

Vizsgaremek címe

Képző intézmény

Postai cím

Vizsgaremek

A programom címe

Készítették:

1. név1
2. név2
3. név3

Tartalom

1	Bevezetés	1
1.1	Feladat leírás	1
1.2	A felhasznált ismeretek	1
1.3	A felhasznált szoftverek.....	1
2	Felhasználói dokumentáció.....	16
2.1	A program általános specifikációja.....	16
2.2	Rendszerkövetelmények	16
2.2.1	Hardver követelmények.....	16
2.2.2	Szoftver követelmények	16
2.3	3. A program telepítése.....	16
2.4	A program használatának a részletes leírása	17
3	Fejlesztői dokumentáció	6
3.1	Az alkalmazott fejlesztői eszközök	11
3.2	Adatmodell leírása.....	6
3.3	Részletes feladat-specifikáció, algoritmusok.....	11
3.4	Tesztelési dokumentáció	11
4	Összefoglalás.....	16
4.1	Önértékelés	19
4.2	Továbbfejlesztési lehetőségek.....	19
5	Felhasznált irodalom	20
6	Ábrajegyzék.....	22

1 Bevezetés

1.1 Feladat leírás

Írd le röviden, milyen problémára, milyen célra készítettél a programot. Milyen feladatok megoldásában kiknek nyújt segítséget a használata.

1.2 A felhasznált ismeretek

Milyen tantárgyakból, témakörökből hasznosítottál tudást? (pl. OOP, SQL, HTML/CSS, stb.)

1.3 A felhasznált szoftverek

Milyen programokat, fejlesztőeszközöket használtál? (pl. VS Code, MySQL, Figma, stb.)

2 Követelménymátrix

ID	Követelmény	Prioritás
K1	Bejelentkezés felhasználónév/jelszóval	Magas

3 Projektmenedzsment

3.1 Projekttervezés

3.1.1 Célkitűzések és határidők

A projekt főbb mérföldköveit és határidejüket egy Gantt-diagram segítségével ábrázoltuk.

- **Feladatok felbontása:**
 - Frontend fejlesztés (3 hét)
 - Backend API kialakítás (2 hét)
 - Tesztelés és hibajavítás (1 hét)

Gantt-diagram a projekt idővonaláról

4 Csapatmunka és verziókövetés

4.1 Feladatmegosztás

- *név1*: Frontend (React), UI tervezés
- *név2*: Backend (C#), adatbázis integráció
- *név3*: Tesztelés, dokumentáció

4.2 Verziókövetés

A kódot GitHubon kezeltük, feature branch-ekkel és pull request-ekkel.

- Repository: github.com/projektnev
- Branch stratégia: GitFlow

5 Kockázatelemzés

6 2.1 Kockázatok és enyhítésük

Kockázat	Hatás	Megoldás
Adatvesztés	Felhasználói adatok törlődnek	Napi mentés MySQL dump-pal
API válaszidő túl magas	Felhasználói élmény romlik	Gyorsítótár bevezetése (Redis)

7 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.

Ajánlott terjedelem: a feladat jellegétől függően 5-10 oldal.

7.1 Részletes feladatspecifikáció

A rendszer célja, hogy a felhasználók számára egyszerű és biztonságos módot biztosítson a [pl. termékek rendelésére, események nyilvántartására, stb.]. A funkciók között szerepel:

- Regisztráció, bejelentkezés
- Adatmódosítás
- Admin jogosultság esetén: adatok hozzáadása, törlése
- Vásárlási folyamat kezelése (pl. kosár, fizetés)
- Statisztikák, keresés, szűrés

7.2 Entitások leírása

Az alkalmazásunk szereplőit itt kell megjelölni. Ezeket az entitásoknak az előfordulásait tulajdonságaik segítségével határozhatjuk meg egyértelműen. Például, ha valamilyen értékesítést választunk, akkor ott szerepelnie kell terméknek, vásárlónak és vásárlásnak

7.2.1 Termék entitás

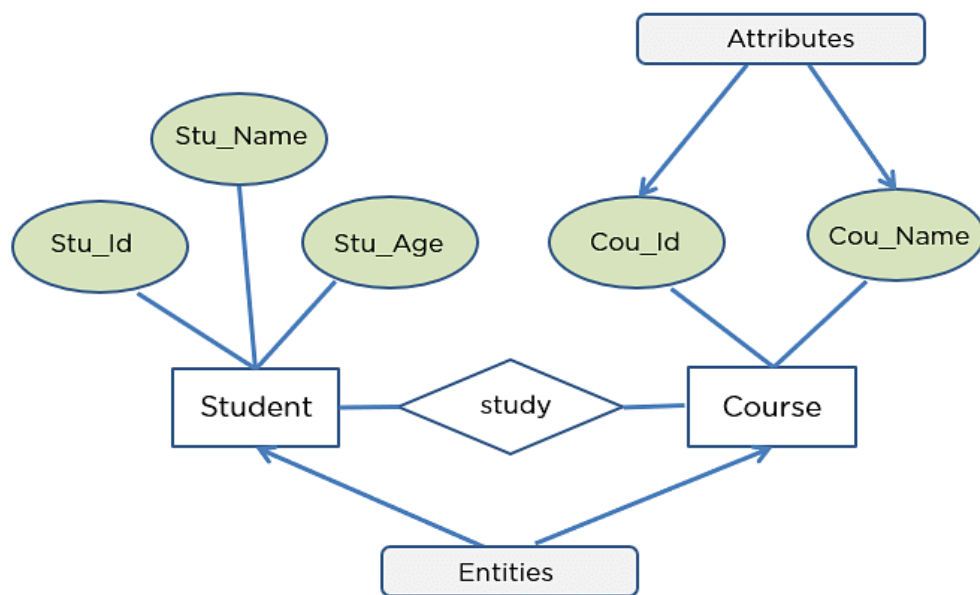
Tulajdonság	Alkalmazáson belül használt azonosító	Meghatározás
név	termek_megnevezes	A termék neve, amely alapján a vásárló beazonosíthatja
ár	termek_eladasi_ara	amelyet a vásárlónak ki kell fizetnie vásárlás esetén
cikkszám	termek_cikkszam	A gyorsabb számítógépes feldolgozás miatt szükséges. Általában a termék gyártási kódja
cikk azonosító	termek_id	Többféle termék kód lehetséges, így sokkal gyorsabb és biztonságosabb a feldolgozás, ha a rendszeren belül kap egyedi kódot

7.2.2 Vevő entitás

Tulajdonság	Alkalmazáson belül használt azonosító	Meghatározás
hivatalos név	vevo_neve	A kommunikációban ennek segítségével tudjuk megszólítani
belépési név	vevo_belepesi_neve	A rendszer használatához kapott név
jelszó	vevo_jelszo	A belépési jogosultság ellenőrzéséhez
email cím	vevo_email	A kapcsolattartás céljára
vevő azonosító	vevo_id	Biztosítja, hogy a vevő egyéb tulajdonságait tudja változtatni

7.2.3 Kapcsolat az entitások között

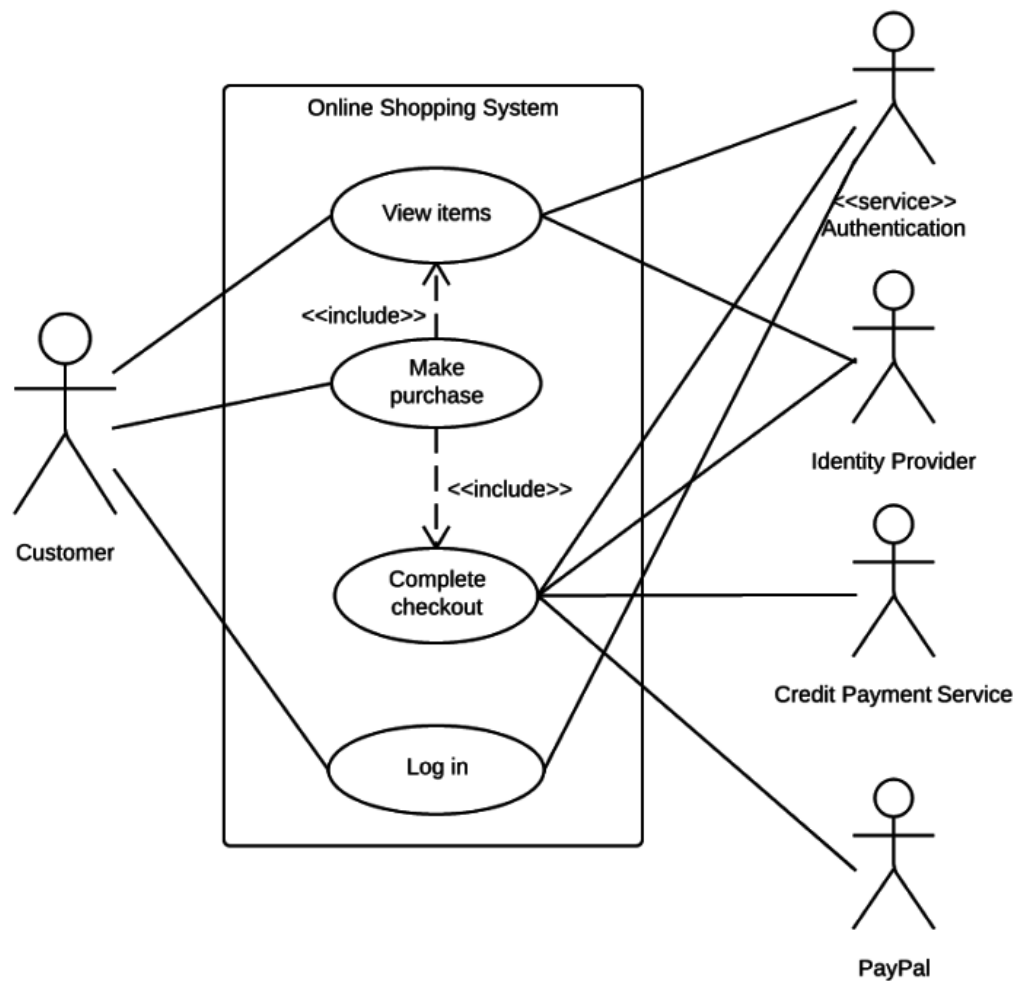
Szemléltetésének legjobb módja az ER diagram



1. ábra ER diagram

7.3 Használati esetek (use-case)

A letárolt adatok mindig egy pillanatnyi állapotot tárolnak. Alkalmazást azonban azért hozunk létre, hogy bizonyos műveletek során az adatok változásait követni tudjuk a segítségével. Ebben a fejezetben össze kell szedni az összes műveletet, amelyet támogatnia kell az alkalmazásunknak.



2. ábra Online vásárlás use-case diagramja

7.3.1 Vendég

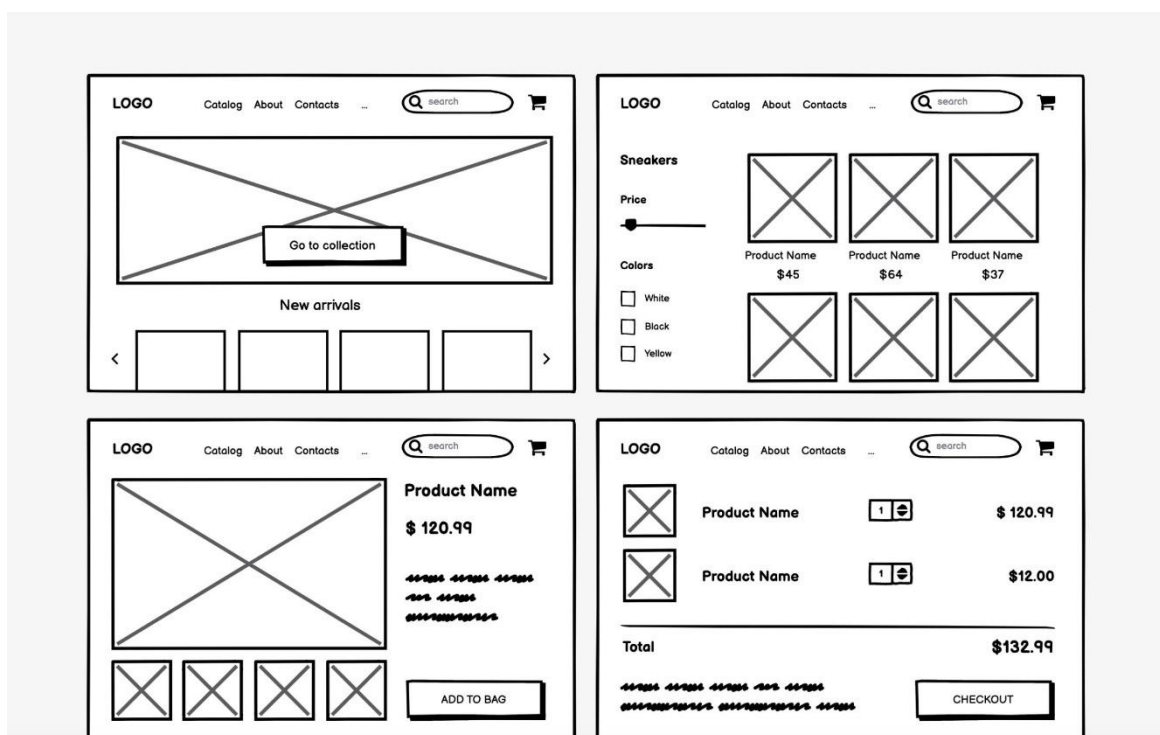
1. **login** művelet eredménye, hogy a vendég vevő entitássá alakul
2. **regisztráció** művelet eredménye, hogy a vendég vevő entitást hoz létre

7.3.2 Vevő

7.3.3 Termék

7.4 Nézetek

Nem valószínű, hogy egy megjelenítési felület elegendő az alkalmazáshoz, így meg kell terveznünk az adott feladat elvégzésére legjobban megfelelő tartalmat és elrendezést. Valószínűleg nem a legmegfelelőbb kezdésnél betűtípusokkal, színekkel és grafikákkal részletesen teletűzdelni ezeket a terveket. Régóta használjuk erre a célra a "képernyő drótváz"-at (angolul "wireframe"), amely egy vázlatszerű, sematikus ábra. Célja mindössze annyi, hogy a készülő weboldal vagy alkalmazás struktúráját és funkcionalitását bemutassa. Nem tartalmaz részletes grafikákat, színeket vagy képeket, hanem egyszerű dobozokat, vonalakat és szövegeket használ, hogy a tartalom helyzetét és a navigációt szemléltesse.



3. ábra képernyő drótvázak

7.5 API végpontok meghatározása

A nézeteket elemezve meghatározhatjuk az adatfeldolgozási igényüket. Mivel követelmény a vizsgaremekkel szemben a REST API-k (*Representational State Transfer Application Programming Interface*) készítése és használata, addig nem szabad elkezdni a kódolást, amíg nem készítjük el az egyes nézeteket kiszolgáló végpontokat.

7.5.1 Végpontok listája

Metódus	URL	Leírás	Request példa
POST	/api/login	Bejelentkezés	{ "username": "admin" }
GET	/api/products	Termékek listázása	-

7.6 Részletes feladat-specifikáció, 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

7.7 Az alkalmazott fejlesztői eszközök

Fel kell sorolnod az összes olyan szoftver eszközt, amelyet a program fejlesztéséhez, illetve a dokumentáció készítéséhez felhasználtál.

7.7.1 programozási nyelv

- **HTML5** a kliens oldali dokumentum szerkezetének a meghatározásához
- **CSS3** a kliens oldali dokumentum megjelenésének meghatározásához
- **ES6** a kliens oldali dokumentum interaktivitásának a biztosításához
- **C#** az asztali alkalmazás elkészítéséhez

7.7.2 Mások által készített keretrendszerek és modulok

- **Express** keretrendszer a backend szolgáltatások biztosításához
- **cors** modul a CORS (*Cross-Origin Resource Sharing*) webes biztonsági mechanizmus megfelelő kezeléséhez
- **mysql2** modul az Express és a MySQL adatbáziskezelő összehangolása érdekében

- **BootStrap 5.3.6** kliens oldali dokumentum gyorsabb formázása és a reszponzivitás biztosításához
- **React**
- ...

7.7.3 Fejlesztő környezet

- **Visual Studio Code** a frontend és backend programok elkészítéséhez
- **Visual Studio Community 2022**-t az asztali alkalmazás készítéséhez

7.7.4 adatbázis-kezelő rendszer

Ár és teljesítmény alapján a MySQL adatbázis kezelő rendszert választottam.

7.7.5 kép-, szöveg-, zeneszerkesztő program

- Office 365 Word alkalmazás a dokumentáció készítéséhez
- Figma

Ajánlott terjedelem: ½ -1 oldal.

8 Tesztelési dokumentáció

A **tesztelési dokumentáció** célja, hogy bemutassa: a programod különféle esetekben a kitűzött célnak megfelelően működik, és hogy tesztelted is azt – nem csak „összeraktad és kész”. Minden fejlesztői környezet biztosít a teszteléshez kiegészítő programokat. Ezekre is utalhatsz.

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

8.1 Tesztesetek

8.1.1 Normál használat – Bejelentkezés

- **Teszteset neve:** Érvényes bejelentkezés
- **Tesztelt funkció:** Felhasználói bejelentkezés
- **Bemenet:** Felhasználónév: admin, jelszó: admin123
- **Elvárt eredmény:** Bejelentkezés sikeres, főmenü megjelenik
- **Kapott eredmény:** Bejelentkezés sikeres
- **Teszt eredménye:** ✓ Sikeres

8.1.2 Extrém eset – Üres mezők

- **Teszteset neve:** Üres felhasználónév/jelszó mező
- **Tesztelt funkció:** Bejelentkezés
- **Bemenet:** Felhasználónév: (üres), jelszó: (üres)
- **Elvárt eredmény:** Hibaüzenet: „Töltse ki a mezőket!”
- **Kapott eredmény:** Hibaüzenet megjelent
- **Teszt eredménye:** ✓ Sikeres

8.1.3 Helytelen adat – Hibás jelszó

- **Teszteset neve:** Rossz jelszó
- **Tesztelt funkció:** Bejelentkezés
- **Bemenet:** Felhasználónév: admin, jelszó: 1234
- **Elvárt eredmény:** Hibaüzenet: „Hibás jelszó”
- **Kapott eredmény:** Hibaüzenet: „Hibás jelszó”
- **Teszt eredménye:** ✓ Sikeres

8.1.4 4. Bolondbiztos teszt – Hosszú bemenet

- **Teszteset neve:** 1000 karakter hosszú jelszó
- **Tesztelt funkció:** Bejelentkezés
- **Bemenet:** Jelszó: aaaa...aaaa (1000 karakter)
- **Elvárt eredmény:** Hibaüzenet, rendszer ne fagyjon le
- **Kapott eredmény:** Hibaüzenet jelent meg, nem fagyott le
- **Teszt eredménye:** ✓ Sikeres

8.2 Talált hibák

- **Hiba #1:** A jelszó mező nem rejtette el a karaktereket.
 - *Állapot:* Javítva – PasswordBox beállítva.
- **Hiba #2:** A hosszú jelszó bemenet első verzióban lefagyasztotta az alkalmazást.
 - *Állapot:* Input hossz limit beállítva (max. 50 karakter).

8.3 Automatizált tesztelés

8.3.1 Egységtesztek (Jest)

```
test("Login API returns token", async () => {  
  const res = await request(app).post("/api/login").send({ username:  
    "admin" });  
  expect(res.statusCode).toEqual(200);  
});
```

8.4 Tesztelési környezet

- **Operációs rendszer:** Windows 10, 64bit
- **Fejlesztői környezet:** Visual Studio 2022
- **Tesztverzió:** v1.0.3 (RC)

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

9 Felhasználói dokumentáció

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.

9.1 A program általános specifikációja

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

Ajánlott terjedelem: 0,5 oldal.

9.2 Rendszerkövetelmények

9.2.1 Hardver követelmények

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. Minimum gépigény (pl. RAM, CPU, tárhely).

9.2.2 Szoftver követelmények

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.)

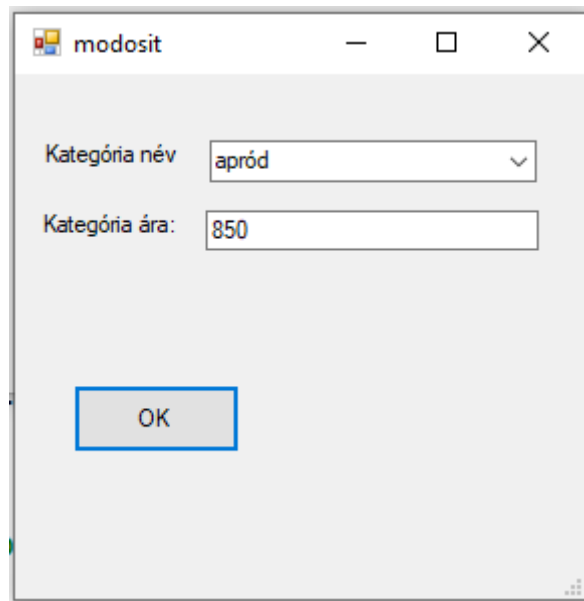
Ajánlott terjedelem: ½ -1 oldal, felsorolásszerűen leírva

9.3 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.

9.4 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 kerülj a laza stílust: rövidítések, smilie-k, szleng kizárva.
- Alkalmazz ábrákat, screenshot-okat, de ne legyen túlzott a képek aránya a szöveghez képest. Kb. 2-3 oldalanként egy ábra megfelelő.

Ajánlott terjedelem: 10-15 oldal, ábrákkal együtt.

9.5 FAQ (*Frequently Asked Questions*)

Kérdés: Hogyan állíthatok új jelszót?

Válasz: A "Elfelejtett jelszó" linkre kattintva a bejelentkezési oldalon.

10 Összefoglalás

10.1 Önértékelés

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

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

pl.:

A projekt során sikerült elmélyíteni a tudásunkat a front-end és back-end fejlesztésben, valamint az adatbázis-kezelésben. A legnagyobb kihívást a [pl. autentikáció, adatvalidálás, aszinkron kommunikáció] megvalósítása jelentette. Ezeket közös munkával és konzultációval sikerült megoldani.

Fejlődtünk a verziókezelés, csapatmunka és dokumentációs technikák területén is.

Ajánlott terjedelem: 0,5-1 oldal

10.2 Továbbfejlesztési lehetőségek

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

pl.:

- Felhasználói jogosultsági szintek bővítése
- Reszponzívabb design
- Admin statisztikai felület
- Exportálási lehetőségek (pl. PDF, Excel)
- Email értesítések integrálása
- API dokumentáció Swagger-rel

Ajánlott terjedelem: 0,5-1 oldal

11 Felhasznált irodalom

- Minden olyan forrás pontos megadása, amelyet a szakdolgozatodban felhasználtál.
- **Ne hagyj ki forrást**, ha onnan tanultál vagy idéztél.
- Ha **szó szerint** idézel, tegyél idézőjelet és lábjegyzetben jelezd a forrást is.
- **Kerüld** a nem megbízható forrásokat (pl. fórumok, ismeretlen blogok, TikTok-videók stb.).

11.1 Nyomtatott könyv

Szükséges adatok:

- Szerző(k) neve
- A könyv címe (*dőlt betűvel vagy idézőjelben*)
- Kiadó
- Kiadás helye (opcionális)
- Kiadás éve
- ISBN (ha van)

Példa:

Szabó Péter: *Bevezetés a programozásba*. Typotex Kiadó, Budapest, 2018. ISBN 978-963-279-993-1.

11.2 Internetes forrás (weboldal, blog, dokumentáció)

Szükséges adatok:

- Szerző neve (ha ismert)
- Weboldal címe vagy cikk címe
- A pontos URL
- Az utolsó letöltés/dátum, amikor elérted

Példa:

MDN Web Docs: *JavaScript Guide*.

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>

Letöltés dátuma: 2025. május 26.

Megjegyzés: Ha nincs szerző, írd a weboldal nevét.

11.3 Elektronikus dokumentum (*PDF, tananyag, prezentáció stb.*)

Szükséges adatok:

- Szerző neve
- Dokumentum címe
- Hol találtad meg (webcím vagy oktatási platform neve)
- Letöltés dátuma

Példa:

Kovács Anna: *Python alapok oktatási segédlet.*

Elérhető: moodle.sulinet.hu

Letöltés dátuma: 2025. április 5.

Ajánlott terjedelem: ½ -1 oldal.

12 Ábrajegyzék

1. kép Ez a módosító ablak a programban	17
---	----