Smart Manager

Szoftver specifikáció



Készítette:

Magyar Márk József,

Nagy Huba Kende

Tartalomjegyzék

[2. Bevezetés 3](#_Toc210119456)

[2.1 Rövid projekt bemutatás 3](#_Toc210119457)

[2.2 Témaválasztási magyarázat 3](#_Toc210119458)

[2.3 Mi célt szolgál 3](#_Toc210119459)

[2.4 Funkciók és extrák 3](#_Toc210119460)

[2.5 Célközönség 4](#_Toc210119461)

[3. Fejlesztőkörnyezet és hardver - Fejlesztői Dokumentáció 5](#_Toc210119462)

[3.1 A fejlesztéshez használt főbb eszközök 5](#_Toc210119463)

[3.2 Fejlesztésre használt számítógép 5](#_Toc210119464)

[3.3 UI/UX tervezés – Figma 5](#_Toc210119465)

[3.4 Technikai konzultáció 5](#_Toc210119466)

[3.5 Tárolt adatok 6](#_Toc210119467)

[4. Elvárások a feladattal kapcsolatban 7](#_Toc210119468)

[4.1 Operációs rendszer, környezet 7](#_Toc210119469)

[4.2 Felhasználandó programozási nyelvek / keretrendszerek 7](#_Toc210119470)

[4.3 Megoldás formátuma 7](#_Toc210119471)

[4.4 Szoftverfejlesztés 7](#_Toc210119472)

[4.5 Modulok 7](#_Toc210119473)

[5. Megjelenés és funkciók 8](#_Toc210119474)

[5.1 Megjelenés 8](#_Toc210119475)

[6. Dokumentáció 11](#_Toc210119476)

[6.1 Erőforrás-terv, munkaidő nyilvántartás 11](#_Toc210119477)

[6.2 Technikai dokumentáció 11](#_Toc210119478)

[6.3 Forráskód dokumentáció 11](#_Toc210119479)

[7. Projekt adatlap 12](#_Toc210119480)

[7.1 Projekt neve 12](#_Toc210119481)

[7.2 Feladat rövid ismertetése 12](#_Toc210119482)

[7.3 Programozási nyelvek / Keretrendszerek 12](#_Toc210119483)

[7.4 Specifikációt összeállította 12](#_Toc210119484)

# Bevezetés

## 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.

## Témaválasztási magyarázat

Azért választottuk ennek a platformnak a megvalósítását, mivel több tanár is számolt be nehézségekről. Többek között a diákok nagy száma, valamint a közös munkát támogató platformok sokfélesége is kihívást jelentett. Hiszen ha például van 4 darab 2-2 fős csapat osztályonként, az egyik csapat discordot és githubot használ (ahol követhető ki mit csinált és mikor), egy másik csapat pedig használt messengert, és google drive-ot (ahol nem lehet olyan egyszerűen vissza követni, hogy ki, mikor, és mit változtatott), akkor jól látható, hogy milyen komplikálttá válik a feladatok ellenőrzése.

## Mi célt szolgál

A Smart Manager célja egy olyan projektmenedzsment rendszer kinálata, amely egyszerre segíti a felhasználók együttműködését és az adminisztrációs munkálatokat, továbbá átláthatóvá teszi a feladatokat, és a statisztikákat. Remek eszköz számos célú felhasználásra, mivel lehet kezelni dokumentumokat, illetve egyéb munkafolyamatokat és a felhasználókat követni, hogy ki, mikor, és mit végzett egy adott projekt során.

## Funkciók és extrák

Három fő funkciót egyesít: filekezelés, feladatmenedzsment és valós idejű kommunikáció. A filekezelő rész lehetővé teszi a projekthez tartozó fájlok rendszerezett tárolását (és verziókövetését). 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.

## Célközönség

A platform elsődleges célközönsége a diákok és 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.

# Fejlesztőkörnyezet és hardver - Fejlesztői Dokumentáció

## A fejlesztéshez használt főbb eszközök

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

## Fejlesztésre használt számítógép

|  |  |  |
| --- | --- | --- |
| Specifikáció | Minimum | Maximum |
| RAM: | 16GB DDR4 | 16GB DDR4 |
| OS: | Windows 10 64bit | Windows 11 64bit |
| GPU: | AMD Radeon RX 570 8gb | Geforce rtx 2070 super 6gb |
| CPU: | AMD Ryzen 5 2600 | Intel Core i7-10970H |

## 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 menürendszerről, szövegkörnyezetről, valamint az interaktív elemek elhelyezkedéséről. A Figma használatának köszönhetően el tudunk kerülni több hibalehetőséget is. Mivel meg van tervezve egy alap koncepció így tudjuk csökkenteni a felesleges újra tervezések számát és hatékonyabban dolgozunk.

## Technikai konzultáció

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 min dolgozik a csapattársunk. Ezen felül iskolai tanórákon, személyes találkozások alkalmával is tudunk együtt dolgozni.

## Tárolt adatok

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, ha későn adja le a User akkor számolja mennyi idővel később adta le)
* Jogosultságok és szerepkörök (pl: Admin, tanár, diák, projektvezető)
* Jogosultság szint: be lehet állítani, hogy ki mit láthat és mit módosíthat létrejött csapat(ok) adatai.

# Elvárások a feladattal kapcsolatban

## 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-on (Safari, Google Chrome).
* Asztali kliens: Windows környezetre készített WinUI3 alkalmazás.

## 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)

## Megoldás formátuma

A végső megoldás kettő komponensből áll:

1.Webes frontend és Asztali kliens –felhasználó számára projektek kezelése / adminisztrátor számára statisztikák és tagok kezelése.

2.Backend REST API – a központi logika és adattárolás.

A teljes forráskód, dokumentációval együtt GitHub repository-ban kerül tárolásra, verzió követéssel.

## Szoftverfejlesztés

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

## Modulok

* Felhasználókezelés (bejelentkezés, regisztráció, jogosultságok).
* Projektfeladat-kezelés (Kanban board, státuszok).
* Statisztikák és riportok (munkaóra, feltöltések száma, stb.).

# Megjelenés és funkciók

## Megjelenés

Felhasználóknak/diákoknak amit tartalmazni fog:

1. Belépés

* bejelentkezés / regisztráció -> e-mail + jelszóval.

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”, „Elkésve”).
* Határidő figyelmeztetés (pl. piros kiemelés, ha lejárt és időzítő indul, hogy mióta nincs leadva).

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).

Adminisztrátor/Tanároknak, amit tartalmazni fog:

1. Belépés

* Tanár/Admin bejelentkezés (magasabb jogosultságokkal).

2. Főoldal

* Legutóbbi beadott feladatok.
* Figyelmeztetések (pl. „5 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

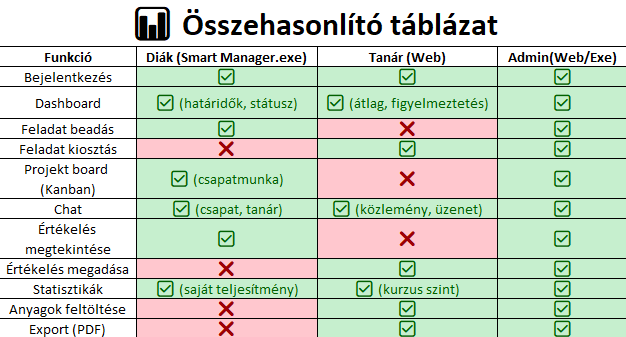
* Beadott feladatok megtekintése (ki mikor adta le, milyen fájlt adott le).
* Pontszám / jegy megadása.
* Szöveges visszajelzés írása.
* Automatikus státusz frissítés (pl. „Értékelve”).

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:

# Dokumentáció

## 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 8–10 óra fejenként, összesen kb. 150–160 munkaóra.
* Ütemezés: 8 hét, heti mérföldkövekkel (lásd Gantt-diagram).

## Technikai dokumentáció

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

## Forráskód dokumentáció

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

# Projekt adatlap

## Projekt neve

Smart Manager

## Feladat rövid ismertetése

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

## Programozási nyelvek / Keretrendszerek

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

## Specifikációt összeállította

Magyar Márk József, Nagy Huba Kende