Debreceni SZC Baross Gábor Technikum, Szakképző Iskola és Kollégium

Reent it

SZAKDOLGOZAT

készítette:

Marinka Szilveszter Richárd

Hagyó Csaba

Cseh Márton Dávid

Debrecen  
2025

Tartalomjegyzék

[1. Bevezető 1](#_gjdgxs)

[2. Elméleti háttér 2](#_30j0zll)

[2.1. A feladathoz tartozó egyik témakör bemutatása 2](#_44sinio)

[2.2. Elméletek 3](#_1fob9te)

[2.2.1. A tervezéshez tartozó elméleti, tudományos háttér 3](#_2jxsxqh)

[2.2.2. A megvalósításhoz tartozó elméletek és módszertanok 4](#_z337ya)

[2.3. Módszerek, algoritmusok 5](#_3znysh7)

[3. A fejlesztett online bank webhely bemutatása 6](#_2et92p0)

[3.1. Előkészítés, tervezés 6](#_tyjcwt)

[3.2. A megvalósításban jelentős részek 6](#_3dy6vkm)

[3.3. Az elkészült webhely bemutatása funkcuonálisan 6](#_1t3h5sf)

[4. Összegzés 7](#_4d34og8)

[5. Irodalomjegyzék 8](#_2s8eyo1)

[6. Függelék 10](#_1ksv4uv)

# Bevezető

Mindhárman egy fontos élethelyzet előtt állunk: egyetemre készülünk, méghozzá Budapestre. Az oda költözéshez elengedhetetlen, hogy találunk egy megfelelő albérletet, de amikor nekiláttunk a keresésnek, gyorsan rájöttünk, hogy a jelenlegi albérletkereső oldalak több szempontból sem ideálisak.

Az egyik legnagyobb probléma, hogy ezek az oldalak nem nyújtanak elegendő információt ahhoz, hogy megbízható döntést hozhassunk. Például nem lehet tudni, milyen értékeléseket kapnak a lakást kiadó tulajdonosok, ami kockázatos helyzetekhez vezethet. Emellett az ingatlanok pontos elhelyezkedése sem mindig elérhető, így nehéz megítélni, hogy egy adott lakás mennyire felel meg a közlekedés szempontjainknak. A reklámok jelenléte tovább rontja a felhasználói élményt, és a legtöbb oldal kezelőfelülete messze nem mondható felhasználóbarátnak.

Ezeket a problémákat látva úgy döntöttünk, hogy elkészítjük a saját albérletkereső oldalunkat, amely a szoftverfejlesztői ismereteinket kihasználva igyekszik megoldani ezeket a hiányosságokat.

A platformunk célja, hogy egyszerű és intuitív legyen, ugyanakkor pontos és megbízható adatokat nyújtson a felhasználóknak. Az oldalunk lehetővé teszi a tulajdonosok értékelését, hogy a bérlők biztosabb döntést hozhassanak, pontos térképes megjelenítést biztosít az ingatlanok elhelyezkedéséről, teljesen mentes a zavaró reklámoktól, a modern felhasználói felület (UI) és a zökkenőmentes felhasználói élmény (UX) érdekében készült.

A fejlesztés során kiemelt figyelmet fordítunk a tervezésre és az egyszerű kezelhetőségre, hiszen az a célunk, hogy az oldal használata ne csak praktikus, hanem élmény dús is legyen. Ez a projekt nemcsak a saját albérletkeresési tapasztalatainkat alapozza meg, hanem egy olyan eszközt hoz létre, amely szélesebb körben is hasznos lehet.

A projekt fejlesztése során több fontos lépést tettünk meg annak érdekében, hogy az általunk tervezett albérletkereső oldal a felhasználói igényeknek maximálisan megfeleljen. A kezdeti szakaszban alapos piackutatást végeztünk, amely során számos meglévő albérletkereső platformot elemeztünk, hogy megértsük, mi az ami működik, és mi az ami nem. A kutatás eredményei alapján egyértelművé vált, hogy a felhasználók leginkább az alábbi három szempontot tartják fontosnak: megbízhatóság, könnyű használhatóság és részletes információk.

Ezeket a szempontokat figyelembe véve kezdtük el a weboldal tervezését. A felhasználói felület (UI) kialakításánál nagy hangsúlyt fektettünk a letisztult és modern dizájnra. Célunk, hogy a felhasználók könnyedén navigálhassanak az oldalon, és gyorsan rátaláljanak a keresett ingatlanokra. Az oldalra látogatva a felhasználók azonnal rálátnak a legfontosabb információkra: az ingatlan elhelyezkedésére, ár-érték arányára, a bérlői és tulajdonosi véleményekre, és az ingatlan állapotára. A térképes megjelenítés a pontos elhelyezkedést biztosítja, így a bérlők könnyen megítélhetik, hogy az adott ingatlan mennyire van közel az egyetemhez, a közlekedési csomópontokhoz, vagy éppen a szükséges üzletekhez.

A felhasználói élmény (UX) javítását célzó tervezési döntések között szerepelt a gyors keresés és szűrési lehetőségek beépítése is. A bérlők szűrhetik az ingatlanokat több szempont szerint is: ár, elhelyezkedés, szoba szám, felszereltség, és egyéb fontos paraméterek. Ezen kívül az oldal folyamatosan frissíti a hirdetéseket, így a felhasználók biztosak lehetnek abban, hogy mindig naprakész információkat látnak.

A "clean code" elvét követve a fejlesztés során arra törekszünk, hogy a kód átlátható, könnyen érthető és könnyen módosítható legyen. A változó- és függvényneveket egyértelmen választjuk, hogy bárki könnyen megértse, mit csinálnak. A kódot egyszerűsítjük és rendszerezzük, hogy a későbbi fejlesztések és javítások könnyebben elvégezhetők legyenek. Így biztosítjuk, hogy a projekt hosszú távon is jól működjön. Emellett a kód strukturálásával és tisztaságával csökkentjük a hibák lehetőségét, és gyorsabbá tesszük a munkát. A fejlesztők számára könnyen érthető és módosítható kód biztosítja a projekt hatékony kezelését a jövőben is.

|  | import { createClient } from "@supabase/supabase-js";    const supabase = createClient(process.env.NEXT\_PUBLIC\_SUPABASE\_URL, process.env.NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY);    export default supabase; |  |
| --- | --- | --- |

*1sd*

A projekt egyik legnagyobb kihívása volt, hogy a program asztali számítógépen kívül minden platformon, így mobiltelefonokon és táblagépeken is zökkenőmentesen működjön. Ehhez reszponzív dizájnt alkalmaztunk, amely automatikusan igazodik a képernyőmérethez, így a felhasználói élmény nem csökken a különböző eszközök használata során.

A fejlesztés során végig tartottuk szem előtt, hogy az oldalunk ne csupán a saját albérletkeresési tapasztalatainkra épüljön, hanem egy olyan eszközként is szolgáljon, amely szélesebb körben hasznos lehet más, hasonló helyzetben lévő felhasználók számára is. Mivel mindhármunk számára fontos, hogy a projekt végén egy jól működő, megbízható platformot adjunk át, ezért folyamatos tesztelést végeztünk, hogy biztosítsuk a hibamentes működést.

Ebben a dokumentációban bemutatjuk az oldalunk fejlesztésének hátterét, a megvalósítás folyamatát, valamint azokat a szoftverfejlesztési döntéseket, amelyek meghatározzák a projektsikerét.

# Milyen keretrendszereket használunk?

Projekt fejlesztése során törekedünk arra, hogy minél kevesebb harmadik féltől származó könyvtárat használjunk, ezáltal pár Utils saját fejlesztésű, és saját szerveren futtatható, így teljes befolyásunk van a működésére.

## Szerveroldali

### MongoDB

A projektünk kezdetekor el kellett döntenünk, hogy az adatbázis kezeléshez használt programmal kapcsolatban maradunk-e a hagyományos MySQL-nél, amit az elmúlt években tanítottak nekünk vagy megpróbálunk valamit, ami nem szokványos, de a munkánk annál inkább felgyorsíthatja, könnyebbé teheti. Ekkor jutottunk a döntésre, hogy az eddig megszokott program helyett MongoDb-t használunk, ami jobban megfelel az igényeinknek. A MongoDB a MySQL-hez képest rugalmasabb adatkezelő, a programunk gyakran változó adat szerkezeteit egyszerűbben kezeli. A MongoDB használata lehetővé teszi számunkra, hogy gyorsabban és hatékonyabban fejlesszük alkalmazásunkat. Az adatmodell könnyen módosítható, ami csökkenti a fejlesztési időt. Ezenkívül a MongoDB skálázhatósága is kiemelkedő, mivel képes nagy mennyiségű adat kezelésére.

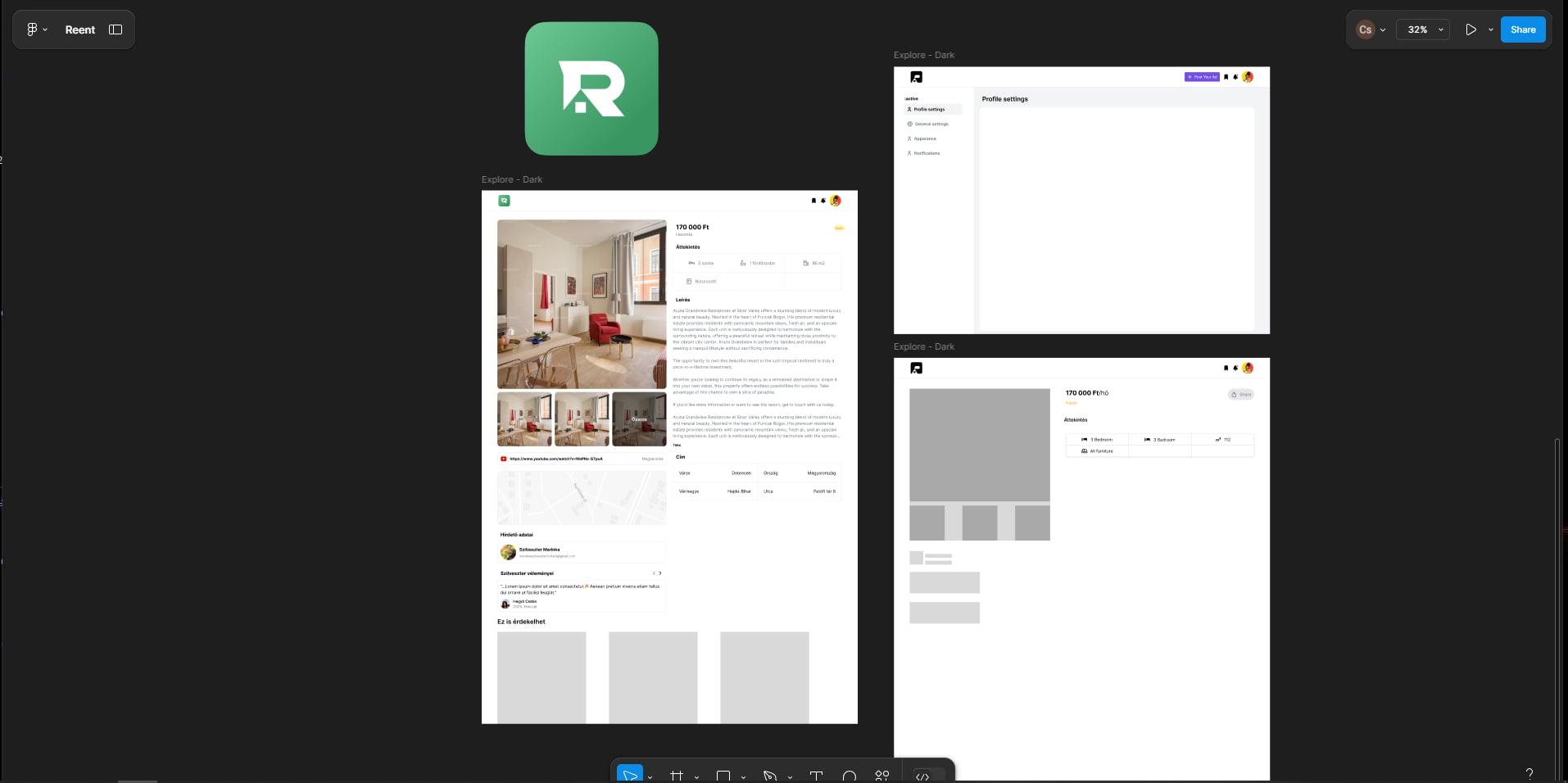
### Prisma

Az adatbázis kezeléshez Prisma ORM-et használunk a MongoDB-vel egyhuzamban. A prisma schema-kat létrehozva jelentősen felgyorsítja a munka folyamát sima sql kódhoz vagy akár más keretrendszerhez képest. Az eszköz automatikusan generál egy típusbiztos lekérdező klienst ami azt teszi lehetővé, hogy közvetlen az adatbázis objectumaira hivatkozhatunk programunkból, ez jelentősen felgyorsítja a fejlesztést és minimalizálja a potenciális hibákat. A különböző adatbázis kezelők támogatása is egy nagy előny hisz a sima MySQL-től eltérve a MongoDB-re való támogatás hatalmas megkönnyebbülés. A Prisma nagyban hozzájárult a zökkenőmentes kezeléséhez adatbázisunkban.

## Kliensoldali

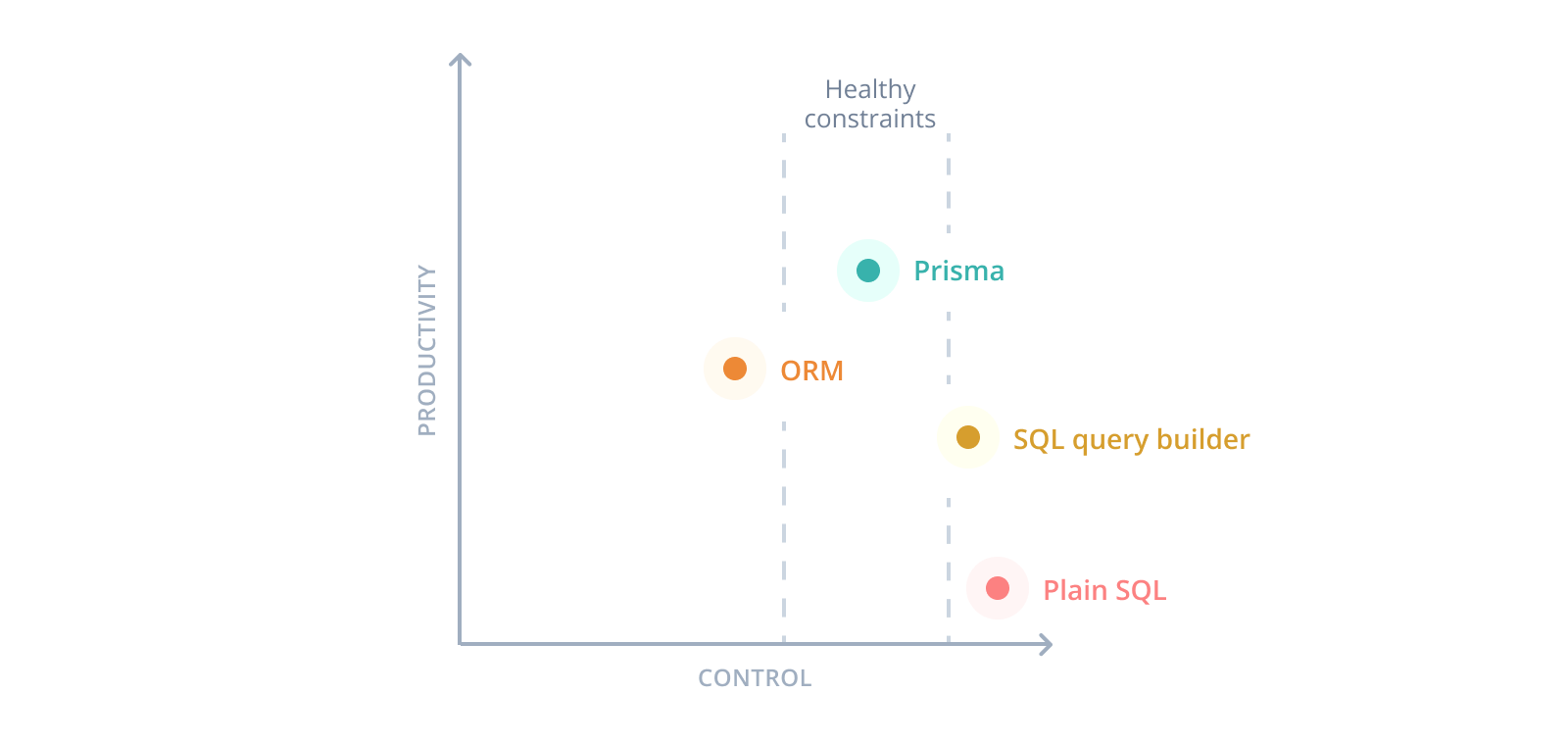
## Figma

Feladat elkezdése előtt nyilvánvaló lett számunkra, hogy a programozás egyik alap lépését, a tervezést nem hagyhatjuk ki. Erre nyújt megoldást a Figma, mint modern tervezőeszköz. A felhasználói felület, a Frontend egyik legnagyobb kihívása a Figma használatával kifejezetten egyszerűbb és élménydúsabb lett. A program egyik legnagyobb előnye, hogy a valós idejű együttműködést lehetővé teszi ezáltal a csapatunk tagjai egyszerre dolgozhatnak ugyanazon a terven, akárhol is helyezkedhettek el, bármilyen időben. A tervezési folyamat időigénye ennek hatására egekbe is képes nőni. Jóval megkönnyítette a kommunikációt és a közös munkát is.

A kommunikáció további fontos tényezője, hogy a Figma böngésző alapú, így külön telepítésre nincs szükség, bárhol bármilyen eszközről el lehet érni az oldalunk egyik legfontosabb területét, akár mobiltelefonról is. A dizájn integrálása más keretrendszerbe egy nagyon fontos funkció, hisz sokszor, több időt igénylő kód megvalósítása sose volt ilyen zökkenőmentes. A projectünk kinézete a modern webdizájnból nyer ihletet, szeretné replikázni és ennek megvalósításához elengedhetetlen eszköz volt a Figma.

### TailwindCSS

A frontend programozás egyik legnagyobb része a dizájn kifejlesztése és a stílusos megoldások megvalósítása. Erre a célra sok fajta megoldás létezik, de mi egy keretrendszer használatával fejlesztjük projectünket. A TailwindCSS egy modern, könnyen használható keretrendszer, ami előre definiált osztályokat biztosít a stílusok közvetlen alkalmazásához a HTML elemekre. Egy stílusos, felhasználóbarát weboldal ki alakításához elengedhetetlen egy megbízható keretrendszer alkalmazása, ami nem korlátoz előre meghatározott komponensekre így lehetővé teszi az egyedi és reszponzív felhasználói felület létrehozását. A keretrendszer integrálása projectünkbe zökkenőmentes, és gyors folyamat, ami lehetővé teszi, hogy a fontos dolgokra koncentráljunk.



## Módszerek, algoritmusok

Ez egy kis minta szöveg. Ez egy kis minta szöveg. Ez egy kis minta szöveg. Ez egy kis kis minta szöveg. Ez egy kis minta szöveg. Ez egy kis minta szöveg. Ez egy kis minta szöveg. Ez egy kis minta szöveg. Ez egy kis minta szöveg. E

z egy kis minta szöveg. Ez egy kis minta szöveg. Ez egy kis minta szöveg. Ez egy kis minta szöveg. Ez egy kis minta szöveg. Ez egy kis minta szöveg. Ez egy kis minta szöveg. Ez egy kis minta szöveg. Ez egy kis minta szöveg. Ez egy kis minta szöveg. Ez egy kis minta szöveg.

# A fejlesztett webhely/alkalmazás bemutatása

Ebben a részben kell a szakdolgozati munka során végzett fejlesztői munkája eredményét bemutatnia legalább 15 oldalon. Azt kell demonstrálnia, hogy az egyetemi tanulmányaira épülően a szakdolgozati munka során összegyűjtött új információkat hogyan ültette át a gyakorlatba, hogyan alkalmazta a megismert módszereket és eszközöket, és hogy milyen eredménnyel járt a munkája, milyen alkalmazás készült el, milyen új módszert dolgozott ki stb.

*Nem felhasználói kézikönyvet ír! Ne írjon le alap kódokat, teljesen egyértelmű, a témakör bevezető szintjéhez tartozó információkat!* (Például itt se írja le, hogyan töltötte be az alap html taggel a css-t, vagy hogy hogyan adta meg a weboldal nyelvét!)

Kizárólag a szakmai szempontból érdekes részeket emelje ki! Azokat, amiket bemutat tanárainak és jövendőbeli munkaadóinak, hogy még ezeket **is** tudja, képes az új információkat értelmezni, adoptálni, tudja alkalmazni!

Itt már ne írjon irodalmi hivatkozásokat, itt csak a saját munkájáról írjon! Érdemes visszahivatkozni az elméleti részben leírtakra, hogy bizonyos dolgokat hogyan adoptált, miket használt fel, esetleg hogy mely technológiák nem voltak alkalmasak, és hogy melyikre cserélte le és miért.

## Előkészítés, tervezés

## A megvalósításban jelentős részek

## Az elkészült webhely bemutatása funkcuonálisan

# Összegzés

Röviden, tömören és világosan ismertetni kell a fontosabb megállapításokat és következtetéseket. Bátran és egyértelműen közölni kell, ha valamely célkitűzést nem sikerült megvalósítani. (Megfelelő indoklás mellett ez egyáltalán nem csökkenti a diplomadolgozat értékét!)

Az összegzésnek 2-3 oldalnak kell lennie.

Ebben a fejezetben a szakdolgozó értékeli, hogy a kitűzött kezdeti szakdolgozati célok közül mit és milyen szinten sikerült elérnie. Kiemeli a problémásabb részeket, hangsúlyozza a megoldott feladatokat. Ezen túl összefoglalja a munka során felmerült, még megoldásra váró problémákat, illetve a témában rejlő lehetséges továbbfejlesztési, kiegészítési lehetőségeket.

Ennek a fejezetnek is legalább másfél oldalnak kell lennie.

# Irodalomjegyzék

1. Nagy József: A kicsi szöveg, *Egy könyv*, Kiadó, Debrecen, 2009, 25. oldal.
2. Egyiptomi írások magyar szemmel, Írástörténet sorozat, *Tudós Virtus* 2008. november 25. <http://tudos.virtus.hu/?id=detailed_article&aid=57927> (Letöltve 2018.02.05.)
3. Friedrich Klára : A Phaisztoszi Korongról másképpen , „Játszani is engedd...”, Rovásírás, 2007. szeptember <http://www.rovasirasforrai.hu/Rovasiras/Jatszani-is-engedd_Phaisztoszi-korong.htm> (Letöltve 2018.04.11.)
4. Leírás fejezete, Másik szöveg, *Másik könyv*, Kiadó, Debrecen, 2008, 25–34.
5. Encryption, Generates the NT-Hash from the given plaintext-password, *PEAR Manual*, PHP.net, 2015-12-08, <http://pear.php.net/manual/en/package.encryption.crypt-chap.crypt-chap-mschapv1.ntpasswordhash.php> (Letöltve 2017. november 6.)
6. Umberto Eco: Hogyan írjunk szakdolgozatot? Kairosz Kiadó, Győr, 1996, <http://www.kodaly-bp.sulinet.hu/pefele/E-k%C3%B6nyvek/Eco%20Hogyan%20%C3%ADrjunk%20szakdolgozatot.pdf>   
   (Letöltve 2018.01.11.)
7. Around the letter? In: A Disagreeably Facetious Type Glossary for the Amusement & Edification of People Beginning a Love Affair with Fonts. Microsoft Corporation, 1998, <https://www.microsoft.com/typography/news/glossary/ch5.htm> (Letöltve 2018.04.11.)

Az irodalomjegyzék tartalmaz minden olyan könyv-, periodika-, elektronikus média-hivatkozást, amelyeknek a felhasználásával készült a dolgozat.

Az irodalomjegyzék minimális hossza a témavezetővel való konzultáció során határozódik meg.

Minimum 15 tételesnek kell lennie az irodalomjegyzéknek, de a szükséges tételszám a témától függ. Elképzelhető, hogy a 30 lesz a minimum a választott témát és a talált szakirodalmi tételeket tekintve.

A javasolt tagolás az alábbi (megfelelő számú irodalom esetében célszerű magyar és külföldi részekre bontani):

Könyvek, cikkek, tanulmányok

Statisztikai adatforrások

Jogszabályok

Internetes adatgyűjtés  
(Itt azokat az oldalakat értjük, amelyek nem online tudományos cikk, könyv vagy tanulmány, hanem például W3C vagy a PHP.net egyik oldalán található ajánlás, standard stb. Ebben az esetben is meg kell adni a dokumentum logikai címét is a fizikai cím mellett, lásd fentebb a példában.)

# Függelék

A függelék nem kötelezően része a szakdolgozatnak/diplomamunkának.

Ide kerülnek:

* A nagyobb méretű táblázatok, ábrák.
* Ide helyezhető el a kérdőíves felmérés alapjául szolgáló dokumentumok, továbbá a statisztikai és matematikai számítások alaptáblái is.
* Egyes esetekben rövidebb szöveges dokumentumok (pl. szerződése, jogszabály részletek, stb.) is helyet kaphatnak itt.
* Ha sok a fénykép, képernyőkép vagy egyéb más, a probléma illusztrálásához, hatékonyabb bemutatásához, teljesebb leírásához tartozó rész, elhelyezésük itt javasolt.

A követelményben megadott, szükséges oldalszámhoz a függelék nem tartozik hozzá. Tehát ha a szakdolgozati követelményben minimális terjedelemként 30 oldal szerepel, a dolgozat eddigi, a függeléket megelőző, oldalszámmal ellátott részeinek kell legalább 30 oldalt kitennie.