejlesztők számára a legújabb eszközöket és funkcionalitásokat, így mindig naprakészen dolgozhatnak.

1. SQL Server / MySQL / PostgreSQL / SQLite: Az **adatbázisok** elengedhetetlenek a backend fejlesztés során, mivel az alkalmazásoknak strukturált módon kell tárolniuk és lekérdezniük az adatokat. Az adatbázis-kezelők közül a **SQL Server**, **MySQL**, **PostgreSQL** és **SQLite** mindegyike különböző funkciókat és előnyöket kínál. A **SQL Server** egy robusztus, vállalati szintű adatbázis-kezelő rendszer, amelyet a Microsoft fejlesztett ki, és kifejezetten nagy terhelésű alkalmazásokhoz alkalmas. A **MySQL** egy nyílt forráskódú, elterjedt relációs adatbázis-kezelő rendszer, amely az egyszerűbb alkalmazások számára kiváló választás. A **PostgreSQL** egy másik nyílt forráskódú adatbázis-kezelő, amely támogatja a bonyolultabb adatstruktúrákat és a transzakciók kezelését. Az **SQLite** egy könnyű, beágyazott adatbázis, amely ideális kisebb alkalmazásokhoz vagy helyi adatkezeléshez, mivel nincs szükség egy különálló szerverre a működéséhez. A választott adatbázis-kezelő rendszer az alkalmazás követelményeitől és méretétől függően változhat, de mindegyik biztosítja az adatok hatékony tárolását és gyors lekérdezését.
2. Entity Framework (EF): Az Entity Framework (EF) egy objektum-relációs leképezési (ORM) keretrendszer a .NET környezetben, amely segíti a fejlesztőket az adatbázisok és a programozási nyelvek közötti kapcsolat kezelésében. Az EF lehetővé teszi, hogy a fejlesztők C# osztályokat és objektumokat használjanak az adatbázis-táblák reprezentálására, így elkerülhetik a közvetlen SQL lekérdezések írását. Ezáltal a kód tisztábbá válik, és az adatbázis-kezelés is egyszerűbbé válik.

Az EF két fő megközelítést kínál az adatbázis-kezeléshez. A **Code First** megközelítés lehetővé teszi, hogy a fejlesztők az adatbázist a kód alapján generálják. Ezzel a módszerrel a modellek (C# osztályok) először elkészülnek, és az EF automatikusan létrehozza az adatbázist. Ezzel szemben a **Database First** megközelítés során a már létező adatbázisból hozhatjuk létre a szükséges kódot, azaz az adatbázis struktúrája először készül el, és az EF ezt használja a kód generálásához.

Az Entity Framework megkönnyíti az adatbázis műveletek kezelését, mint az adatok lekérdezése, frissítése és törlése. A fejlesztők komplex lekérdezéseket és tranzakciókat is végezhetnek anélkül, hogy kézi SQL kódra lenne szükségük. Az EF tehát jelentős idő- és munkaerő-megtakarítást jelent a fejlesztési folyamat során.

Az **EF Core** az Entity Framework könnyebb, gyorsabb és platformfüggetlen verziója, amely kifejezetten a .NET Core környezethez készült. Az EF Core számos új funkcióval és fejlesztéssel érkezett, így gyorsabb működést biztosít, miközben lehetővé teszi az alkalmazások futtatását különböző operációs rendszereken, például Windows, Linux és macOS platformokon.

1. REST API: A REST (Representational State Transfer) egy architektúra-stílus, amely az interneten keresztül történő adatkommunikációra összpontosít. A RESTful API egy olyan alkalmazásprogramozási felület (API), amely a HTTP protokollt használja az adatok átvitelére a szerver és a kliens között. A REST alapú API-k széles körben elterjedtek, mivel lehetővé teszik a különböző platformok és rendszerek közötti egyszerű és hatékony kommunikációt.

A REST API-k **stateless** működnek, ami azt jelenti, hogy minden egyes kérés teljes körű információt tartalmaz, így a szerver nem tárol semmilyen adatot az előző kérésekről. Mivel nincs szükség állapotkezelésre a szerver oldalán, a REST API-k képesek hatékonyan kezelni a nagy forgalmat, ami különösen fontos nagy skálázhatóságú alkalmazások esetén. Az egyes kérések önállóan is feldolgozhatók, függetlenül a korábbi kérésektől, így javítva a rendszer válaszidejét és teljesítményét.

A RESTful API-k lehetővé teszik, hogy különböző alkalmazások és rendszerek kommunikáljanak egymással, és gyors adatátvitelt biztosítanak. A REST API-k az **HTTP** metódusokat használják, például a GET, POST, PUT, DELETE, PATCH műveleteket, amelyek az adatok lekérdezésére, létrehozására, módosítására és törlésére szolgálnak. Ez az egyszerű és intuitív megközelítés biztosítja, hogy a fejlesztők gyorsan megértsék és implementálják a RESTful API-kat.

A **JSON** (JavaScript Object Notation) formátumot széles körben támogatják a REST API-k, mivel könnyen olvasható és írható, valamint könnyen integrálható a legtöbb programozási nyelvbe. A JSON formátum előnye, hogy kis helyet foglal, gyorsan feldolgozható, és emberi olvasásra is alkalmas, így ideális választás az adatok közötti kommunikációhoz.

A RESTful API-k továbbá könnyen integrálhatók más rendszerekbe, mivel az HTTP protokoll az internetes kommunikáció alapja, így a REST API-k bármely eszközről vagy programozási környezetből elérhetők. A REST API-k által használt stateless és egyszerűségüknek köszönhetően gyors adatcserét tesznek lehetővé, és rugalmasan alkalmazhatóak a különböző típusú webes alkalmazásokban és szolgáltatásokban.

A REST API-k emellett biztonságosak is lehetnek, ha megfelelő autentikációs és autorizációs mechanizmusok, például az OAuth2, JWT (JSON Web Token) vagy API kulcsok alkalmazásával biztosítjuk, hogy a kommunikáció biztonságos maradjon. Összességében a RESTful API-k az internetes alkalmazások és szolgáltatások közötti adatcserét egyszerűsítik, miközben lehetővé teszik a nagy teljesítményű, könnyen skálázható és platformfüggetlen rendszerek létrehozását.

## Felhasználói dokumentáció

### Motiváció és tervezés

A projekt célja egy olyan webes alapú ingatlanbérlési platform létrehozása, amely egyszerű, átlátható felhasználói élményt nyújt mind a bérbeadók, mind a bérlők számára. A platform kiindulópontként szolgálhat egy későbbiekben továbbfejleszthető rendszerhez, amely a magyarországi ingatlanbérleti piac egyik korszerű és könnyen használható alternatívája lehet.

#### a. Célfelhasználók

A platformot elsősorban magánszemélyek és kisebb ingatlanügynökségek számára terveztük, akik lakásokat vagy egyéb ingatlanokat szeretnének bérbe adni. A másik fő felhasználói csoportot azok az emberek alkotják, akik egyszerűen és gyorsan szeretnének kiadó lakásokat keresni. A cél, hogy a regisztrációtól a keresésen és hirdetésfeladáson át a kapcsolatfelvételig minden lépés logikus, gyorsan átlátható és zökkenőmentes legyen.

#### b. Hasonló rendszerek

A piacon számos hasonló rendszer létezik, mint például az Airbnb, az Ingatlan.com vagy a Rentingo. Ezek a rendszerek sokszor komplex funkciókat kínálnak, melyek kezdő felhasználók számára bonyolultnak tűnhetnek. A mi megközelítésünk egy egyszerűsített változat: az első verzió az alapvető funkciókra fókuszál – hirdetés létrehozása, böngészés, keresés, kapcsolatfelvétel – és ezek felhasználóbarát megvalósítására helyezi a hangsúlyt.

#### c. Innováció és alkalmazhatóság

Bár a jelenlegi implementáció az alapfunkciókat valósítja meg, a platform úgy lett kialakítva, hogy a jövőben könnyen bővíthető legyen. Lehetőség van például:

* online fizetési rendszer integrálására,
* értékelési és visszajelzési modul bevezetésére (mind bérlők, mind tulajdonosok részéről),
* térképes keresés implementálására,
* valamint mobilalkalmazás fejlesztésére is.

A modularitás és az átlátható struktúra biztosítja a projekt hosszú távú alkalmazhatóságát és fenntarthatóságát.

#### d. Fejlesztési feladatmegosztás

A projektcsapat három főből állt, és a feladatok elosztása a csapattagok erősségei szerint történt:

* **Jancsurák Napsugár** a frontend fejlesztéséért felelt. Ő alakította ki a webes felületet, gondoskodott a reszponzív dizájnról és az intuitív felhasználói élményről, emellett az egységtesztelést és az integrális tesztelést is magára vállalta.
* **Katona Alexandra** az adatbázistervezést és a WPF-es prototípus elkészítését vállalta magára. Ő dolgozta ki az adattárolás logikáját, és létrehozta az adatkapcsolatokat biztosító struktúrákat.
* **Varga Antónia** a backend fejlesztését végezte, ideértve az API-k elkészítését, a szerveroldali logikát és az adatkezelést, majd mikor ezzel végzett Selenium WebDriver tesztelést végzett.

A projekt során verziókezelő rendszert (Git) használtunk, így mindhárom csapattagnak lehetősége volt nyomon követni a fejlesztés menetét és szükség esetén segíteni egymás munkájában. A rendszeres kommunikáció és a közös tesztelés biztosította, hogy a fejlesztés egységes maradjon és gördülékenyen haladjon előre.

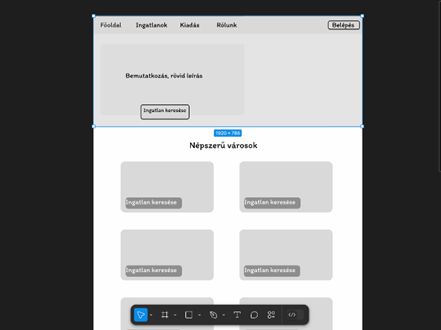
### Adatbázis és kapcsolatok

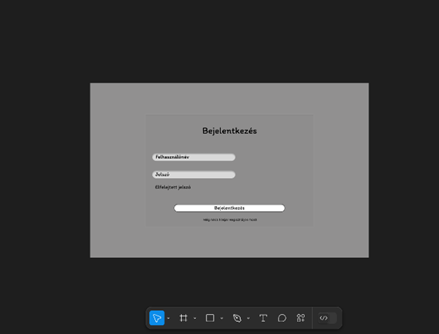
#### Kapcsolatok

Ez az adatbázis egy ingatlan-kölcsönző rendszert ír le. A felhasznalok lehetnek tulajdonosok (az ingatlanok táblához kapcsolódva) vagy bérlők (a foglalasok táblához kapcsolódva). Minden ingatlan egy tulajdonoshoz tartozik, több képe (ingatlankepek) lehet, és egy településen (telepulesek) található. A foglalasok összekötik a bérlőt, az ingatlant és a foglalási időszakot. A szerepkorok és jogosultsagok táblák határozzák meg, hogy melyik felhasználó mit tehet a rendszerben.

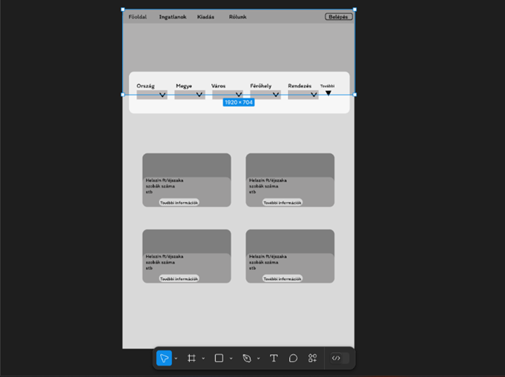
### Első elképzelések

#### Drótváz és vizuális terv

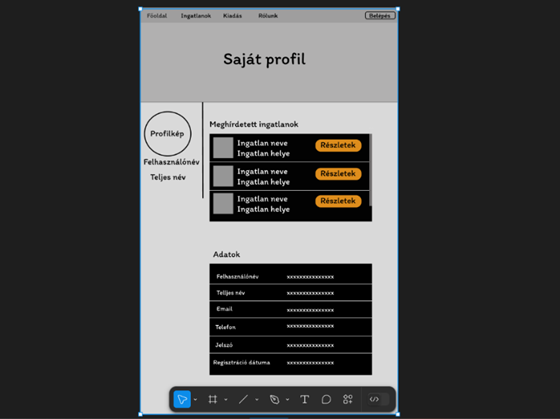
Első elképzeléseink alapján a weboldalunk a következőképpen épül majd fel, az elképzeléseinket próbáltuk a projektünk végére maximálisan megvalósítani.

* Főoldal (navbar, belépés gomb, népszerű városok)
* Belépés/ regisztráció aloldal

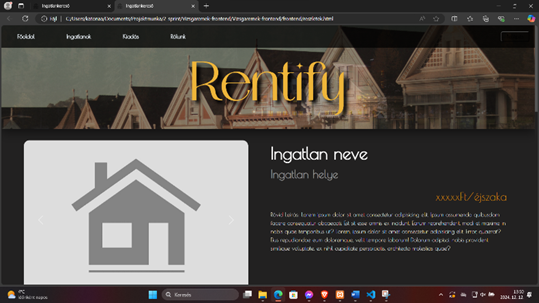




* Ingatlan részletek oldal (teljes leírás, kép(ek), foglalás gomb)
* Adminisztrációs felület tulajdonosoknak (ingatlanok listázása, új ingatlanok létrehozása, foglalás elfogadása vagy elutasítása)



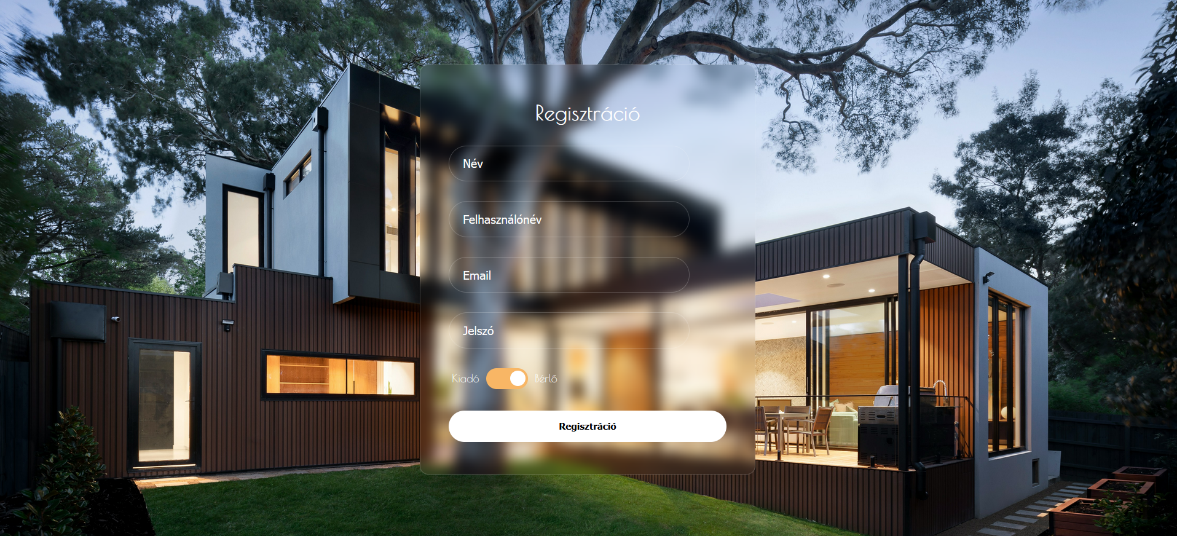
### React alapú weboldal

A weboldalt az elképzeléseinkhez hűen fejlesztettük tovább. Törekedve arra, hogy a legjobban hasonlítson a tervezetekre. A továbbiakban a weboldalról készített képernyőképek láthatóak.

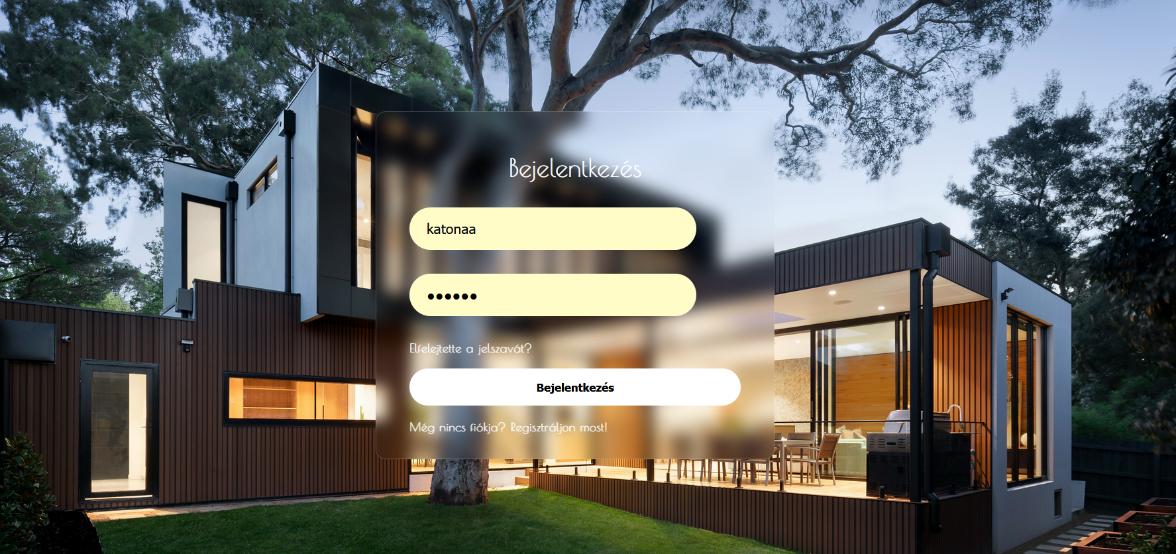
### Bejelentkezés és regisztráció

#### Regisztráció funkció

* A regisztráció során az új felhasználók két féle szerepkör közül választhatnak (tulajdonosok, bérlők).
* A regisztráció folyamata a következőképpen épül fel:
  + Kötelezően kitöltendő mezők (név, felhasználónév, e-mail cím, jelszó, slider a szerepkör kiválasztására)



#### Bejelentkezés funkció

* A bejelentkezés egy egyszerű, felhasználóbarát felületen történik, ahol a felhasználók felhasználónevük és jelszavuk megadásával férhetnek hozzá fiókjukhoz.
* A hitelesítés során a megadott adatokat valós időben ellenőrzi az adatbázisban tárolt, regisztráció során megadott értékekhez képest.
* Hibakezelés esetén, amennyiben a beírt felhasználónév vagy jelszó nem egyezik meg a nyilvántartott adatokkal, a rendszer figyelmeztető üzenetet küld.
  + Ha elfelejtettük a jelszavunkat, lehetőség van annak visszaállítására. A rendszer egy ideiglenes jelszót küld a regisztrált e-mail címünkre, amellyel egyszer be tudunk jelentkezni. Ezután a profiloldalon kötelező új jelszót megadnunk a biztonságos további használat érdekében.

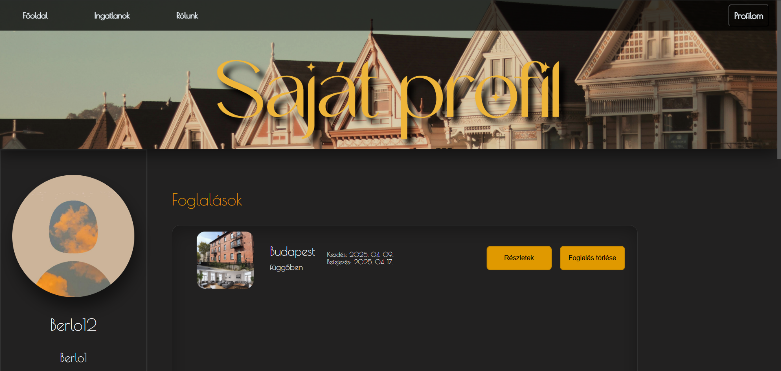
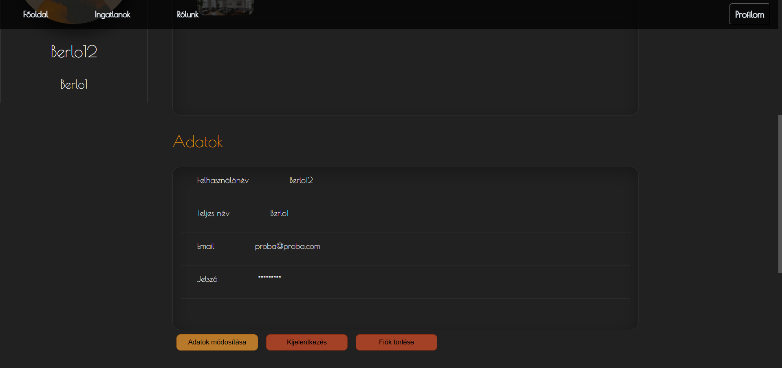
### Profiloldal

A profiloldal funkciójának célja, hogy a felhasználók könnyedén kezelhessék saját személyes adataikat, például nevüket, email-címüket vagy jelszavukat, valamint áttekintsék korábbi tevékenységeiket, feltöltött tartalmakat, ezzel elősegítve a személyre szabott és átlátható felhasználói élményt.

#### Felhasználói adatok megtekintése

A **profiloldalon** a regisztráció során megadott adatok jelennek meg, és a felhasználó lehetőséget kap ezek módosítására (például név, e-mail cím, jelszó stb.).

A regisztrációkor kiválasztott szerepkör (Kiadó vagy Bérlő) meghatározza, hogy milyen funkciók érhetőek el a felhasználó számára az oldalon.

* Kiadó szerepkör esetén:
  + A navigációs sávban megjelenik egy **„Kiadás”** fül, amely csak kiadóknak érhető el.
  + A „Saját profil” oldalon plusz funkciók válnak elérhetővé, például új ingatlan hozzáadása, meglévő ingatlan szerkesztése, valamint a beérkező foglalási kérelmek kezelése.
  + A kiadó dönthet arról, hogy **elfogadja vagy elutasítja** a beérkezett foglalásokat.
  + A foglalás állapotáról (elfogadás/elutasítás) a **bérlő e-mail értesítést kap** a regisztrált e-mail címére.
  + A foglaláskezelés a weboldalon keresztül történik, egyszerű és gyors felületen.
* Bérlő szerepkör esetén:
  + A „Kiadás” fül nem jelenik meg.
  + A bérlő **ingatlanokat kereshet,** megtekintheti a részletes leírást, képeket, árakat, valamint foglalási kérelmet küldhet be.
  + A bérlő a saját profiloldalán nyomon követheti foglalásait.
  + Amennyiben a foglalást a tulajdonos elfogadja vagy elutasítja, erről a bérlő **e-mail értesítést** kap.
  + A bérlő a foglalás részleteit is megtekintheti a felületen, például dátum, helyszín, státusz.

### Pontosabb szűrés elérése

#### Lokáció alapján

Idővel és a fejlesztéseink nyomán a pontosabb szűrés miatt az adatbázisban létrehozásra került a „telepulesek” tábla. A tábla tartalmazza a magyarországi megyék összes települését. A tábla négy mezőt tartalmaz; nev, megye, leiras és kép.

#### További szűrési feltételek

A **keresési élmény javítása érdekében** továbbfejlesztettük a részletes szűrési funkciót, amely lehetővé teszi a felhasználók számára, hogy még pontosabban és hatékonyabban találják meg a számukra ideális ingatlant. Az új szűrési lehetőségek segítségével az érdeklődők gyorsabban szűkíthetik a találatokat, és könnyebben eligazodhatnak a kínálatban.

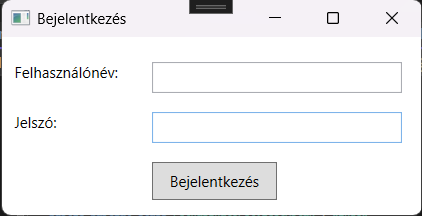
* A szűrési lehetőségek a következő szempontokat tartalmazzák:
  + Szobák száma: A felhasználók megadhatják, hogy hány szobás ingatlant keresnek – akár egy garzont, akár többszobás lakást. Így kizárhatják a nem releváns találatokat, és csak az igényeiknek megfelelő ingatlanokat látják.
  + Rendezés szerinti szűrés: Lehetőség van az ingatlanokat különböző szempontok szerint sorba rendezni, például:
    - ár
    - szobaszám
  + Extra szolgáltatások: A felhasználók szűrhetnek kényelmi funkciók szerint is.
* Ezen fejlesztés előnyei:
  + Gyorsabb döntéshozatal: a szűrés révén csak a releváns ingatlanokat látják a felhasználók, így nem kell feleslegesen átböngészniük a teljes kínálatot.
  + Nagyobb felhasználói élmény: az átlátható, jól strukturált szűrők javítják a navigációt, és csökkentik a kereséssel töltött időt.
  + Testreszabható keresés: mindenki a saját prioritásai szerint állíthatja be a szűrési feltételeket.

A részletes szűrési rendszer célja, hogy **egyszerűbbé, gyorsabbá és élvezetesebbé** tegye a tökéletes ingatlan megtalálását – akár bérlésről, akár kiadásról van szó.

### WPF karbantartó app

A karbantartási feladatok hatékony kezelése érdekében egy különálló, WPF-alapú asztali alkalmazást fejlesztettünk, amely kifejezetten az adminisztrátorok számára készült. Az alkalmazás célja, hogy egyszerűbbé, gyorsabbá és biztonságosabbá tegye az adatkezelést, különösen a háttérrendszerek felügyeletét és karbantartását illetően.

Az alkalmazás nyitóképernyője egy bejelentkezési felület, amely kizárólag meglévő felhasználók számára érhető el. A regisztráció lehetőségét eltávolítottuk, így az alkalmazás kizárólag adminisztrációs célokat szolgál.



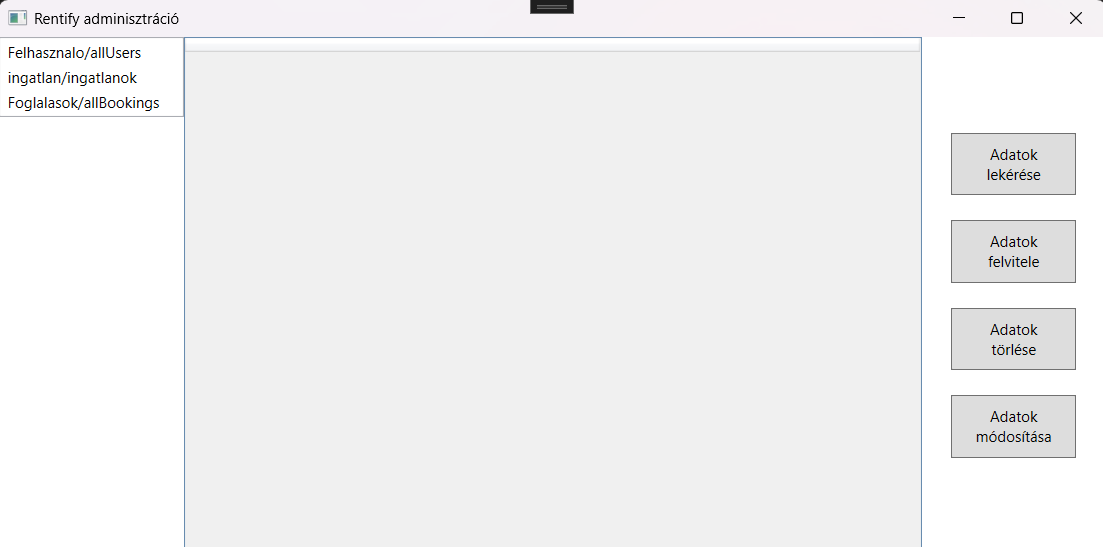
Az adminfelület intuitív és könnyen kezelhető grafikus felületet biztosít, amely lehetővé teszi a rendszerben tárolt adatok átlátható kezelését.

#### A rendszer az alábbi három fő adatbázistábla karbantartására lett kialakítva

* Felhasználók (felhasznalok)
* Ingatlanok (ingatlanok)
* Foglalások (foglalasok)

Az alkalmazásban ezekhez a táblákhoz teljes körű CRUD (Create, Read, Update, Delete) műveletek érhetők el, azaz:

* adatok lekérdezése és megtekintése
* új rekordok felvitele (pl.: új felhasználó hozzáadása)
* rekordok módosítása
* rekordok törlése a rendszerből



#### További funkciók

* Valós idejű visszajelzések a műveletek eredményéről (pl.: sikeres mentés, hibás adatbevitel).
* Jogosultságkezelés, amely biztosítja, hogy csak adminisztrátorok férhetnek hozzá a rendszerhez.

### React Native mobilapplikáció

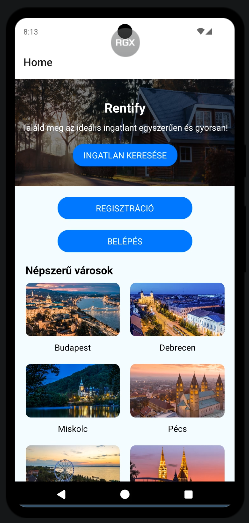
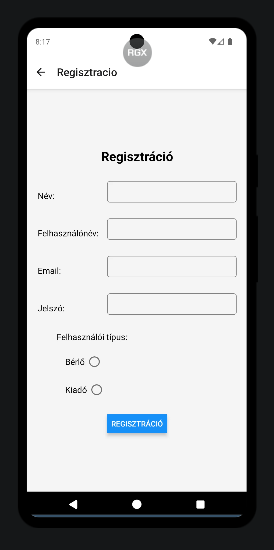
#### Ismertető

A React Native egy nyílt forráskódú fejlesztői keretrendszer, amelyet a Meta (korábban Facebook) fejlesztett ki. Lehetővé teszi mobilalkalmazások létrehozását JavaScript és React segítségével, natív élményt biztosítva mind iOS, mind Android platformokon. A React Native egyik legnagyobb előnye, hogy egyetlen kódbázisból készíthetők platformfüggetlen, mégis natív hatású, gyors és reszponzív alkalmazások. Emellett a fejlesztők közvetlen hozzáférést kapnak a natív eszközök és funkciók széles tárházához, ami tovább növeli a testreszabhatóságot és a teljesítményt.

#### Rentify mobilapplikáció

#### Projektünk célja az volt, hogy meglévő, már elkészített webes felületünket átültessük egy mobilalkalmazásba a React Native segítségével. Kezdetben megpróbáltuk közvetlenül átírni a webes projektünket úgy, hogy megfeleljen a React Native szintaxisának és követelményeinek. Azonban az első próbálkozásaink során világossá vált, hogy ez nem fog maradéktalanul működni, mivel a két technológia működési elve és struktúrája több ponton is eltér egymástól.

#### Ezért úgy döntöttünk, hogy a mobilalkalmazást gyakorlatilag újraalkotjuk az alapoktól, szem előtt tartva a React Native sajátosságait. A végeredmény így ugyan nem hasonlít mindenben a weblapos verzióhoz, de a lehetőségekhez mérten a lehető legjobb eredményt igyekeztünk elérni.

* Főbb oldalak és funkciók
  + Kezdőlap
    - A mobilalkalmazás kezdőlapja funkciójában megegyezik a webes verzióval, ugyanakkor dizájnban és elrendezésben különbözik. A gombok és interaktív elemek eltérő helyeken helyezkednek el, figyelembe véve a mobilos felhasználói élmény igényeit és az érintőképernyős navigáció sajátosságait.
  + Regisztráció
    - A regisztrációs oldal szintén eltér a webes megoldástól. A mobilnézet világosabb, letisztultabb, élénkebb színeket használ, és a felhasználói élmény optimalizálása érdekében könnyen átlátható és navigálható. A formok egyszerűbbek és kompaktabbak, így gyorsan és egyszerűen elvégezhető a regisztráció akár egy kézzel is.
  + További oldalak
    - A Rentify alkalmazásban elkészítettük a többi fontos oldalt is, mint például a bejelentkezés, bérléshez kapcsolódó oldal, profiloldal alapfunkciói. Bár ezek a nézetek funkcionálisan a webes verzió logikáját követik, több helyen módosításokat kellett eszközölni a mobilos platformhoz való alkalmazkodás miatt.

Fontos megjegyezni, hogy az alkalmazás jelenleg is fejlesztés alatt áll. Vannak olyan funkciók és oldalelemek, amelyek még nem működnek teljesen, vagy további optimalizálásra, hibajavításra szorulnak. Ezeket a fejlesztéseket a későbbiekben folyamatosan végre fogjuk hajtani. Célunk, hogy egy stabil, könnyen kezelhető, modern mobilalkalmazást hozzunk létre, amely a Rentify webes élményét tovább viszi a mobilplatformokra.

A mobilalkalmazás jelenlegi verziója elérhető és kipróbálható az alábbi linken keresztül: <https://snack.expo.dev/zOEAz6GGQLesL8KICsZnx>.

### Tesztelés és minőségbiztosítás

Miután úgy ítéltük, hogy a backend és a frontend oldali fejlesztések elérték azt a stabilitást, amely már lehetővé teszi a következő fejlesztési fázist, elkezdtük a tesztelési folyamatokat. Célunk az volt, hogy meggyőződjünk az alkalmazás funkcionalitásáról, hibamentes működéséről, valamint arról, hogy a felhasználói élmény megfelel az elvárt szintnek.

A tesztelés során különös figyelmet fordítottunk:

* az alapvető funkciók működésére (pl. regisztráció, bejelentkezés, adatbevitel),
* a felhasználói interakciókra,
* a hibakezelésre és visszajelzésekre,
* valamint a különböző eszközökön való megjelenésre és viselkedésre.

A tesztelés dokumentációját elérhetővé tettük nyilvánosan a GitHub-on, az alábbi linken:

🔗 [Tesztdokumentáció – GitHub](https://github.com/JNapsugar/Vizsgaremek/tree/teszt)

### Szerverre való kihelyezés

#### Adatbázis – Aiven

* Ismertető

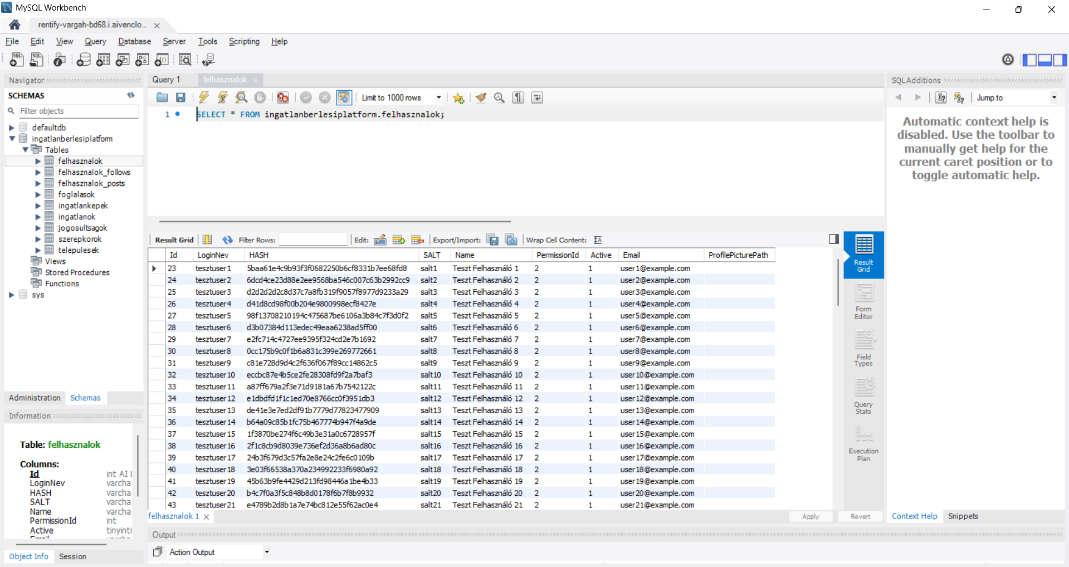
Az Aiven egy felhőalapú platform, amely menedzselt, nyílt forráskódú adatbázisokat és adatfeldolgozó szolgáltatásokat kínál, például PostgreSQL, Apache Kafka, Redis, OpenSearch és más népszerű technológiák formájában. Lehetővé teszi a vállalatok számára, hogy gyorsan telepítsenek, üzemeltessenek és skálázzanak adatkezelő rendszereket anélkül, hogy mélyebben bele kellene merülniük az infrastruktúra menedzselésébe.

Az Aiven támogatja a vezető felhőszolgáltatókat (AWS, Google Cloud, Azure stb.), így a felhasználók szabadon választhatnak régiót és szolgáltatót, megőrizve a rugalmasságot és elkerülve a vendor lock-in-t. A platform kiemelten kezeli a biztonságot, magas rendelkezésre állást.

Az Aiven ideális választás azoknak a fejlesztőcsapatoknak, akik nyílt forráskódú megoldásokat használnának professzionális környezetben, gyorsan szeretnének piacra lépni, és közben csökkentenék az üzemeltetési terheket.

* Választás oka

A fent említett előnyök – mint a nyílt forráskódú technológiák támogatása, a többfelhős infrastruktúra rugalmassága, a magas rendelkezésre állás, valamint a könnyen kezelhető és biztonságos környezet – mind hozzájárultak ahhoz, hogy az **Aiven** platform mellett döntsünk. Az Aiven lehetővé teszi számunkra, hogy megbízható adatkezelő rendszereket üzemeltessünk anélkül, hogy az infrastruktúra karbantartása túlzottan leterhelné a fejlesztői folyamatokat. Így több időt és erőforrást fordíthatunk az alkalmazás valódi értéket teremtő funkcióinak fejlesztésére, miközben biztosak lehetünk abban, hogy adataink biztonságban és stabil környezetben futnak.



#### Backend – Render

* Ismertető

A Render egy modern, fejlesztőbarát felhőalapú platform, amely lehetővé teszi webalkalmazások, statikus oldalak, háttérszolgáltatások és adatbázisok egyszerű telepítését és üzemeltetését. Támogatja a legnépszerűbb nyelveket (pl. Node.js, Python, Ruby), automatikus CI/CD integrációval, skálázással és beépített biztonsági funkciókkal (pl. TLS, DDoS védelem). A GitHub-bal való integrációnak köszönhetően gyors és automatizált a fejlesztési folyamat. Ideális választás azok számára, akik egy egyszerű, de rugalmas alternatívát keresnek a nagyobb felhőszolgáltatókkal szemben.

* Választás oka

A fent említett előnyök – mint a könnyű telepítés, az automatikus skálázás, a beépített biztonsági funkciók, valamint a GitHub-bal való zökkenőmentes integráció – mind hozzájárultak ahhoz, hogy a Render platformot válasszuk backend szolgáltatásaink hosztolására. A Render lehetővé tette számunkra, hogy gyorsan és egyszerűen telepítsük és üzemeltessük alkalmazásunkat, miközben nem kellett a háttérinfrastruktúra karbantartásával tölteni az időnket. A platform menedzselt adatbázisai és a magas rendelkezésre állás biztosítása különösen fontos volt számunkra, mivel ezek a stabilitást és a megbízható működést garantálták a backend rendszerek számára. A Render rugalmassága és skálázhatósága ideálissá tette a választást, különösen akkor, amikor a projekt növekedésével párhuzamosan könnyedén alkalmazkodni tudtunk az új igényekhez.

#### Frontend – Render

* Választás oka

A Render.com-ot választottuk a frontendünk hosztolására, mert számos előnyt kínál, amelyek segítik a gyors és hatékony fejlesztést. Az egyszerű használat és az automatikus deploy folyamatok lehetővé teszik, hogy gyorsan és zökkenőmentesen telepíthessük alkalmazásunkat. A platform automatikusan skálázza az alkalmazást a forgalom alapján, így biztosítva a folyamatos rendelkezésre állást és a kiváló teljesítményt globálisan.

A Render könnyed integrációt biztosít más backend szolgáltatásainkkal és API-kkal, így egységes fejlesztői workflow-t hozunk létre. Emellett költséghatékony megoldást kínál, mivel az erőforrásokat dinamikusan kezeli, elkerülve a felesleges költségeket, miközben biztosítja a gyors betöltést és a megbízható felhasználói élményt.

Összességében a Render.com lehetővé teszi számunkra, hogy gyorsan, költséghatékonyan és skálázható módon hosztoljuk a frontendünket, miközben magas szintű felhasználói élményt biztosítunk.

# Irodalomjegyzék

1. HTML5 és CSS3

* HTML5: <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>
* CSS3: <https://developer.mozilla.org/en-US/docs/Web/CSS>

1. Bootstrap

* Hivatalos dokumentáció: <https://getbootstrap.com/docs/>

1. WPF - Windows Presentation Foundation

* Microsoft Learn: <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/>

1. C# nyelv

* Microsoft C# dokumentáció: <https://learn.microsoft.com/en-us/dotnet/csharp/>

1. .NET (Core, 5, 6, 7)

* Hivatalos oldal: <https://dotnet.microsoft.com/>

1. ASP.NET Core

* Microsoft dokumentáció: <https://learn.microsoft.com/en-us/aspnet/core/>

1. SQL Server / MySQL / PostgreSQL / SQLite

* SQL Server: <https://learn.microsoft.com/en-us/sql/sql-server/>
* MySQL: <https://dev.mysql.com/doc/>
* PostgreSQL: <https://www.postgresql.org/docs/>
* SQLite: <https://www.sqlite.org/docs.html>

1. Entity Framework Core

* Microsoft EF Core dokumentáció: <https://learn.microsoft.com/en-us/ef/core/>

1. REST API

* RESTful API alapelvek: <https://restfulapi.net/>

1. React és React Native

* React: <https://reactjs.org/docs/getting-started.html>
* React Native: <https://reactnative.dev/docs/getting-started>

1. Aiven – menedzselt adatbázis

* Aiven hivatalos oldal: <https://aiven.io/>

1. Render – felhőszolgáltatás

* Render platform: https://render.com/docs