BGSzC Logisztikai és Kereskedelmi Technikum és Szakképző Iskola

1087 Budapest, Szörény utca 2-4.

1. Záró dolgozat

Bolti nyilvántartó program

Konzulens tanár: Készítette:

Boros Sándor Hummel Vendel, Hunka Róbert Emánuel

Tartalom

[1 Bevezetés 3](#_Toc85723173)

[1.1 Feladat leírás 3](#_Toc85723174)

[1.2 A felhasznált ismeretek 3](#_Toc85723175)

[1.3 A felhasznált szoftverek 3](#_Toc85723176)

[2 Felhasználói dokumentáció 4](#_Toc85723177)

[2.1 A program általános specifikációja 4](#_Toc85723178)

[2.2 Rendszerkövetelmények 4](#_Toc85723179)

[2.2.1 Hardver követelmények 4](#_Toc85723180)

[2.2.2 Szoftver követelmények 4](#_Toc85723181)

[2.3 3. A program telepítése 4](#_Toc85723182)

[2.4 A program használatának a részletes leírása 5](#_Toc85723183)

[3 Fejlesztői dokumentáció 6](#_Toc85723184)

[3.1 Az alkalmazott fejlesztői eszközök 6](#_Toc85723185)

[3.2 Adatmodell leírása 6](#_Toc85723186)

[3.3 Részletes feladatspecifikáció, algoritmusok 6](#_Toc85723187)

[3.4 Tesztelési dokumentáció 7](#_Toc85723188)

[4 Összefoglalás 8](#_Toc85723189)

[4.1 Önértékelés 8](#_Toc85723190)

[4.2 Továbbfejlesztési lehetőségek 8](#_Toc85723191)

[5 Felhasznált irodalom 9](#_Toc85723192)

[6 Ábrajegyzék 10](#_Toc85723193)

# Bevezetés

## Feladat leírás

A felhasználói dokumentáció célja, hogy segítséget adjon a leendő felhasználónak a programmal telepítésével és használatával kapcsolatos minden probléma megoldásában, segítse a program kezelésének az elsajátítását.

1. **Fájlok előkészitese**

* **Látogasson el a “**[**https://github.com/zenerra/vizsgaremek**](https://github.com/zenerra/vizsgaremek)**” linken keresztül elérhető repositoryhoz**
* **A zöld ”Code” nevezetű gombot kattintva válassza ki a “Download ZIP” feliratot és töltse le a fájlt a számitógépére majd bontsa ki.**
* Importálja a db\_nyilvantartas.sql nevű fájlt a XAMPP-ba

1. **Adatbázis és Szerver telepítése**

* Futtassa le az alábbi parancsokat egy parancsor ablakban.

1 mkdir vizsgaremek

2 cd vizsgaremek

3 mkdir backend

4 cd backend

* Másolja be az alábbi fájlokat a backend mappából.

│ app.js

│ db.js

│ db\_nyilvantartas.sql

│ package-lock.json

│ package.json

└─── routes

alkalmazott.js

beszallito.js

cim.js

szamla.js

termek.js

tetel.js

Hozzon létre egy .env fájlt. Az alábbi minta alapján, tegye a “backend” nevezetű mappába. Győződjön meg arról, hogy az adatbázis kacsolathoz megfelelő adatokat tartalmazzon.

1 DB\_PASSWORD=

2 DB\_PORT=3306

3 DB\_NAME=db\_nyilvantartas

4 DB\_USER=root

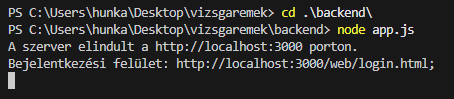
5 DB\_HOST=localhost

Futtassa le a következő parancsokat a backend mappa termináljában a szerver futtatásához.

1 npm install

2 node app.js

Sikeres esetben elindult a szerver, ami az alábbi példában a “<http://localhost:3000>”-en található. A webes felületet pedig a “<http://localhost:3000/web/login.html>” URL-en érhet el.

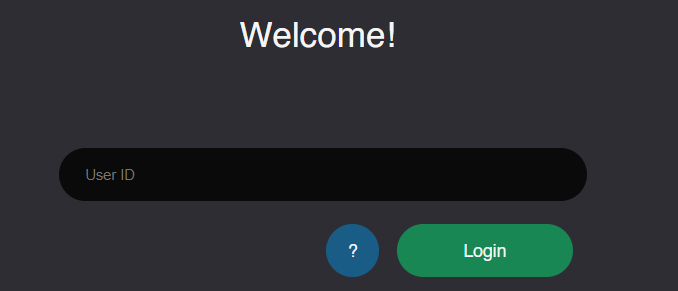


1. kép Terminál tartalma sikeres telepites esetén

1. Asztali felülethez való hozzáférés

VENDEL

1. Bejelentkező felület webes és asztali felületen



2. kép Webes bejelentkező felület

Mindkét felület eloszor a dolgozó azonositojat fogja kerni a funkciok hozzaferesehez felteve hogy az adott dolgozonak van joga a felulet belepesere, az azonositot kizárólag az adatbazisbol nyerheti ki. Egyik azonosito, ami mindegyik feluletre belepest biztosit, az “1”.

## A felhasznált ismeretek

Nagyban épitkeztünk a szakmai tanáraink tudására. Tóth Sándor oktató adta át az asztali applikációhoz való tudását, ezzel megalapozva a WinFormos képességégeinket. Másik szakmai tanárunk Boros Sándor, aki sokoldalúan támogatott minket a projekt során a tudásával és tapasztalataival. Elsőként a backendes programozásban tudok rengeteg újat tanitani nekunk. Problema vagy kerdes eseten, rá is tudtunk hagyatkozni

A projekt során használtunk API-t is, pontosabban a Chart.js JavaScript alapú statisztikak megjelenitesere alkalmas konyvtarat. Ehhey a hivatalos weboldalukon keresztul talalt dokumentaciok alapjan integraltuk.

Szamos esetben kisebb nagyobb hibakodokkal talalkoztunk, amik megoldasahoz gyakran hasznaltuk a Stack Overflow-ot, ami egy nyilvanos megoldaskereso oldal programozoknak és a Youtube nevezetu videomegosztot.

Altalanos frontendbeli tudasunkat az agazati alapvizsga alapozta meg, ezaltal konnyuszerrel epitettuk fel a weboldal kulsejet.

## A felhasznált szoftverek

Office360-at, azon belül Microsoft Word, Teams és Power Point, valamint a Windows operációs rendszert a fejlesztés során.

Ezen kivul szamtalan mas szoftvert alkalmazott a projekt, errol a “3.1 Az alkalmazott fejlesztői eszközök” cimsor alatt van kifejtve.

# Felhasználói dokumentáció

## A program általános specifikációja

E bolti nyilvántartó szoftver egy modern, integrált rendszer, amely hatékonyan segíti a készletek és tranzakciók kezelését bármelyik vegyeskereskedelmi kisboltnak. A szoftver két fő modulból áll: egy webes felületből és egy asztali applikációból. A webes felület lehetővé teszi a beérkező áruk és a raktárkészlet nyomon követését, a termékek adatainak szerkesztését, valamint részletes statisztikák és kimutatások készítését. A bolti alkalmazottak itt láthatják az eladások összegét, a készletmozgásokat. Az asztali applikáció elsősorban a vásárlói tranzakciók feljegyzésére és nyugtázására szolgál.

Vásárlásokat és értékesítéseket lehet rögzíteni egy egyszerű, intuitív felületen, amely lehetővé teszi a nyugták automatikus formázását. Az alkalmazás offline módban is működik, a raktári mozgásokat pedig valós időben szinkronizálja a webes felülettel, amikor elérhető az internetkapcsolat. A rendszer többfelhasználós működést támogat, különböző jogosultsági szintekkel, és biztosítja a tranzakciók, valamint az adatmódosítások naplózását. A szoftver rugalmasan testreszabható, felhasználóbarát kialakítása pedig megkönnyíti a mindennapi munkavégzést a boltvezetők és az alkalmazottak számára.

Ez a rész a program fontosabb jellemzőit és funkcióit tartalmazza. A cél, hogy a leendő felhasználó ezt a fejezetet elolvasva el tudja dönteni, hogy a program megfelelő-e a számára.

Ajánlott terjedelem: 0,5 oldal.

## Rendszerkövetelmények

### Minimális Hardver követelmények

A szerver futtatja a Node.js alapú backend alkalmazást (app.js), kezeli a HTTP kéréseket, és kapcsolódik a MySQL adatbázishoz. Az alábbi minimális konfiguráció biztosítja, hogy a rendszer alapvető funkcionalitása (tranzakciók lekérdezése, frissítése, az asztali alkalmazás, statikus fájlok kiszolgálása) működjön.

* **Processzor**: Kétmagos CPU 2,0 GHz órajellel (például Intel Core i3 vagy azzal egyenértékű AMD processzor).
* **RAM**: 4 GB (a Node.js, a MySQL és az alapvető kéréskezelés futtatásához).
* **Tárhely**: 10 GB szabad lemezterület (a Node.js, az asztali app, a MySQL és az adatbázis tárolásához).

**További megjegyzések**

• Adatbázis méret: A MySQL adatbázis mérete a tranzakciók (szamla tábla) és tételek (tetel tábla) számával arányosan nő. A szerveroldali tárhelyet ennek megfelelően kell méretezni (például 1 GB 100 000 tranzakcióra, az adatok méretétől függően).

• Tárhelyhiány esetén: Ha nincs több hely az adatbázis tárolására, a következő lépéseket teheti:

* + Régi adatok archiválása: Az inaktív vagy kevésbé fontos tranzakciókat exportálja egy külső fájlba (például SQL dump vagy CSV), majd törölje azokat az adatbázisból.
  + Tárhely bővítése: Telepítsen nagyobb kapacitású merevlemezt vagy SSD-t a szerver tárolására.
  + Adatbázis optimalizálása: Tisztítsa meg az adatbázist az elavult vagy felesleges rekordoktól, és használjon indexeket a tárolási igények csökkentésére. A ”DELETE FROM szamla WHERE skiallitas < DATE\_SUB(CURDATE(), INTERVAL 15 YEAR);“ utasitassal torolheti a 15 evnel regebbi tranzakciokat az adatbazisbol

Ebben a részben kell leírni a minimális és ajánlott hardver konfigurációt, amely a program futtatásához szükséges. Pontos paramétereket kell megadni, még akkor is, ha a program amúgy minden gépen lefut.

### Szoftver követelmények

**Operációs rendszerek**:

* A program szerveroldali komponensei a következő operációs rendszereken futnak:
  + Windows 10 / 11 (64 bites)
  + Ubuntu 18.04 LTS (vagy újabb, asztali program nem futtatható)
  + macOS 11 (vagy újabb, asztali program nem futtatható)
* A kliensoldali komponensek (webfelület) bármely, modern webböngészőt támogató operációs rendszeren elérhetők:
  + Windows 10 / 11 (64 bites)
  + macOS 10.15 (vagy újabb)
  + Ubuntu 18.04 LTS (vagy újabb)
  + Android 9.0 (vagy újabb)
  + iOS 14 vagy újabb (mobil eszközökhöz)

Ha mobil eszközén kivanja megtekinteni, csatlakoztassa a telefont ugyanahhoz a Wi-Fi hálózathoz, és használja a szerver helyi IP-címét (pl. http://192.168.1.100:3000). Az IP-címet a szerver gépen a ipconfig (Windows) vagy ifconfig / ip addr (Ubuntu/macOS) paranccsal tudja lekérdezni.

**Webböngésző**:

* Bármely naprakész böngésző, peldaul:
  + Google Chrome
  + Mozilla Firefox
  + Microsoft Edge
  + Safari

Az alkalmazás modern JavaScript funkciókat (például fetch API, async/await) használ. Régebbi böngészők esetén kompatibilitási problémák léphetnek fel, ezért a legfrissebb böngészőverzió használata erősen ajánlott.

**Hálózat:**

* A fájlok előkészítése miatt indokolt stabil internetkapcsolat a szoftverkomponensek (pl. Node.js, MySQL) letöltéséhez és a Github oldala betöltéséhez.

További megjegyzesek:

Le kell írni, hogy mely operációs rendszere(ke)n fut a program, és milyen egyéb szoftver komponensek szükségesek a működéshez (pl. .NET, DirectX, esetleg adatbázis-szerver és adatbázis-állományok, stb.) A beadott CD-n ezeknek is ott kell lenniük.

Ajánlott terjedelem: ½ -1 oldal, felsorolásszerűen leírva

## 3. A program telepítése

Képekkel illusztrált, részletes leírás a program telepítésének a menetéről.

* A leírás alapján a felhasználónak hiba nélkül telepíteni kell tudni a programot. A leírásnak ki kell térnie a telepítés során kiválasztható opciókra is.
* Ha esetleg nincs telepítőprogram, akkor kellő részletességgel le kell írni, hogy mely fájlokat, pontosan hova kell felmásolni, és hogy lehet a programot futtatni. Módosító ablak(1. kép)
* 

1. kép Ez a módosító ablak a programban

Ajánlott terjedelem: 2 -4 oldal, ábrákkal együtt.

## A program használatának a részletes leírása

Mindenre kiterjedő, részletes leírás a program használatáról. Alapszabályok:

* Amit leprogramoztál, azt a dokumentációban írd is le, ne legyenek eltitkolt funkciók.
* Minden pontosan, „szájbarágósan” legyen leírva. A dokumentáció alapján a teljesen kezdő, vagy laikus felhasználóknak is használniuk kell tudni a programot.
* A stílus legyen pontos és közérthető, vedd figyelembe, hogy a felhasználói dokumentáció nem szakembereknek készül.
* Ugyanakkor kerüld a laza stílust: rövidítések, smilie-k, szleng kizárva.
* Alkalmazz ábrákat, screenshot-okat , de a ne legyen túlzott a képek aránya a szöveghez képest. Kb. 2-3 oldalanként egy ábra megfelelő.

Ajánlott terjedelem: 10-15 oldal, ábrákkal együtt.

# Fejlesztői dokumentáció

A fejlesztői dokumentáció célja, hogy a segítse program logikájának, illetve a program kódjának a megértését, illetve a program továbbfejlesztését. Szakemberek számára készül, elvárás tehát a pontosság és a szakmai jellegű stílus.

## Az alkalmazott fejlesztői eszközök

Fel kell sorolnod az összes olyan szoftver eszközt amelyet a program fejlesztéséhez, illetve a dokumentáció készítéséhez felhasználtál. (Pl. programozási nyelv, fejlesztői környezet, adatbázis-kezelő rendszer, kép-, szöveg-, zeneszerkesztő program stb.)

* Ha használtál mások által kifejlesztett modulokat, akkor azt is le kell írnod, a forrás pontos megjelölésével.
* Tájékozódj ezeknek a moduloknak a jogszerű felhasználásáról!

Ajánlott terjedelem: ½ -1 oldal.

Programozási és leiró nyelvek:

* C# (a WinForms alkalmazás fejlesztéséhez)
* JavaScript (a kliensoldali logika és dinamikus elemek implementálásához, pl. transactions.js)
* HTML5 (a webes felület struktúrájához, pl. transactions.html)
* CSS (a webes felület dizájnjához és responszív megjelenítéséhez)
* Markdown (a dokumentáció részeként, például jegyzetekhez vagy formázott szövegekhez)

• Fejlesztési környezetek:

* Microsoft Visual Studio (a C# és WinForms kódolásához)
* Microsoft Visual Studio Code (a JavaScript, HTML és CSS szerkesztéséhez, valamint a Node.js projektekhez)
* Microsoft Word (a dokumentáció véglegesítéséhez és formázásához)

Adatbázis-kezelő rendszerek:

* MySQL (az adatbázis kezelésére, pl. szamla és tetel táblákhoz)

Szerveroldali technológiák:

* Node.js (a backend futtatásához, pl. app.js)

felhasznált moduljai:

* + Express.js (keretrendszer a szerveroldali útvonalakhoz)
  + MySQL2 (a MySQL-lel való kommunikációhoz)
  + npm (a Node.js csomagkezeléséhez, projektfüggőségek kezelésére)
  + Body-parser (HTTP kérések adatfeldolgozásához (részben felváltotta az express.json())
  + Path (fájlrendszer elérési utak kezelésére a szerveren)
  + URL (fileURLToPath, fájl-URL-ek konvertálására helyi elérési utakra, az url modul részeként)
* Dotenv (A szerverkonfiguráció és érzékeny adatok titkosítására.)

Verziókezelés:

* Git (a forráskód verziókezelésére és a projekt változásainak nyomon követésére)
* GitHub (Távoli tárolás és csapatmunka közremukodesere)

Projektmenedzsment és kommunikációs eszközök:

* Trello (a feladatok szervezésére és követésére)
* Discord (a csapat hangalapú kommunikációjához)
* Microsoft Teams (a gyakran használt jegyzetek, képanyagok tárolására)

Dizájn és grafikai eszközök:

* Figma (az oldal elrendezésének, grafikai elemek és a dizájnnyelv megtervezéséhez)
* Webes erőforrások (betűtípusok és színpaletta kiválasztásához, pl. Dafontfree és Coolors)

Külső API-k és könyvtárak:

* Chart.js (diagramok és statisztikák megjelenítéséhez a webes felületen)

Hardver és operációs rendszer:

Windows alapú számítógép (a fejlesztés elsődleges környezeteként)

## Adatmodell leírása

A feladatban alkalmazott adatbázis vagy adatszerkezet részletes leírása

* Adatbázis esetén az adattáblák leírása, a közöttük lévő kapcsolatok megadása, lehetőleg diagram is legyen
* Ha nincs adatbázis, akkor a program fő adatszerkezeteinek a specifikációja, célszerű diagramot is rajzolni
* OOP jellegű megvalósítás esetében az UML osztálydiagram

Ajánlott terjedelem: a feladat jellegétől függően 2-5 oldal.

## Részletes feladatspecifikáció, algoritmusok

A program lényeges függvényeinek, az osztályok metódusainak a specifikációja (mit valósít meg az adott függvény, illetve metódus, milyen paraméterei vannak, mi a visszatérési érték)  Az algoritmizálható részek leírása valamilyen algoritmus-leíró eszközzel (struktogram, pszeudo-kód, esetleg UML aktivitás-diagram) Ajánlott terjedelem: a feladat jellegétől függően 2-5 oldal. 5. Forráskód  A teljes forráskódot a nyomtatott dokumentációba nem kell beletenni!  Lehet viszont a nyomtatott dokumentációban a fontosabb kódrészeket magyarázattal szerepeltetni

## Tesztelési dokumentáció

1. Legalább 3 különböző teszteset részletes bemutatása.

* különböző felhasználó tevékenységek esetén hogyan reagált a program
* milyen üzeneteket kaptunk
* mi a teendő az egyes üzenetek esetében

1. Normál teszteset, extrém teszteset (bolondbiztosság tesztelése)
2. A tesztelés során kiderült hibák felsorolása A tesztelési dokumentációból derüljön ki, hogy ismered a különböző tesztelési módszereket (pl. fekete doboz, fehér doboz módszer)

Ajánlott terjedelem: a feladat jellegétől függően 2-5 oldal.

# Összefoglalás

## Önértékelés

Hummel Vendel:

A kitűzött célok elérése, a felmerült problémák és megoldásuk felsorolása.

A saját fejlődés bemutatása (mit tanult meg, hogyan alkalmazta…)

Ajánlott terjedelem: 0,5-1 oldal

Hunka Róbert Emánuel:

- Probléma (1):

1. A kliens oldalon egy form elküldése utan az input mezők kattintásra nem reagáltak, mert egy üzenetdoboz (message box) hibája blokkolta őket.
2. Ok és hogyan történt : A probléma abból adódott, hogy a Node.js nem támogatja megfelelően ezt a funkciót, emiatt az üzenetdoboz zavarokat idézett elő a kliensoldali JavaScriptben.
3. Megoldás: Ido es egyertelmu megoldas hianyaban vegul nem használtam üzenetdobozokat, helyette egy értesítési rendszert építettem fel, ami megoldotta a problémát.
4. - Probléma (2):
5. Amikor megpróbáltam frissíteni egy adatot a szamla táblában, kaptam egy hibát, ami miatt nem ment a művelet.
6. Ok és hogy történt : A hiba onnan eredt, hogy MySQL-ben a tetel tábla sazon mezője idegen kulcsként (foreign keys) hivatkozott a szamla táblára, és a frissítés megsértette ezt a korlátozást. Olyan rekordot próbáltam módosítani, amit a tetel tábla még használt.
7. Megoldás: Először ellenőriztem a kapcsolódó rekordokat a tetel táblában, és frissítettem őket, hogy ne legyen korlátozási hiba. Utána már mukodott a szamla tábla frissítése.
8. - Miért történt : A MySQL-ben az idegen kulcsok (foreign keys) biztosítják az adatbázis integritását. Ha egy szamla rekordot próbáltál módosítani vagy törölni, de a tetel tábla még hivatkozott rá, a MySQL hibát dobott, mert nem engedélyez változásokat, amelyek megsértenék a kapcsolatot.
9. - Céljaim:  
   Célom volt egy olyan UI-t épiteni, amit szép, barátságos és közérthető használni egy átlagos embernek. Emiatt animációk mellett Figmát (egy kollaboratív felülettervező szoftver) is használtam a letisztult kinézet elérése érdekében. Igyekeztem olyan programit késziteni, ami akár valóban segitsegul szolgalna peldaul egy kiskereskedő vallalkozo számára. Ugy gondolom, ezek bizonyos mertekben teljesultek is, amire buszke vagyok

A kitűzött célok elérése, a felmerült problémák és megoldásuk felsorolása.

1. A projekt során elsajátítottam az adatbázis és a backend közötti kapcsolat kialakítását, amely korábban nehézséget okozott számomra. A termékek CRUD felépítésének sikeres kidolgozása jelentős előnyt jelentett, mivel lehetővé tette, hogy a tranzakciók szerkesztését ugyanilyen hatékonyan valósítsam meg.

A saját fejlődés bemutatása (mit tanult meg, hogyan alkalmazta…)

Ajánlott terjedelem: 0,5-1 oldal

## Továbbfejlesztési lehetőségek

Hummel Vendel:

*Jo lenne egyikonk megemlitene hogy jo lenne ha lehetne valtoztatni a program nyelvén, csak en mar tul sokat irtam. De radbizom igazabol.*

* Olyan ötletek, amelyeket meg akartál valósítani, de nem sikerült, vagy nem fért bele az időbe
* Olyan ötletek, amelyeket még érdemes a jövőben megvalósítani

Ajánlott terjedelem: 0,5-1 oldal

Hunka Róbert Emánuel:

1: Szerettem volna lehetőséget biztosítani a statisztikák fájlba történő exportálására, hogy a szerveren kívül is használhatóak legyenek, de időhiány miatt ez nem valósult meg.

2. Eleinte tervben volt egy admin felület is, amiben testreszabhatók az egyes alkalmazottak jogosultságai, adatai (például fizetés), és erősebb biztonságot biztosít a bejelentkezési oldalon, de ezt az iskolai évek alatt nem tanultuk meg alaposan, így nem került rá sor.

3. A tranzakciók nem teljesen szerkeszthetők a webes felületen, mert a termékek CRUD fejlesztése sok időt vett igénybe, ezért egy kisebb, kompakt verziót készítettem a tranzakciók szerkesztésére a teljes körű szerkeszthetőség helyett, hiszen a termek tablahoz hasonloan rengeteg entitást érintett egyszerre.

* Olyan ötletek, amelyeket meg akartál valósítani, de nem sikerült, vagy nem fért bele az időbe

4: A projekt vegso szakasza fele rajottem, hogy a termékeknek rendelkezniük kellene egy "elérhető-e" tulajdonsággal, amely meghatározza, hogy egy termék elérhető-e az üzletben, vagy csak az adatbázisban szerepel, például limitált kiadású termék részeként.

A probléma az, hogy nem tudom módosítani a termékek értékeit (peldaul learazni), mert az befolyásolná a korábbi tranzakciók összegét, ami jelentős hibát okoz például a statisztikák megjelenitesekor.

Megoldás az lenne, ha külön entitást hoznánk létre a múltbéli értékek tárolására, így a tranzakciók mentése zavartalanul történhetne, anélkül hogy a termékárak vagy más hasonló változázók problémát okoznának. Igy az egyetlen atmeneti megoldas az, ha letrehoz a felhaszanlo egy uj termeket, hasonlo nevvel, eltero árazással, a réginek pedig a keszletegységét 0-ra csokkenti. Ezzel megakadalyozható a statisztikak hamis adatait és az asztali feluleten levo elerheto termekek duplikalt megjeleniteset. Úgy gondolom, ezen a jövőben érdemes lenne javitani a társammal.

* Olyan ötletek, amelyeket még érdemes a jövőben megvalósítani

Ajánlott terjedelem: 0,5-1 oldal

# Felhasznált irodalom

Minden olyan forrás pontos megadása, amelyet a szakdolgozatodban felhasználtál.

A forrás lehet pl.

Könyv. Meg kell adnod a következőket: szerző(k), cím, kiadó, kiadás éve.

Weboldal. Meg kell adnod a linket, az oldal címét Mikor láttad utoljára.

Elektronikus dokumentum. Meg kell adnod a szerzőt, a letöltés helyét, idejét

Ha a szakdolgozatban valamely forrásból szó szerint idézel, akkor a megfelelő szövegrészt idézőjelbe kell tenni, és lábjegyzetben meg kell jelölnöd az idézet forrását.

Ajánlott terjedelem: ½ -1 oldal.

Felhasznalt weboldalak:

* W3Schools (2025.05.02) - <https://www.w3schools.com>
* Boros Sandor GitHub repository (2025.04.22) - <https://github.com/borossandor27/orai_anyagok>
* CSS Unit Converter (2025.04.25) - <https://cssunitconverter.vercel.app/rem-to-em>
* ⓘ Character (2025.04.16) - https:// [www.compart.com/en/unicode/U+24D8](http://www.compart.com/en/unicode/U+24D8)
* Helvetica Font (2024.11.17) - <https://font.download/font/helvetica-255>
* Chart.js API (2025.03.27) - <https://www.chartjs.org/docs/latest/charts/bar.html>
* MySQL Node.js Express videó Sam Meech-Wardtól (2024.12.02) - <https://youtu.be/Hej48pi_lOc?si=Vav7bRaEriQxE7SA>
* Chat GPT adatok automatikus feltoltesere (2024.11.28) - <https://chatgpt.com/>
* Colors (2024.11.17) - <https://coolors.co>
* Munkavallaloi idezet Charles Dickenstől (2025.05.02) - <https://www.vantagecircle.com/en/blog/employee-motivational-quotes/>
* Stack Overflow - (2025.11.18) <https://stackoverflow.com/>

# Ábrajegyzék

[1. kép Ez a módosító ablak a programban 5](#_Toc63250394)