

SMART MANAGER

Projekt specifikáció



Készítette:
Magyar Márk József,
Nagy Huba Kende

1. Tartalomjegyzék

2.	Bevezetés	3
2.1	Rövid projekt bemutatás	3
2.2	Témaválasztási magyarázat.....	3
2.3	A projekt célja.....	3
2.4	Funkciók és extrák	3
2.5	Célközönség	3
3.	Fejlesztési folyamat és rendszerkövetelmények.....	4
3.1	Rendszerkövetelmények	4
3.2	A fejlesztéshez használt főbb eszközök.....	4
3.3	UI/UX tervezés – Figma	4
3.4	Kommunikáció és együttműködés	4
4.	Fejlesztési környezet és megvalósítás	5
4.1	Operációs rendszer, környezet.....	5
4.2	Felhasználható programozási nyelvek / keretrendszerek	5
4.3	Szoftverfejlesztés	5
4.4	Három fő komponens:.....	5
5.	Megjelenés és funkciók	6
5.1	Megjelenés.....	6
6.	Dokumentáció	8
6.1	Erőforrás-terv, munkaidő nyilvántartás	8
6.2	Technikai dokumentáció	8
6.3	Forráskód dokumentáció.....	8
6.4	Adat dokumentáció	8
7.	Projekt adatlap.....	9
7.1	Projekt neve	9
7.2	Feladat rövid ismertetése	9
7.3	Programozási nyelvek / Keretrendszerek	9
7.4	Projektet összeállították	9

2. Bevezetés

2.1 Rövid projekt bemutatás

A Smart Manager egy oktatási projekt menedzsment rendszer. A szoftver támogatja a tanárok és diákok közötti hatékony együttműködést. A rendszer célja, hogy biztosítson egy átható megoldást a projektek egy helyen történő kezelésére.

2.2 Témaválasztási magyarázat

Projektünk ötlete abból a tapasztalatból származik, hogy több tanár is nehézségekről számolt be a csoportos iskolai projektek nyomon követésében és értékelésében. A problémát egyrészt a diákok nagy száma, másrészt az általuk használt különböző együttműködési platformok okozzák, ami megnehezíti az egységes ellenőrzést és az előrehaladás követését.

2.3 A projekt célja

A Smart Manager célja egy olyan projektmenedzsment rendszer kínálata, amely egyszerre segíti a felhasználók együttműködését és az adminisztrációs munkákat, továbbá a feladatokat, és a statisztikákat átláthatóbbá teszi. Remek eszköz számos, különböző célú felhasználásra. Lehet kezelni dokumentumokat, illetve munkafolyamatokat és a felhasználókat követni.

2.4 Funkciók és extrák

Három fő funkciót egyesít: filekezelés, feladatmenedzsment és valós idejű kommunikáció. A filekezelő rész lehetővé teszi a projekthez tartozó fájlok rendszerezett tárolását. A feladatmenedzsment egy Kanban tábla segítségével támogatja a csapatok munkaszervezését és nyomon követését. A beépített chat modul biztosítja a gyors és egyszerű kommunikációt a csapattagok között, így minden fontos eszköz egyetlen felületen érhető el.

2.5 Célközönség

A platform elsődleges célközönsége a diákok, illetve tanárok, akik egyszerű, átlátható eszközt keresnek közös projektek, beadandók és csapatfeladatok megszervezésére. Segíti a feladatok kiosztását, a fájlrendszerezést, valamint a gyors kommunikációt. Emellett a rendszer hasznos lehet kisebb fejlesztői csapatoknak, vagy hobbi projektekhez, ahol fontos a könnyen kezelhető, integrált munkakörnyezet.

3. Fejlesztési folyamat és rendszerkövetelmények

3.1 Rendszerkövetelmények

<u>Specifikáció</u>	<u>Minimum</u>	<u>Maximum</u>
RAM:	4GB DDR4	8GB DDR4
OS:	Windows 11 64bit	Windows 11 64bit
GPU:	Nvidia 9800 Gt 1 GB	Nvidia gtx 660 2GB
CPU:	Intel Core 2 Quad CPU Q6600	Intel Core i5-3470

3.2 A fejlesztéshez használt főbb eszközök

- IDE: Visual Studio 2022, Visual Studio Code
- Célplatform: Windows, Android, IOS
- Verziókezelés: Git (GitHub)

3.3 UI/UX tervezés – Figma

A program felhasználói felületének és képernyőelrendezésének megtervezésére a Figma nevű online UI/UX eszközt használjuk. Ez lehetővé teszi, hogy már a fejlesztés megkezdésekor vizuálisan átgondolt terveket készítsünk a menürendszerrel, szövegkörnyezetről, valamint az interaktív elemek elhelyezkedéséről. A Figma használatának köszönhetően, mivel meg van tervezve egy alap koncepció, csökkenteni tudjuk a felesleges újra tervezések számát és hatékonyabban végezzük az egyes munkafolyamatokat.

3.4 Kommunikáció és együttműködés

A közös munka során megbeszélésekre és egyeztetésekre a Discordot használjuk hiszen rendkívül hasznos és széleskörben elterjedt alkalmazás. Ez biztosítja, hogy meg tudjuk egymással osztani hívás közben a képernyőnket és élőben tudjuk követni az aktuális fejlesztési szakaszt. Ezen felül iskolai tanórákon, és személyes találkozások alkalmával is hatékonyan tudunk együttműködni

4. Fejlesztési környezet és megvalósítás

4.1 Operációs rendszer, környezet

- A rendszer többplatformos működésre lett tervezve.
- Reszponzív web alkalmazás, amely modern böngészőkben (Chrome, Edge, Safari) és mobil eszközökön is működik, webböngészőn keresztül, egyaránt Androidon és iPhone készüléken.
- Asztali kliens: Windows környezetre készített WinUI3 alkalmazás.

4.2 Felhasználandó programozási nyelvek / keretrendszerek

- Backend: Node.js → Express
- Frontend: Vue.js
- Asztali kliens: C# (WinUI3 / Avalonia)
- Adatbázis: SQL (PostgreSQL)
- Valós idejű kommunikáció: Websocket (Socket.IO)

4.3 Szoftverfejlesztés

A fejlesztés során Visual Studio-t, Postman-t és Figma-t használunk, a kommunikáció Discordon zajlik. A tiszta kód elveit követjük, és folyamatosan teszteljük a rendszer stabilitását.

4.4 Három fő komponens:

Frontend (webes felület és asztali kliens): A felhasználói interfész felel a projektek, feladatok és csapatok vizuális megjelenítéséért. A webes verzió böngészőben fut, rezponzív kialakításának köszönhetően asztali és mobil eszközökön is használható. Az asztali kliens Windows platformra készül, WinUI 3 technológiával, amely gyors hozzáférést nyújt a rendszer fő funkcióihoz.

Backend (REST API): A szerveroldali komponens felel az alkalmazás logikájáért és az adatkezelésért. A Node.js és Express alapú REST API kapcsolatot biztosít a kliens és az adatbázis között. A rendszer kezeli a felhasználói hitelesítést, a projektekhez és feladatokhoz tartozó CRUD műveleteket, valamint a valós idejű kommunikációt Socket.IO segítségével.

Adatbázis (PostgreSQL): A relációs adatbázis tárolja a felhasználók, projektek, feladatok, üzenetek és statisztikák adatait. Az adatkezelést ORM-en (Prisma) keresztül végezzük, amely biztosítja a biztonságos és strukturált adatkapcsolatot.

5. Megjelenés és funkciók

5.1 Megjelenés

Felhasználók / diákok felülete:

1. Belépés

- bejelentkezés / regisztráció → e-mail + jelszó.

2. Főoldal

- Közelgő határidők (beadandók).
- Saját státusz (hány feladat folyamatban, hány kész).
- Legutóbbi közlemények.

3. Feladatkezelés

- Kapott feladatok listája.
- Feladat beadása (fájl feltöltés → pl. .zip, .docx, .cs).
- Saját beadások állapotának követése (pl. „Folyamatban”, „Elküldve”, „Értékelve”, „Elkészve”).
- Határidő figyelmeztetés (pl. piros kiemelés, ha lejárt).

4. Csapatmunka

- Kanban tábla: „Teendő – Folyamatban – Hiányzik – Kész”.
- Feladatkártyák: név, felelős, határidő.
- Közös fájlok feltöltése.
- Változásnapló: ki mikor adott hozzá/módosított valamit.

5. Chat modul

- Egyéni chat (felhasználó ↔ felhasználó).
- Csapatchat (projekt szinten).
- Adminisztrátor-felhasználó chat (pl. kérdés beadandóval kapcsolatban).

6. Statisztika / Visszajelzések

- Beadott feladatok száma és státusza.
- Értékelések megtekintése.
- Grafikon: teljesítési arány, késésekkel együtt.

7. Beállítások

- Dark/Light mód.
- Profil Adatok (e-mail, jelszó).

Adminisztrátorok/Tanárok felület:

1. Belépés

- Tanár/Admin bejelentkezés (magasabb jogosultságokkal).

2. Főoldal

- Legutóbbi beadott feladatok.
- Figyelmeztetések (pl. „adott fhsz. nem adta le a feladatot időre”).
- Statisztikai összegzés (átlag pontszám, teljesítési arány).

4. Feladatkiosztás

- Új feladat létrehozása (név, leírás, határidő, csatolmány).

- Feladat kiadása.
- Határidők követése → listázás, ki adta le időben.

5. Értékelés

- Beadott feladatok megtekintése (ki mikor adta le, milyen fájl adott le).
- Pontszám / jegy megadása.
- Szöveges visszajelzés írása.
- Automatikus státusz frissítés (pl. „Értékelve”).

6. Statisztikák és riport, osztályátlag megjelenítése.

- Hiányzó beadások, késések kimutatása.
- Exportálás PDF-be.

7. Kommunikáció

- Üzenet küldés.
- Egyéni üzenet felhasználóknak/diákoknak.
- Visszajelzés beadott munkákhoz közvetlenül.

Összefoglalva:



Összehasonlító táblázat

Funkciók	Diák / Felhasználó	Tanár / Adminisztrátor
Bejelentkezés	✓	✓
Dashboard	✓	✓
Feladat Beadás	✓	✗
Feladat Kiosztás	✗	✓
Projekt Board (KanBan)	✓	✗
Chat	✓	✓
Értékelések kezelése	✗ (Csak megtekintheti, nem szerkezhetheti)	✓ (Diákok beküldött munkáira osztályzat szerkeztés)
Értékelések megadása	✗ (Csak megtekintheti, nem szerkezhetheti)	✓ (Diákok beküldött munkáira osztályzat)
Statisztika	✓ (Saját teljesítmény)	✓ (Diákok teljesítménye)
Anyagok Feltöltése / Leadása	✓ (Projektek Leadása)	✓ (Munkák feltöltése Diákoknak)
Export (PDF)	✗	✓

6. Dokumentáció

6.1 Erőforrás-terv, munkaidő nyilvántartás

- Csapatlétszám: 2 fő fejlesztő.
- Munkamegosztás: egyik fő a backend + desktop fejlesztésért felel, a másik fő a frontend + UI megvalósításáért.
- Munkaidő: heti 10–15 óra fejenként, összesen kb. 240–360 munkaóra.
- Ütemezés: 24 hét, heti mérföldkövekkel (lásd Gantt-diagram).

6.2 Technikai dokumentáció

- Adatbázismodell: ERD diagram, SQL dump.
- Rendszerarchitektúra: backend, frontend és desktop komponensek kapcsolata.
- API dokumentáció: Swagger/OpenAPI specifikáció a végpontokról.

6.3 Forráskód dokumentáció

- Kommentek: tiszta kód elvek, metóduson kenti rövid leírás.
- README fájl: telepítési és futtatási útmutató.
- Kódfelépítés: modulok és mappastruktúra bemutatása.
- Tesztdokumentáció: futtatott tesztek és eredmények.

6.4 Adat dokumentáció

A programhoz kapcsolódó összes változó, adat, ide tartoznak többek között:

- Felhasználói adatok (felhasználónév, jelszó, email cím)
- Projektmunka adatai (név, tagok)
- A feltöltött munka adatai (fileok - Pl: css, js, html, cs, xml, txt...stb.)
- Valós idejű kommunikáció (üzenetek)
- Statisztika (adott projektben munkaóra, stb.)
- Aktivitás (ki mikor volt utoljára elérhető)
- To-Do táblázat adatai (Határidő, feladat státusz: Befejezett, folyamatban, Késés)
- Jogosultságok és szerepkörök (pl: Admin, tanár, diák, projektvezető)
- Jogosultság szint: be lehet állítani, hogy ki mit láthat és mit módosíthat létrejött csapat(ok) adatai.

7. Projekt adatlap

7.1 Projekt neve

Smart Manager

7.2 Feladat rövid ismertetése

Projekt menedzser, amely kiszolgál egy teljes projektfolyamat lebonyolításához szükséges minden igényt.

7.3 Programozási nyelvek / Keretrendszerek

C#, HTML, CSS, JavaScript, SQL, Vue.js, Node.js

7.4 Projektet összeállították

Magyar Márk József, Nagy Huba Kende