

Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Drexler Nándor

**Ingatlan hirdető portál megvalósítása React és .NET Core alapokon**

Dr. Ekler Péter

BUDAPEST, 2020

Tartalomjegyzék

[Összefoglaló 5](#_Toc433184091)

[Abstract 6](#_Toc433184092)

[1 Bevezetés 7](#_Toc433184093)

[1.1 Frissítsd a dokumentumot 8](#_Toc433184094)

[1.2 Szakdolgozat, vagy diplomaterv 8](#_Toc433184095)

[1.3 Témaválasztás 8](#_Toc433184096)

[2 A dolgozat szerkezete 10](#_Toc433184097)

[2.1 Fejezetek 10](#_Toc433184098)

[2.1.1 Feladatkiírás 10](#_Toc433184099)

[2.1.2 Címoldal 10](#_Toc433184100)

[2.1.3 Tartalomjegyzék 10](#_Toc433184101)

[2.1.4 Nyilatkozat 11](#_Toc433184102)

[2.1.5 Tartalmi összefoglaló 11](#_Toc433184103)

[2.1.6 Bevezetés 11](#_Toc433184104)

[2.1.7 Irodalomkutatás, technológiák, hasonló alkotások bemutatása 12](#_Toc433184105)

[2.1.8 A feladatkiírás pontosítása és részletes elemzése 12](#_Toc433184106)

[2.1.9 Önálló munka bemutatása 12](#_Toc433184107)

[2.1.10 Önálló munka értékelése, mérések, eredmények bemutatása 13](#_Toc433184108)

[2.1.11 Összefoglaló 14](#_Toc433184109)

[2.1.12 Köszönetnyilvánítások 14](#_Toc433184110)

[2.1.13 Részletes és pontos irodalomjegyzék 14](#_Toc433184111)

[2.1.14 Ábrajegyzék, táblázatjegyzék 14](#_Toc433184112)

[2.1.15 Függelék 14](#_Toc433184113)

[2.2 Egyéb tartalmi elemek 15](#_Toc433184114)

[2.2.1 Stílus 15](#_Toc433184115)

[2.2.2 E/1 15](#_Toc433184116)

[2.2.3 Rövidítések 15](#_Toc433184117)

[2.2.4 Technológia megválasztása 16](#_Toc433184118)

[3 Formázási tudnivalók 17](#_Toc433184119)

[3.1 Általános tudnivalók 17](#_Toc433184120)

[3.2 Stílusok 17](#_Toc433184121)

[3.3 Címsorok 18](#_Toc433184122)

[3.4 Másolás, beillesztés 18](#_Toc433184123)

[3.5 Mezőfrissítés 19](#_Toc433184124)

[3.6 Helyesírás 19](#_Toc433184125)

[3.6.1 Elgépelések 19](#_Toc433184126)

[3.6.2 Egyeztetés hiánya 19](#_Toc433184127)

[3.6.3 Külföldi szavak, kifejezések 19](#_Toc433184128)

[3.6.4 Stb 20](#_Toc433184129)

[3.6.5 Helyesírás ellenőrző 20](#_Toc433184130)

[3.7 Képek 21](#_Toc433184131)

[3.7.1 Beszúrás, formázás 21](#_Toc433184132)

[3.7.2 Képminőség 21](#_Toc433184133)

[3.8 Kereszthivatkozások 22](#_Toc433184134)

[3.9 Irodalomhivatkozások 23](#_Toc433184135)

[3.9.1 Pozícionálás 23](#_Toc433184136)

[3.9.2 Mikor kell hivatkoznom? 24](#_Toc433184137)

[3.10 Word tippek és trükkök 24](#_Toc433184138)

[3.10.1 Navigációs ablak 24](#_Toc433184139)

[3.10.2 Megjegyzések 24](#_Toc433184140)

[3.10.3 Korrektúra 25](#_Toc433184141)

[3.10.4 Gyorsbillentyűk 25](#_Toc433184142)

[3.11 Kódrészletek 25](#_Toc433184143)

[3.11.1 Formázás 26](#_Toc433184144)

[3.11.2 Irodalomjegyzék 27](#_Toc433184145)

[3.12 Utolsó simítások 27](#_Toc433184146)

[4 Irodalomjegyzék 28](#_Toc433184147)

[Függelék 29](#_Toc433184148)

Hallgatói nyilatkozat

Alulírott **Drexler Nándor**, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző, cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2020. 10. 08

...…………………………………………….

Drexler Nándor

Összefoglaló

Az ingatlanvásárlás mindenki életebén egy meghatározó mérföldkő. Ahány ember annyi különböző elképzelés a megfelelő otthonról, nem beszélve a jelentős összegekről, akár hitelekről, amelyek a vásárlót hosszú évekre röghöz kötik. Ugyanakkor az eladói oldalon is jelentős kihívás megtalálni a megfelelő vevőt, aki értékeli az ingatlan adottságai, belefektetett munkát és költségeket, ennek következtében, alkudozások helyett hajlandó kifizetni egy a tervezett eladási árnak megfelelő összeget. Manapság már általában nem (csak) az újságból vagy ingatlanirodában tájékozódnak az emberek, hanem az interneten, így az ingatlanreferensek sem mehetnek el szótlanul az internetes hirdetőportálok mellet.

A „RealState” webalkalmazás a fent említett kihívásokban próbál segíteni mind a vásárlóknak, mind az eladóknak. Részletes keresési lehetőséget kínál, ahol nem csak adatok alapján kereshetünk, hanem elhelyezkedés alapján is. Miután a vásárlók nem csak puszta adatok alapján hozhatók lázba, ezért kedvcsináló képeket tölthetünk fel a hirdetésekhez. Illetve, abban az esetben, ha nagyon elburjánzik az ingatlanok kínálata, nem valószínű, hogy minden oldalt végig lapoznak az érdeklődők. Virtuális kreditekért cserébe kiemelhetjük, még több érdeklődőt megtalálva ezzel. Közvetlenül a portálon írhatunk üzenetet a hirdetőknek, megkönnyítve ezzel a kapcsolatfelvételt.

Egy jól használható hirdetőportál hamar népszerűségre tehet szert a felhasználók körében. A felhasználó elégedettség itt nem csak a konkrét felhasználói élményben mérhető. Másszóval nem elegendő, hogy gyorsan betöltsön az oldalam, illetve intuitív felületek legyenek. Vevői szemszögből meghatározó lehet, hogy mennyi idő alatt találtam nekem megfelelő ingatlant, illetve eladó szemszögből, hogy milyen áron, mennyi idő alatt tudtam eladni az ingatlanom. Ezen úgy lehet segíteni, hogy amellett, hogy az alkalmazás kedvező felhasználói élményt ad, fontos a folyamatos fejlesztés, az új funkciók hozzáadása, hogy az aktuális trendeket követve megtartsuk hosszútávon a felhasználóbázisunkat és mindig legyen megfelelő mennyiségű hirdetés az oldalon.

Abstract

Purchasing a real estate is remarkable achievement in everyone’s life. Every single person has different desires of a comfortable home. The enormous costs makes people take loans, forcing them to stay in their chosen apartment for decades. Sellers are facing the challange of finding clients who actually acknowledges the features of the apartment and also the invested money and time to build or renovate the certain estate. Therefore the client leaves the haggling and offers a reasonable price for the estate. Nowadays people favor browsing estates on the internet instead of looking for them in the newspaper or going to real estate agency. Real estate agents have to take the online advertisement possibilities into account.

The „RealState” web application offers help for the challanges mentioned above, both the client and seller side. It provides a various search features, not only for certain properties of the estates, but allowing to show them by their location on the map. Since most clients cannot be attracted by promising detailed information, sellers have the ability to upload spectacular pictures, which cheers up the client’s interest. In most cases, while clients are browsing on the internet, they won’t check all the pages of their search’s results due to lack of time. In exchange of virtual credits, we can highlight our advertisement in order to make it reach more people. Finally, we can contact the seller by sending a message directly on the page.

A well-designed, easy to use real estate portal can soon achieve a high popularity among users. Here, the user’s satisfaction does not only consist of good user experience. In other words a fast, responsive and intuitive web page is not enough. From the buyers point of view the time required to find a real estate which appeals to them is really significant. This amount of time can be reduced by certain features, however, these features are only effective if we maintain the necessary amount of active advertisements on the portal and also keep our users in the long run. This requires constans development, along popular trends and techniques.

# Bevezetés

Egyetemi tanulmányaim során kifejezetten tetszettek a webfejlesztéssel kapcsolatos oktatóanyagok és feladatok, valamint a munkaerőpiacon is elég keresettek a webes technológiákhoz értő szakemberek. Ezen két tény segített meghozni a döntés, hogy valamilyen webes témával és abból is a korszerű és gyakran alkalmazott technológiákkal szeretnék foglalkozni.

Egyetemi pályafutásom alatt, több különböző albérletben laktam. Mindegyiknek megvoltak a maga előnyei és hátrányai. Például egy a legfelső emeleten elhelyezkedő lakásban általában elég meleg van nyáron, cserébe a nincs felső szomszéd, aki lehetetlen időpontokban hangosan sétáljon felettünk. Egy másik érdekes igény lehet, ha például fontos, hogy legyen a lakásunkhoz közel zöld terület, iskola, óvoda vagy gyógyszertár, akkor egy egészen másfajta keresési módot kell alkalmaznunk, amit a legtöbb már létező ingatlan hirdető portál nem támogat. Itt fogant meg először az ötlet, hogy egy ilyen ingatlanhirdető portált én magam is megpróbáljak implementálni.

## Felhasznált technológiák jelentősége

A mai világban egyre nagyobb hangsúlyt kapnak a vékonykliens alkalmazások. Ez nem is csoda, hiszen mivel minden modern böngészőt futtató eszköz képes ezek futtatására. A legújabb *HTML (HyperText Markup Language)* és CSS (*Cascading Style Sheets*) szabványok támogatják a különböző kijelző méreteket és felbontásokat, így nem csak asztali számítógépen, hanem mobil és tablet eszközökön is kényelmes, intuitív a megjelenő felület. A *JavaScriptnek* köszönhetően a weboldalak innentől kezdve nem statikus dokumentumok, hanem felhasználó interakciókat kezelő dinamikus felületek. Elméletben ez remekül hangzik, azonban manapság ritkán fejlesztünk így. A gyakorlatban sokkal hatékonyabb a fejlesztés és karbantarthatóbb a kód, ha van egy keretrendszer, ami a fent említett három különböző elemet összefogja. Ilyen keretrendszerek például az Angular, Vue.js vagy amit jelen alkalmazáshoz használtam a React.js. Ez utóbbi a manapság legelterjedtebb ilyen keretrendszer. Az 1. ábrán betekintést nyerünk az elmúlt félévben, hogy alakult a legnépszerűbb keretrendszerek letöltéseinek száma.



1. ábra: Legnépszerűbb *JavaScript* keretrendszerek letöltésének száma [1]

A szép felület azonban nem elég a hosszútávon fenntartható működéshez. A bővíthetőség, biztonság és a sebesség is meghatározó tényezők ma már. Ezen felül egy ilyen alkalmazásnál fel kell készülni a módosításokra is. A változó trendekhez, elvárásokhoz minél gyorsabban célszerű alkalmazkodni, illetve folyamatosan fenntartani a fejlesztést, a funkciók bővítését. A biztonságot elsősorban szerver oldali validációk sokaságával, a gyorsaságot pedig a backend teljesítménye mellett, megbízható adatbáziskezelővel lehet garantálni. Adatbázisok tekintetében még mindig a relációs adatbázisok a legelterjedtebbek, azonban számottevő a séma nélküli dokumentum alapú (másnéven NoSQL) adatbázisok. Fő előnyük a relációs adatbázisokkal szemben, hogy általában nincs meghatározott séma, ezáltal nincs szükség séma migrációra például egy új funkció hozzáadása esetén és jóval egyszerűbb a már élesben működő alkalmazás továbbfejlesztése. Ezen előnye miatt döntöttem a NoSQL vonal és azon belül is a MongoDb mellett.



. ábra: Relációs és dokumentum alapú adatbáziskezelők piaci részesedése (2019) [2]

A szerver oldali validációkat általában a backend üzleti logikája végzi, ezzel megőrzi az adatbázis konzisztenciáját, illetve meggátolja az illetéktelen hozzáféréseket. Még egy alapvető és fontos követelmény, hogy titkosított legyen a kapcsolat (*https*) a webszerver és a kliens között, ezáltal elkerülhető, hogy a felhasználók érzékeny adatait, például jelszavakat, e-mail címeket, bankkártya adatokat egy harmadik fél is megismerjen. Korábbi tapasztalataim alapján az *Asp.Net Core* backend mellett sok érv szólt. A legfőbb érv mellette mégis az volt, hogy könnyen felkonfigurálható az előző fejezetben említett MongoDb adatbáziskezelőre.

## Szakdolgozat tartalma

A dokumentum további fejezetiben a következő témák találhatóak:

* A 2. fejezetben a feladatspecifikációról fogok írni, melyben a feladat részletes leírásán túl a program funkcióit foglalom össze és Use Caseket szemléltetem diagramokon.
* A 3. fejezetben részletezem az irodalomkutatást, a megvalósításhoz felhasznált technológiákat, illetve más lehetséges alternatív megoldásokat is, valamint az indoklást egy adott technológia választása mellett.
* A 4. fejezetben betekintést adok a felsőszintű architektúrába, szemléltetem a rendszer felépítését, különböző komponenseit.
* Ezután az 5. fejezetben a részletes megvalósítás következik. Osztálydiagramon, szekvencia diagramokon és egyéb UML-es eszközökkel szemléltetem az alkalmazás működésének elvét.
* A 6. fejezetre maradnak a tesztelések, valamint a felhasználói leírás, képernyőképekkel.
* Végül a 7. fejezetben összefoglalom a munkám eredményét és felvetek néhány továbbfejlesztési lehetőséget.

# Feladatspecifikáció

## Feladat részletes leírása

A feladat egy ingatlan hirdető webalkalmazás architektúrájának kidolgozása, majd implementációja. A portálon legyen lehetőség hirdetések listázására, keresésésre, rendezésére többféle szempont alapján. Ezen szempontok lehetnek az ingatlan címe (település, kerület), az ingatlan ára minimum és maximum értékkel intervallum keresés, az ingatlan típusa (családi ház, lakás, ikerház, stb. ), valamint a hirdetés leírásában található kulcsszavak alapján. A felhasználók tudjanak feladni hirdetéseket, azokat szerkeszteni illetve törölni. A hirdetésekhez képek töltehetők fel, illetve virtuális kreditért cserébe kiemelhetőek, előresorolhatóak. Legyen lehetőség megadni a hirdetett ingatlanok pontos elhelyezkedését a térképen és lehessen ezen elhelyezkedés alapján is keresni ingatlanokat. Azon felhasználóknak amelyeknek van aktív hirdetésük, egyenként láthassák, hogy melyik hirdetést hányan tekintették meg. A felhsználók tudjanak üzenetet küldeni egymásnak, a könnyebb kapcsolatfelvétel érdekében.

A feladat részét képezi, bizonyos nem funckionális követelmények feltárása, illetve implementálása. A weboldalon nem lehetnek statikus újratöltések. Szükséges a különböző kijelzőméretek támogatása. A bejelentkezés és a regisztráció miatt elkerülhetetlen, hogy a backend biztonságos (*https*) kapcsolatot használjon. Az adatok validációját kliens és szerver oldalon is el kell végezni. Továbbá az authorizáció és autentikáció kimaradhatatlan lépés bizonyos funkciók esetében. Például a hirdetést csak az a felhasználó szerkeszthesse vagy törölhesse aki ténylegesen fel is adta korábban.

## Használati esetek

Ez az alfejezet a tervezés legelső fázisát tartalmazza. Felmérjük, hogy a felhasználó milyen interakciókat végezhet az alkalmazásunkkal, ezzel segítve a tervezés későbbi fázisait.

### Hirdetések keresése

Alapvető elvárás, hogy a felhasználók meg tudják tekinteni az aktív hirdetéseket és keresni is tudjanak közöttük, a korábban említett paraméterek alapján. Ezt a funkciót nem szükséges még bejelentkezéshez sem kötni. A rendezési funkció is megtalálható szinte már minden hirdető portálon. Általában az ingatlanok esetében három fő rendezési elvet szoktunk követni a feltöltés ideje, a vételár, vagy az alapterület mérete szerint. Első körben csak a „legújabb”, az „ár szerint növekvő” és „ár szerint csökkenő” rendezéseket terveztem bele, de a lehetőséget mindenképp fenntartom, további rendezési elvek bevezetésére.

### Hirdetés feladása, módosítása

Hogyha tovább folytatom, gondolatmenetemet hamar nyilvánvalóvá válik, fel is kell adni azokat a hirdetéseket a felhasználóknak. A hirdetés feladását célszerű mindenképp egy előzetes regisztrációhoz, valamint bejelentkezéshez kötni. Az elgépeléseken túl előfordul, hogy menet közben csökkenteni szeretné a felhasználó a meghirdetett irányárat, ha például nagyon csekély az érdeklődés és egyre sürgetőbb lenne az eladás, ezért erre is biztosítani kell a lehetőséget. A hirdetések szerkeszthetőségéhez szorosan kapcsolódik a követelményekben felvázolt képfeltöltés és hirdetés kiemelése funkció, tehát jó eséllyel ezen funkciók is módosítás felületről legyenek elérhetőek.

### A hirdető adatlapja és értékelése

Megfogalmazódott bennem egy olyan ötlet, hogy ha már regisztráció szükséges a hirdetések feladásához, lehessen értékeléseket leadni a felhasználókra és ezek az értékelések nyilvánosan is megtekinthetőek legyenek. Ez nagyban segíti az új felhasználókat és esetleg felesleges köröket spórolhat meg egy megbízhatatlan eladó esetén. Három fajta értékelés adható le, pozitív, negatív és semleges, mindegyikhez fűzhető egy komment, amelyben a felhasználó kifejezheti elégedettségét, avagy részletezheti mivel volt probléma. Ezen értékelések megtekintését nem tartottam szükségesnek regisztrációhoz kötni, ellenben értékelést írni csak bejelentkezett felhasználó tud.

### Use Case diagram

A feladat felmérése után az alábbi (2. árbán) látható *use case* diagramot készítettem. Bár a funkcionalitást nem fedi le teljes egészeben, a lényeges felhasználói interakciókat tartalmazza. A struktúrát igyekeztem úgy felépíteni, hogy tükrözze az alkalmazás felületének struktúráját. Például, ha a hirdetés feladása felület csak a bejelentkezés után érhető el, a hirdetés módosítása (és annak további funkciói) pedig csak azután, ha már legalább egy hirdetést feladtunk.



3. ábra: Use case diagram

# Felhasznált technológiák

Manapság egy webalkalmazás alatt nem a klasszikus értelemben vett weboldalt értjük. A modern weboldalak ma már alapvetően az *SPA (Single Page Application)* megközelítést alkalmazzák, ugyanis így natív érzetű gyors, reszponzív webes felületet kapnak a felhasználók. Ahhoz, hogy ez működőképes legyen mindenképp szükség van egy webszerverre, amely kiszolgálja a kliensek kéréseit, illetve gondoskodik a felhasználók hitelesítéséről és az adatok validációjáról. Miután nagy mennyiségű adatokkal dolgozunk, elengedhetetlen egy megbízható adatbáziskezelő rendszer.

## MongoDb

Mint korábban említtetettem a relációs adatbáziskezelők mellett jelentős a dokumentum alapú adatbáziskezelők térnyerése. Egy ilyen legnépszerűbb dokumentum alapú adatbáskezelőt, a *MongoDb*-t választottam. Ez egy nyílt forráskódú, platformfüggetlen adatbázis szoftver, a dokumentumokat egy *BSON (Binary JSON)* formátumban tárolja, amely szintaktikájában nagyon hasonlít a *JSON (JavaScript Object Notation)* formátumhoz. [3] A *MongoDb* támogatja a keresést mező alapján, érték-tartomány alapján, vagy reguláris kifejezéssel, valamint lehetővé teszi, hogy a dokumentum bármelyik mezője alapján indexet készítsünk. [4] Amennyiben grafikus felületen keresztül szeretnénk használni az említett adatbáziskezelőt, akkor a *Robo3t* nevű program lehetőséget ad erre.

## Asp .Net Core Web Application

### MongoDb Driver

### AspNetCore Identity MongoDbCore

## React.js

## Bootstrap

## Google Maps

# Irodalomjegyzék

|  |  |
| --- | --- |
| [1] | S. Martin, „The Top JavaScript Frameworks For Front-End Development in 2020,” [Online]. Available: https://www.freecodecamp.org/news/complete-guide-for-front-end-developers-javascript-frameworks-2019/. [Hozzáférés dátuma: 14 október 2020]. |
| [2] | ScaleGrid, „2019 Database Trends – SQL vs. NoSQL, Top Databases, Single vs. Multiple Database Use,” 4 március 2019. [Online]. Available: https://scalegrid.io/blog/2019-database-trends-sql-vs-nosql-top-databases-single-vs-multiple-database-use/. [Hozzáférés dátuma: 15 október 2020]. |
| [3] | MongoDb, „JSON and BSON,” [Online]. Available: https://www.mongodb.com/json-and-bson. [Hozzáférés dátuma: 15 10 2020]. |
| [4] | Wikipédia, „MongoDB,” [Online]. Available: https://hu.wikipedia.org/wiki/MongoDB. [Hozzáférés dátuma: 15 10 2020]. |
| [5] | w3schools.com, „JSON - Introduction,” [Online]. Available: https://www.w3schools.com/js/js\_json\_intro.asp. [Hozzáférés dátuma: 15 10 2020]. |

Függelék