BGSzC Pestszentlőrinci Közgazdasági és Informatikai Szakgimnáziuma

1184 Budapest Hengersor 34.

1. Záró dolgozat
2. PlanUP

Konzulens tanár: Készítette:

Szekrényes Gábor Molnár Levente, Nagy Martin, Kovács-Major Márton

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 munkafolyamat során minden eddig szerzett ismeretünket felhasználtuk, és még újakat is szereztünk. Célunk a programmal, hogy egy olyan alkalmazást készítsünk, amely segítséget nyújt a felhasználóknak szabadidő programajánlásokban és szűrésekben, miközben egy felhasználóbarát felületen akár többedmagával is válogathat a lehetőségek közül.

A dokumentáció egyszerre nyújt segítséget a program telepítéséhez és használatához, és nyújt betekintést annak készüléséről és tesztelési folyamatairól.

## A felhasznált ismeretek

Tanulmányaink során rengeteg ismeretet szereztünk melyek kifizetődőnek bizonyultak a projekt készülése során. Mindenekelőtt legfontosabb volt a csapatmunka, és a feladatok hatékony felosztása egymás között. A csapatunknak voltak erősségei, és gyengeségei is, melyeket figyelembe véve próbáltunk meg egy minél kiegyensúlyozottabb munkamegosztást teremteni egymás között.

Az évek során rengeteget tanultunk az adatbáziskezelésről, webszerkesztésről, és asztali alkalmazásfejlesztésről. Ezen tanulmányok a mestermunkán is viszontláthatók.

Az iskolai tanulmányokon kívül is szert tettünk új ismeretekre, melyekre a mestermunkák bizonyos részei miatt volt szükség. Ezek a későbbiekben részletezve lesznek.

## A felhasznált szoftverek

A projekt során különböző programokat használtunk, volt olyan ami feltétlen szükséges volt a koncepció megvalósításához, és volt ami tanórai kötelezettséggel került használatra, és bizonyos szempontokból hasznosnak bizonyultak.

**Trello**: A csapatmunkához rendkívül hasznos internetes alkalmazás, mellyel nyomon tudtuk követni a haladásunkat egymás között, és szét tudtuk válogatni az ötleteket, fejlesztés alatt lévő komponenseket, és a már elkészült részeket.

**ChatGPT**: A ChatGPT-vel rengeteg problémát oldottunk meg a fejlesztés során, nagyon sok esetben tudott nekünk segíteni, a kódunkat ellenőrizni és javítani hibás tesztek során. Az alkalmazásban szereplő képeket a mesterséges intelligencia készítette az esetlegesen felmerülő szerzői jogi problémák elkerülése végett.

**Visual Studio Code**: A mestermunka során a legtöbbet használt alkalmazás, ezen belül készült el a program. A bővítményekről és egyéb specifikációkról a program telepítésénél (3) lesz még szó.

**MySQL Workbench**: Az adatbázis kezeléséhez használtuk ezt az alkalmazást, itt el tudtuk készíteni az egyed-kapcsolat diagramokat, és az adatbázist szerkeszteni a Visual Studio-s alkalmazásunkkal.

**XAMPP**: Ez az alkalmazás biztosította a MySQL kapcsolatot.

**Github**: A mestermunka egyik központi eleme, amin keresztül történtek a fejlesztések hónapokon keresztül, különböző számítógépeken otthon és közoktatásban egyaránt. A verziókövető rendszernek köszönhetően bármikor vissza tudtunk váltani előző verziókra, és nyomon tudtuk követni egymás munkáját, és hozzáférni a fájlokhoz bárhol, bármikor, bármilyen eszközről.

**Teams**: A közoktatásban is elterjedt alkalmazást használtuk a kommunikációra és bizonyos adatok tárolására annak chat felületén, illetve itt tartottuk a kapcsolatot és töltöttük fel a haladásunkat szaktanárainknak és konzulensünknek.

**Postman**: A backend részen használtuk az API-k tesztelésére és a Frontend-el való kapcsolat ellenőrzésére.

**//FileZilla**: Ez részünkről egy teljesen opcionális döntés volt, hogy szeretnénk saját webtárhelyre feltölteni a munkákat, és nem egy lokális hálózaton bemutatni azt.

# Felhasználói dokumentáció

## A program általános specifikációja

A PlanUP lehetőséget biztosít arra, hogy a felhasználók közösen válogassanak különböző hétköznapi programok közül. Fedezze fel egyedül vagy akár társaságban Budapest programjait, válogasson kedvére, vagy hasonlítsa össze véleményét társaival, hogy közösen találjanak egy olyan programot, amely mindenki számára megfelel.

*LIKE/DISLIKE Rendszer*

Szimplán eldöntheti, hogy tetszik-e egy program, vagy sem, és ezt az adatbázis megjegyzi, és kidobja a választott programokat és a hozzá hasonlókat.

*Szobakódok*

Hozzon létre saját szobát magának és társainak, ahol egymással párhuzamosan tudnak válogatni a programok közül, amelyeket a rendszer később összevet, hogy a legoptimálisabb programokat jelenítse meg.

*Több mint 50 program*

Válogasson több mint 50 program közül akár ingyenes, fizetős, fix időpontú vagy rugalmasabb időbeosztás szerint.

*Bízza ránk a tervezést!*

A PlanUP nem csak programajánlatokat tesz fel önnek, hanem megkönnyíti még azoknak a megvalósítását is közvetlen foglalásokkal, megadott elérhetőségekkel vagy akár ajánlott tervekkel is, így önnek nincs más dolga, mint megtalálnia az ideális programot.

## Rendszerkövetelmények

### Hardver követelmények

Hardver követelmények:

- Minimum: 4GB RAM, Intel i3 processzor

- Ajánlott: 8GB RAM, Intel i5 vagy erősebb processzor

### 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.) 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)
* 

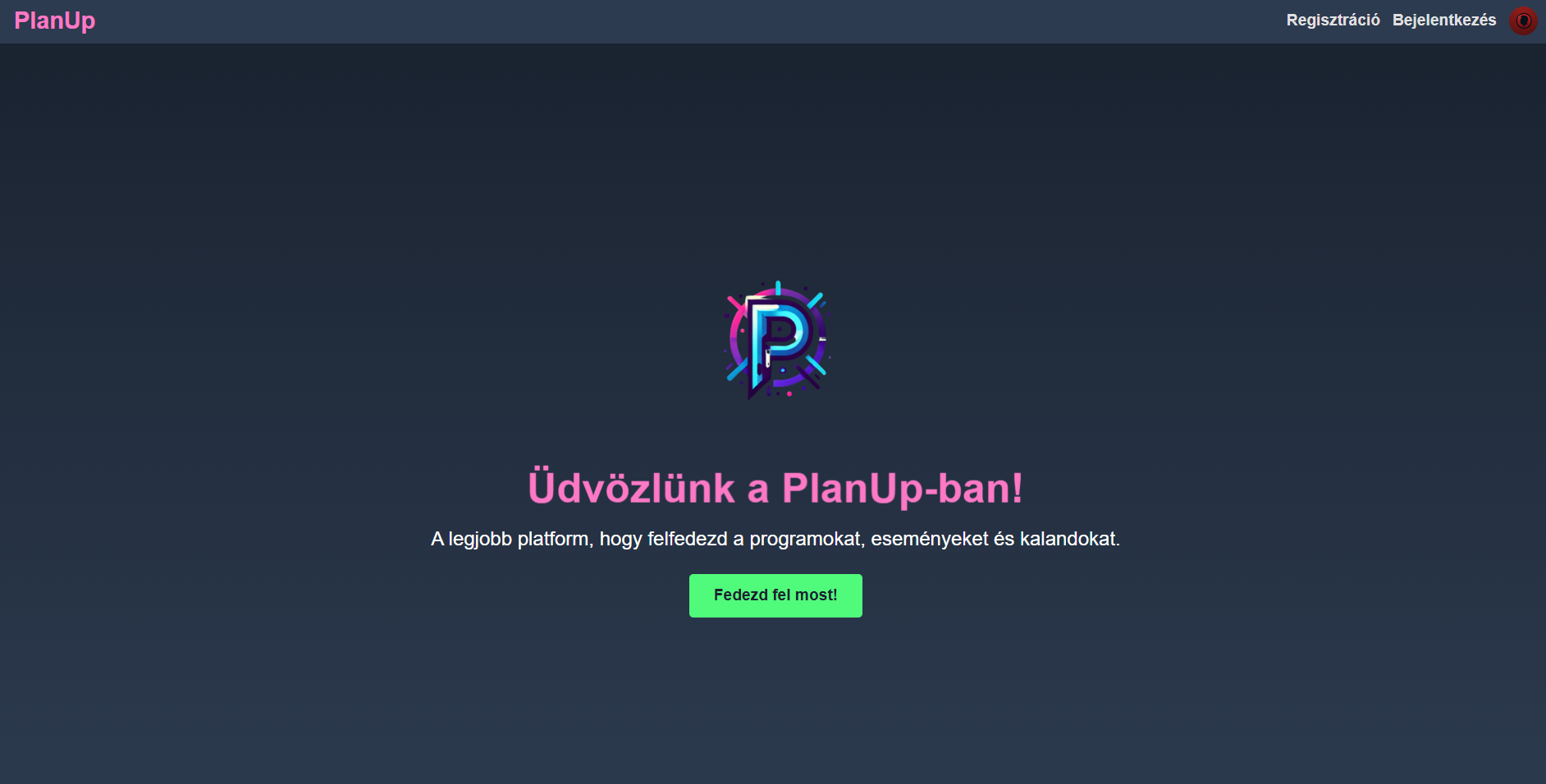
1. kép Ez a módosító ablak a programban

Ajánlott terjedelem: 2 -4 oldal, ábrákkal együtt.

## A program használatának a részletes leírása

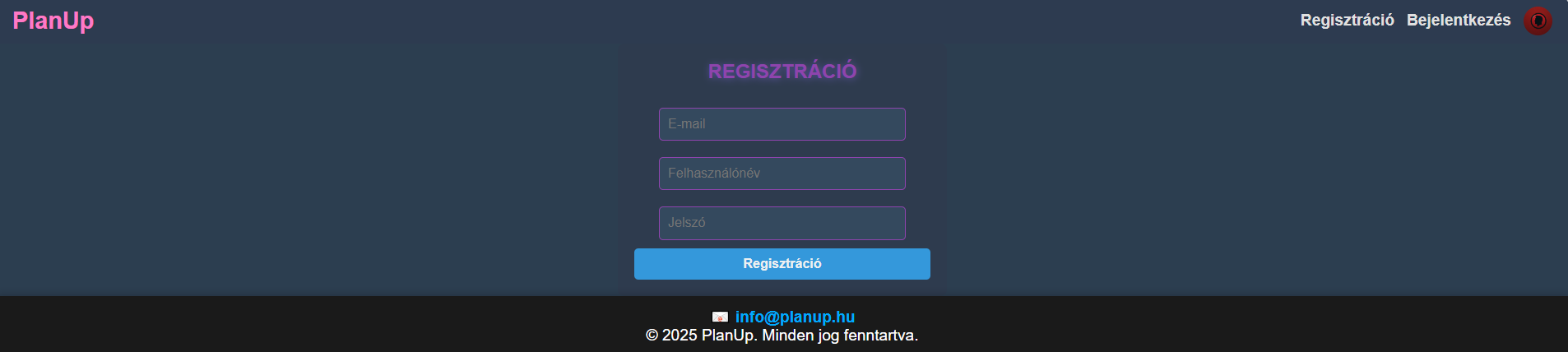
***Főoldal planup.hu***

Megnyitáskor a főoldalon találjuk magunkat, ahol üdvözöl az alkalmazás, és a menüpontok mellett a Fedezd fel most! gombbal lehet elindítani a programot. A menüsorból még elérhető a *bejelentkezés* és *regisztrációs* gomb. Ha már van felhasználói fiókunk, akkor a gombra kattintva az oldal átvisz a bejelentkező felületre, viszont mindenekelőtt szükséges a regisztráció az oldalra.



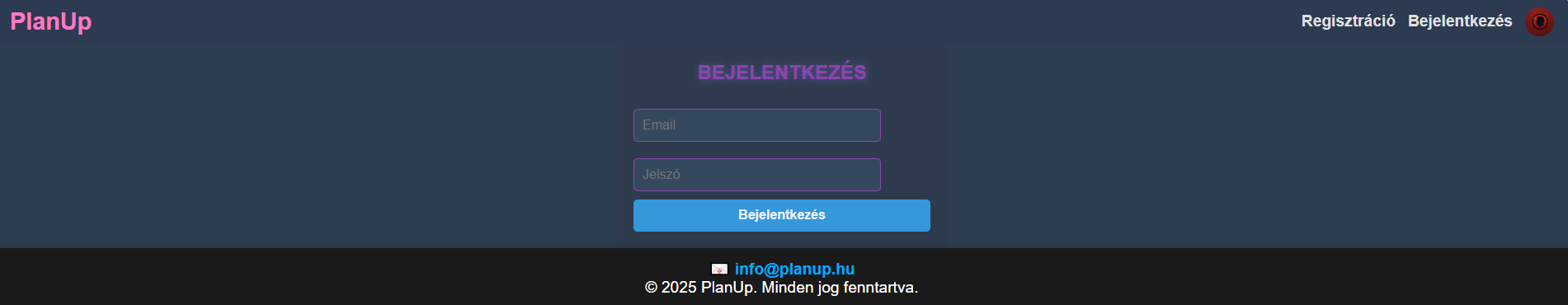
***Regisztrációs felület planup.hu/register***

A regisztrációs oldalon lehetősége van az új felhasználóknak egy profilt létrehozniuk. Ennek a módja rendkívül leegyszerűsített, mindössze egy email, felhasználónév és jelszó megadása szükséges ehhez. Ha megtörtént a regisztráció, az oldal felviszi az adatokat az adatbázisba, és ezután a megadott adatokkal már be is léphetünk újonnan létrehozott fiókunkba!



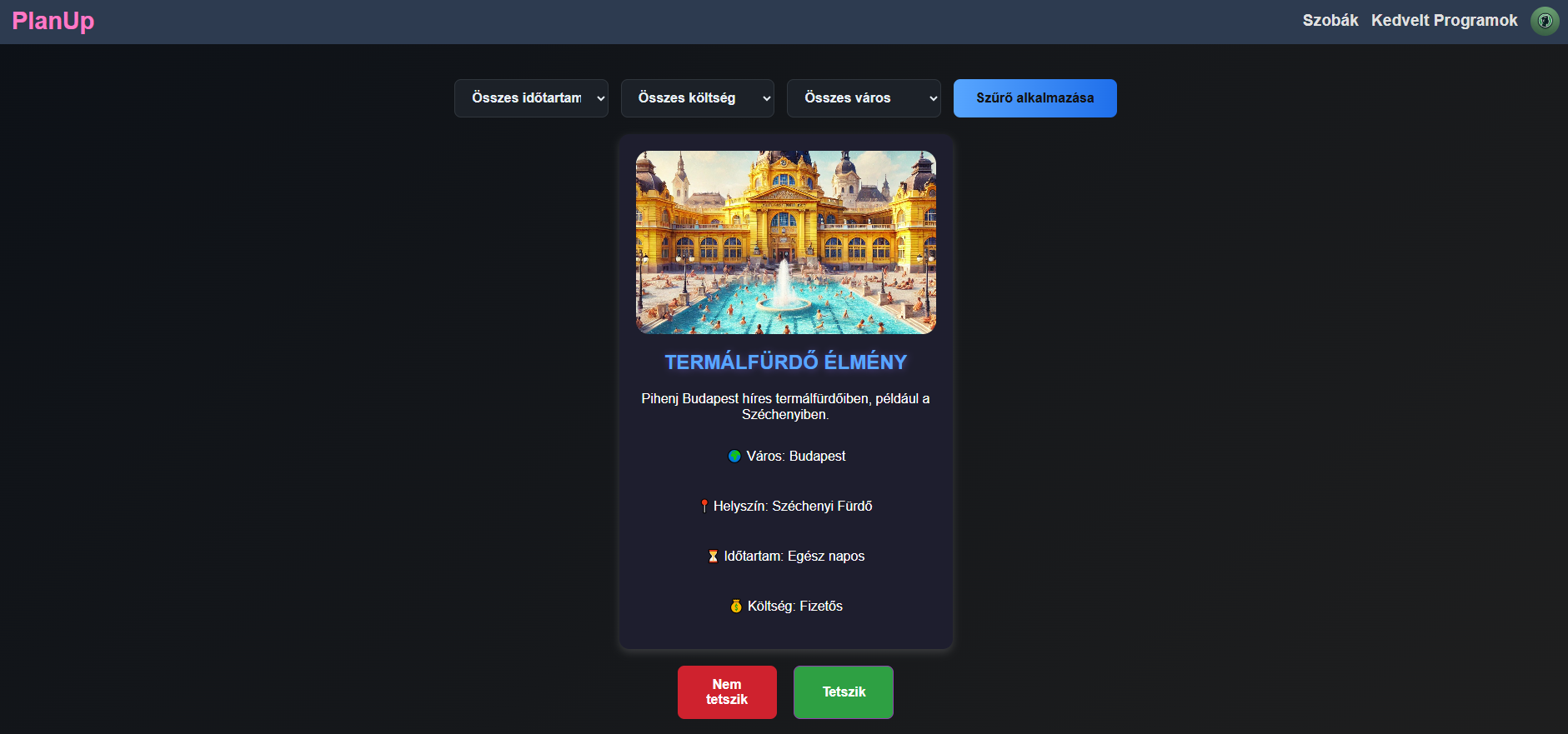
***Bejelentkezési felület planup.hu/login***

Az imént regisztrált, vagy már azelőtt létrehozott profil adataival lesz elérhető a bejelentkezés. Csak írjuk be az e-mail címet és a jelszót, és a gomb megnyomása után az oldal kiírja, ha sikeres volt a bejelentkezés, és átirányít a válogató felületre. A felső menüsorban bejelentkezés után a profil ikon zölddé változik, ezzel indikálva a belépett felhasználó állapotot.



***Programpörgető felület planup.hu/swipe***

Megérkeztünk a fő attrakcióhoz! A PlanUP itt dobja fel a program ajánlásait a felhasználónak, kezdve egy véletlenszerű programmal. Elsőként menjünk végig a szűrő feltételeken:

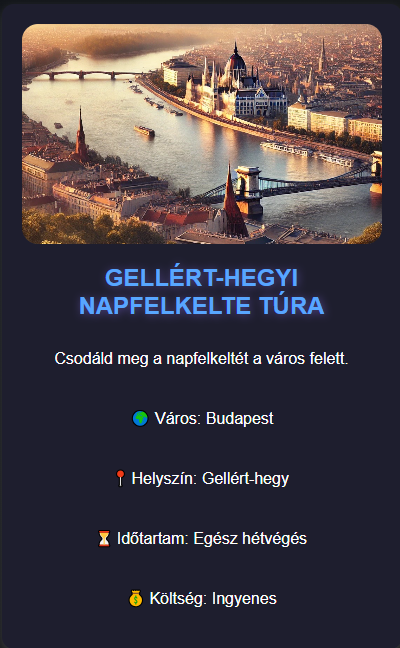


Időtartamok: A PlanUP által összegyűjtött programok sokszínűek és mértékűek, éppen ezért lehetőséget nyújtunk arra, hogy preferenciáid alapján válogathass olyan programok közül, amelyek különböző hosszúságúak.   
A választható időtartamok között szerepel például az "*Egész hétvégés*" opció, ami nem feltétlen azt jelenti, hogy a program egy egész hétvégét vesz igénybe, inkább mint csak egy indikátorként működik, hogy a program egész hétvégén elérhető, bármikor tervez is menni a felhasználó.  
A *félnapos* programok közé tartoznak olyan lehetőségek, amik nem vesznek igénybe egy teljes napot, ilyen például egy vacsora, egy hajókirándulás vagy egy opera előadás.   
Az *egésznapos* programok, mint például egy termálfürdő élmény vagy egy kirándulás akár egy teljes napot felölelhetnek, éppen ezért akkor érdemes ezeket az opciókat szűrni, amikor tudjuk, hogy akár a teljes nap is a program időtöltésével telhet.

Költségopciók: A PlanUP által összegyűjtött programok különbözhetnek még költség alapján is, rengeteg ingyenes programmal is szolgálunk, amikhez nincs szükség belépőjegy vagy részvételi díj vételére. Természetesen lehetőséget nyújtunk nagyobb költséggel járó programok böngészésére is.

Város: A PlanUP lehetőséget nyújt arra, hogy akár pontos helyszíneket megjelölve keressünk programokat, ugyanakkor nem garantált, hogy az adott helyszínre kidob programokat.

A szűrök alkalmazásához meg kell nyomni a "Szűrő alkalmazása" gombra, majd ha változtatnánk a feltételeken, akkor a "Szűrő kikapcsolása" után tudjuk csak újra megnyomni a "Szűrő alkalmazása" gombot újra az új paraméterekkel.



*A programkártya leírása:*

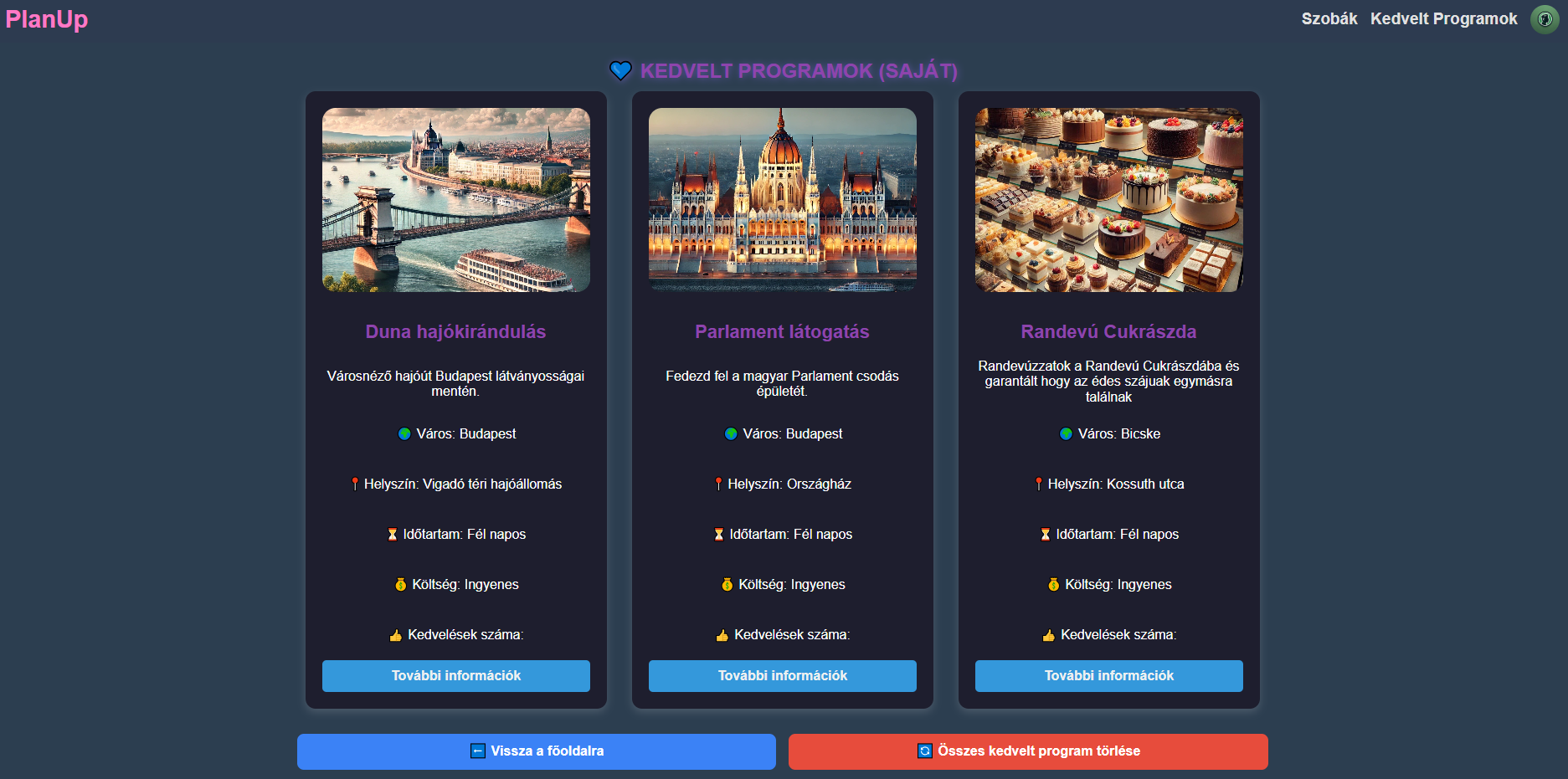
A legelső dolog, ami szembe tűnhet a kártyán az az illusztráció, ami a programhoz készült. Ezek a mestermunkánkban jelenleg mesterséges intelligencia által generált képek az esetlegesen felmerülő szerzői jogi problémák elkerülése végett. A képet követi a program címleírása, az alatt pedig a rövid, egymondatos összefoglalója. Végezetül a program helyszíne, időtartama és költsége is felsorolt, amiből az utóbbi kettő a fentebb említett lehetőségek alapján szűrhető a felhasználó által.

*Tetszik/Nem Tetszik:*

A programkártya alatt megjelenő "Nem teszik" és "Tetszik" gombbal dönthetünk arról, hogy a kártyán megjelent program érdekli-e a felhasználót, vagy sem. Ezt az adatot elmenti az adatbázis. Miután elfogytak a programok, megjelenik egy "Összegzés megtekintése" gomb, amivel meg lehet tekinteni a kedvelt programokat.

***Kedvelt programok összegzése planup.hu/liked-programs | planup.hu/summary***

Helyzettől függetlenül dob tovább a program a következő oldalakra: A program a „liked-programs” részre irányít át, ha a felhasználó egyedül válogatott a programok közül. A program szobakód használata esetén a „summary” oldalra dob át minden felhasználót, akinek a szobakódja egyezik.  
 Az oldal feldob minden programot, ami kedvelve volt bármelyik felhasználó által. Minden kártya ugyanúgy jelenik meg mint a válogatásnál, annyi különbséggel, hogy a kártya alján láthatjuk, hogy a szobából hányan kedvelték az adott programot. (Ha a felhasználó egyedül pörget, csak az ő preferált programjai fognak megjelenni az oldalon).  
 Ezután lehetőség nyílik a választásra, kétféle módon. Ha a felhasználók a kedvelések alapján döntésre jutnak, akkor az általuk választott programról további információkat kaphatnak, ha rákattintanak a kártyára. Alternatívaként, ha még a lájkok száma ellenére sem tudnak döntésre jutni - vagy döntetlen helyzet áll fent két vagy több program között - akkor az oldal alján lehetőség van egy szerencsekerék pörgetésre. Ezen a keréken az összes program megjelenik, ami kedvelve volt, és a programok amik nagyobb népszerűséget élveztek a választásoknál több teret kapnak. Itt lehetőség van pörgetni egyet, és a kerék szerencse alapon kidob egy programot.



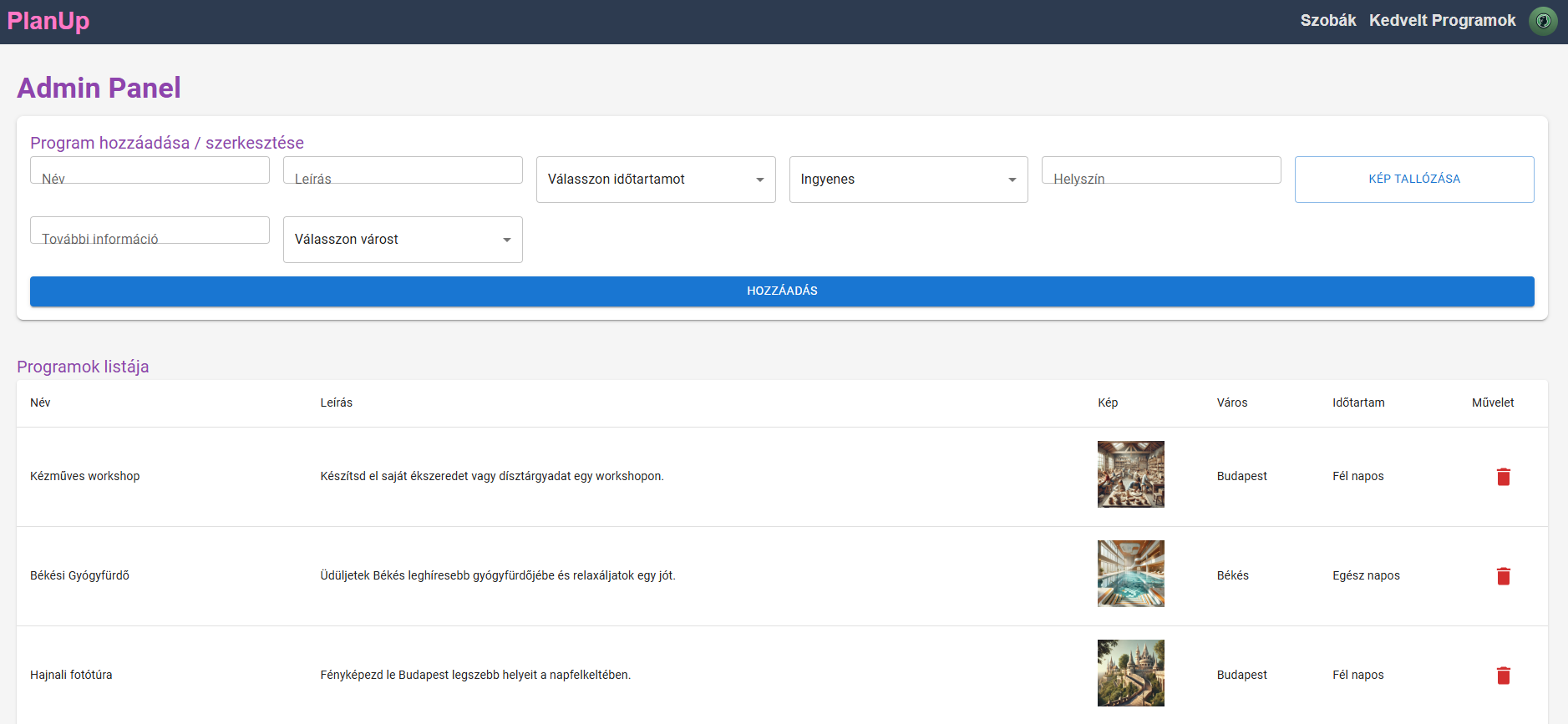
***Szobák felület planup.hu/rooms***

A szobák oldalon van lehetőség összekapcsolódni más felhasználókkal szobakódok alapján. Új szobát létrehozni is van lehetőségünk, és a kapott kódot kimásolva mások is becsatlakozhatnak a szobába. A felület kiírja az összes felhasználó nevét aki a szobában tartózkodik, legfelül jelenik meg a szoba létrehozójának a neve. Amint mindenki készen áll, és megnyomja a „pipa” gombot, a válogatásra átirányító gomb kattinthatóvá válik mindenkinél, és ez átirányítja a felhasználók a „swipe” oldalra, aminek URL-je a szoba azonosító kódjával is el van látva.



***Admin felület planup.hu/admin***

Ez a felület fejlesztői szempontból releváns, átlagos felhasználók számára nem elérhető a PlanUP ezen része. Jelenleg 3 admin jogosultsággal rendelkező profil létezik, ezek mind a PlanUP fejlesztőihez tartoznak. Az admin felület egy felhasználóbarát környezetet biztosít az új programok feltöltéséhez, frissítéséhez vagy törléséhez, ugyanígy a felhasználók profiljának kezeléséhez is. Ha arra kerülne a sor, hogy a programunkat eladnánk, az admin felülettel biztosíthatjuk azt is, hogy a következő tulajdonosok mélyebb programozói tudás nélkül is tudják kezelni a PlanUP dinamikus elemeit.



# 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

A PlanUP alkalmazás fejlesztéséhez és dokumentációjának elkészítéséhez az alábbi fejlesztői eszközöket használtam:

* **Programozási nyelvek:**
  + JavaScript (Node.js backend, React frontend)
  + HTML és CSS (webes felület kialakítása)
* **Fejlesztői környezet:**
  + Visual Studio Code (a forráskód szerkesztéséhez és hibakereséshez)
  + Vite (frontend fejlesztési környezet)
  + Node.js runtime környezet (backend futtatása és kezelése)
* **Adatbázis-kezelő rendszer:**
  + MySQL Workbench 8.0 (adatbázis-tervezéshez, kezeléshez)
  + XAMPP Control Panel (MySQL szerver futtatása)
* **Verziókezelő rendszer:**
  + Git és GitHub (verziókövetés, projektmenedzsment és közös elérés)
* **Modulok, csomagok, könyvtárak:**
  + React (*https://react.dev/*) – webes felhasználói felület kialakítására, MIT licenc alapján
  + Express.js (*https://expressjs.com/*) – backend API-kezeléshez, MIT licenc alapján
  + Axios (*https://axios-http.com/*) – HTTP kérések kezelésére, MIT licenc alapján
  + JSON Web Token (JWT) (*https://jwt.io/*) – autentikációhoz, MIT licenc alapján
  + Cookie-parser (*https://github.com/expressjs/cookie-parser*) – HTTP sütik kezeléséhez, MIT licenc alapján
* **Képszerkesztő program:**
  + Adobe Photoshop CC (képek szerkesztése, grafikai elemek előkészítése)
* **Dokumentációkészítő eszközök:**
  + Microsoft Word (szöveges dokumentáció összeállítása)
  + Microsoft PowerPoint (bemutatók, prezentációk készítése)

A felhasznált külső modulok mindegyike szabadon felhasználható, nyílt forráskódú licenc alatt érhető el (MIT licenc). Ezek felhasználása és integrálása megfelel a licencfeltételeknek.

## 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

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

## 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

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.

# Ábrajegyzék

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