BGSzC Pestszentlőrinci Közgazdasági és Informatikai Szakgimnáziuma

1184 Budapest Hengersor 34.

1. Záró dolgozat

S.O.S. Munka

Konzulens tanár: Készítette:

Szekrényes Gábor Fábián Zsolt Ferenc, Rámháp Gergő, Wágner János

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.

Az S.O.S. munka egy gyors, egyszerű szakmunkás hirdető és kereső weboldal. A felhasználók regisztrálnak felbérlőként vagy munkásként, lehetőségük van hirdetéseket létrehozni, valamint hirdetések között böngészhetnek. A hirdetéseken keresztül lehet üzenetet küldeni, időpontot foglalni, valamint értékelést hagyni.  
  
A navigációs sávon lévő profil ikonra kattintva lehet bejelentkezni vagy regisztrálni.   
A regisztrációnál az összes mező kötelezően kitöltendő, a felhasználó meg kell adja a vezetéknevét, keresztnevét, egy felhasználónevet, jelszót, e-mail-címet, telefonszámát, megyéjét, és hogy szakmunkás fiókot szeretne-e.    
A regisztrációs mezők kitöltése után az oldal magától átirányít a bejelentkező felületre.  
Sikeres regisztráció és sikeres bejelentkezés esetén zöld szöveggel kiírjuk a képernyőre.  
A szakmunkásként regisztrált fiók sajátossága, hogy posztolási engedélyük van, így tudnak magukról hirdetést létrehozni.   
Bejelentkezés után a navigációs sávon található profil ikonon belül a Posztjaim-ra kattintva lehet új posztot létrehozni, valamint a saját posztjaimat megnézni (ha már van/vannak).   
Új poszt létrehozásánál meg kell adni vezeték és keresztnevet, a posztnak kell adni egy címet, legördülő listából ki kell választani a megyét, meg kell adni egy telefonszámot és a munka kategóriáját. Egy kinyíló naptárból ki kell választani a poszthoz egy dátumot, ezen kívül egy rövid leírást, ami minél többet magyaráz majd az érdeklődőknek! Az utolsó kitöltendő mező a fénykép feltöltése, ami majd szintén a felbérlők érdeklődését kelti fel.   
A mezők kötelezően kitöltendőek. Az összes mező kitöltése után a „Poszt létrehozása” gombbal az adatok bekerülnek az adatbázisba és a poszt létrejött.   
A létrejött poszt megtekinthető a navbar-on található posztok gombra kattintva, vagy a navbar-on profil ikonra, majd a posztjaim gombra kattintva. A posztot a törlés gombra kattintva lehet törölni.  
   
Egy másik felhasználó által létrehozott posztra lehet

* üzenetet küldeni a felhasználónak az „Írj rám” gombbal
* időpontot foglalni az „Időpont foglalás” gombbal
* szöveges véleményt hagyni
* 1-tól 5-ig terjedő skálán csillagokkal értékelni a posztot
* „Lementeni” a posztot a kedvencek közé a csillag ikonnal

Ha egy másik felhasználó üzenetet küld neked, akkor a profilon belül az Üzenetek gombra kattintva lehet megnézni, esetleg választ írni.  
Ha időpontkérelmet kapunk, azt 2 féle módon látjuk:

* navbar-on a csengő felett piros pont jelzi a bejövő kérelmet. Erre kattintva átirányít az időpont kérések oldalra
* a profilon belül a fiók beállítások, majd az Időpont kérések menüpontnál nézhetjük meg.

## A felhasznált ismeretek

Frontend telepítése React keretrendszerben JavaScript nyelven

Backend telepítése server.js használatával

Segédmodulok telepítése például Bootstrap, React-Router-Dom, Axios

Adatbázis létrehozás, szerkesztés és kezelés SQL nyelven a MySQL WorkBench-ben

Bootstrap Grid System felosztás különböző frontend elemek rendezéséhez

Flexbox CSS formázás

Különböző oldalak és modulok exportálása és importálása (például a css stíluslapok és a fényképek használatához)

React-Router route-ok beállítása az oldalak közötti „lapozáshoz”

## A felhasznált szoftverek

Visual Studio Code (és az ebbe beleépített mesterséges intelligencia)

MySQL WorkBench

XAMPP Control Panel

Microsoft Word

Microsoft Teams

PowerPoint

GitHub Desktop

Discord

# Felhasználói dokumentáció

## A program általános specifikációja

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.

Az S.O.S. munka egy gyors, egyszerű szakmunkás hirdető és kereső weboldal. A felhasználók regisztrálnak felbérlőként vagy munkásként, lehetőségük van hirdetéseket létrehozni, valamint hirdetések között böngészhetnek. A hirdetéseken keresztül lehet üzenetet küldeni, időpontot foglalni, valamint értékelést hagyni.  
  
A navigációs sávon lévő profil ikonra kattintva lehet bejelentkezni vagy regisztrálni.   
A regisztrációnál az összes mező kötelezően kitöltendő, a felhasználó meg kell adja a vezetéknevét, keresztnevét, egy felhasználónevet, jelszót, e-mail-címet, telefonszámát, megyéjét, és hogy szakmunkás fiókot szeretne-e.    
A regisztrációs mezők kitöltése után az oldal magától átirányít a bejelentkező felületre.  
Sikeres regisztráció és sikeres bejelentkezés esetén zöld szöveggel kiírjuk a képernyőre.  
A szakmunkásként regisztrált fiók sajátossága, hogy posztolási engedélyük van, így tudnak magukról hirdetést létrehozni.   
Bejelentkezés után a navigációs sávon található profil ikonon belül a Posztjaim-ra kattintva lehet új posztot létrehozni, valamint a saját posztjaimat megnézni (ha már van/vannak).   
Új poszt létrehozásánál meg kell adni vezeték és keresztnevet, a posztnak kell adni egy címet, legördülő listából ki kell választani a megyét, meg kell adni egy telefonszámot és a munka kategóriáját. Egy kinyíló naptárból ki kell választani a poszthoz egy dátumot, ezen kívül egy rövid leírást, ami minél többet magyaráz majd az érdeklődőknek! Az utolsó kitöltendő mező a fénykép feltöltése, ami majd szintén a felbérlők érdeklődését kelti fel.   
A mezők kötelezően kitöltendőek. Az összes mező kitöltése után a „Poszt létrehozása” gombbal az adatok bekerülnek az adatbázisba és a poszt létrejött.   
A létrejött poszt megtekinthető a navbar-on található posztok gombra kattintva, vagy a navbar-on profil ikonra, majd a posztjaim gombra kattintva. A posztot a törlés gombra kattintva lehet törölni.  
   
Egy másik felhasználó által létrehozott posztra lehet

* üzenetet küldeni a felhasználónak az „Írj rám” gombbal
* időpontot foglalni az „Időpont foglalás” gombbal
* szöveges véleményt hagyni
* 1-tól 5-ig terjedő skálán csillagokkal értékelni a posztot
* „Lementeni” a posztot a kedvencek közé a csillag ikonnal

Ha egy másik felhasználó üzenetet küld neked, akkor a profilon belül az Üzenetek gombra kattintva lehet megnézni, esetleg választ írni.  
Ha időpontkérelmet kapunk, azt 2 féle módon látjuk:

* navbar-on a csengő felett piros pont jelzi a bejövő kérelmet. Erre kattintva átirányít az időpont kérések oldalra
* a profilon belül a fiók beállítások, majd az Időpont kérések menüpontnál nézhetjük meg.

## Rendszerkövetelmények

### 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.

Processzor: Intel Core i5-2400 3.10GHz 4 szál 4 mag

8GB DDR3 1333 MHz

Seagate Barracuda HDD 250GB 7200.12

### 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

A program Windows 10 Home és Windows 11 operációs rendszeren volt tesztelve  
A program futtatásához ezek a szoftverek szükségesek:

* GitHub Desktop
* Visual Studio Code
* XAMPP Control Panel
* MySQL WorkBench
* bármilyen böngésző (Google Chrome-ot ajánlok!) és internetkapcsolat

## 3. A program telepítése

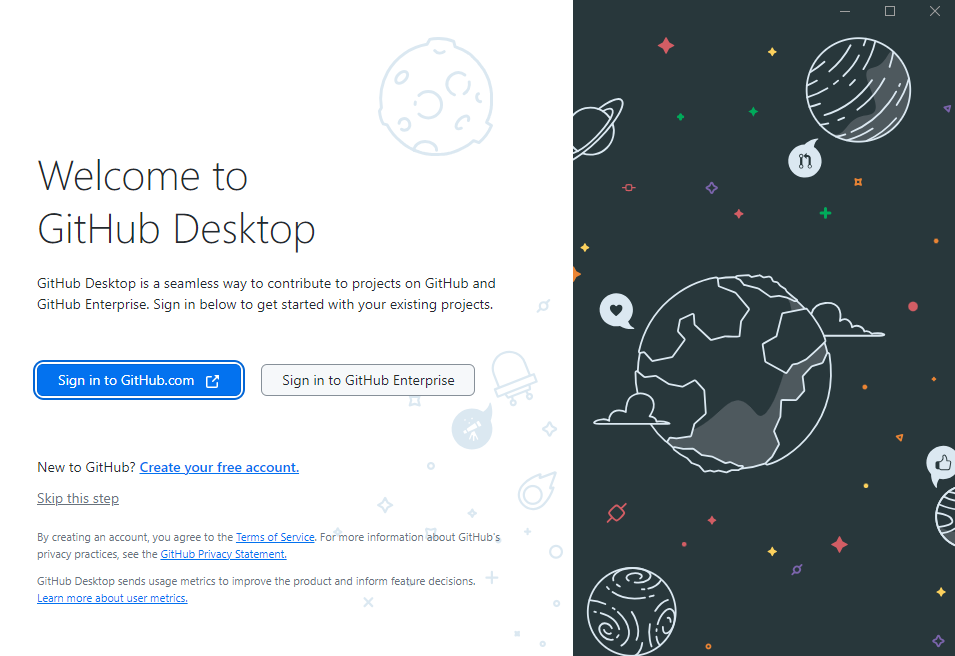
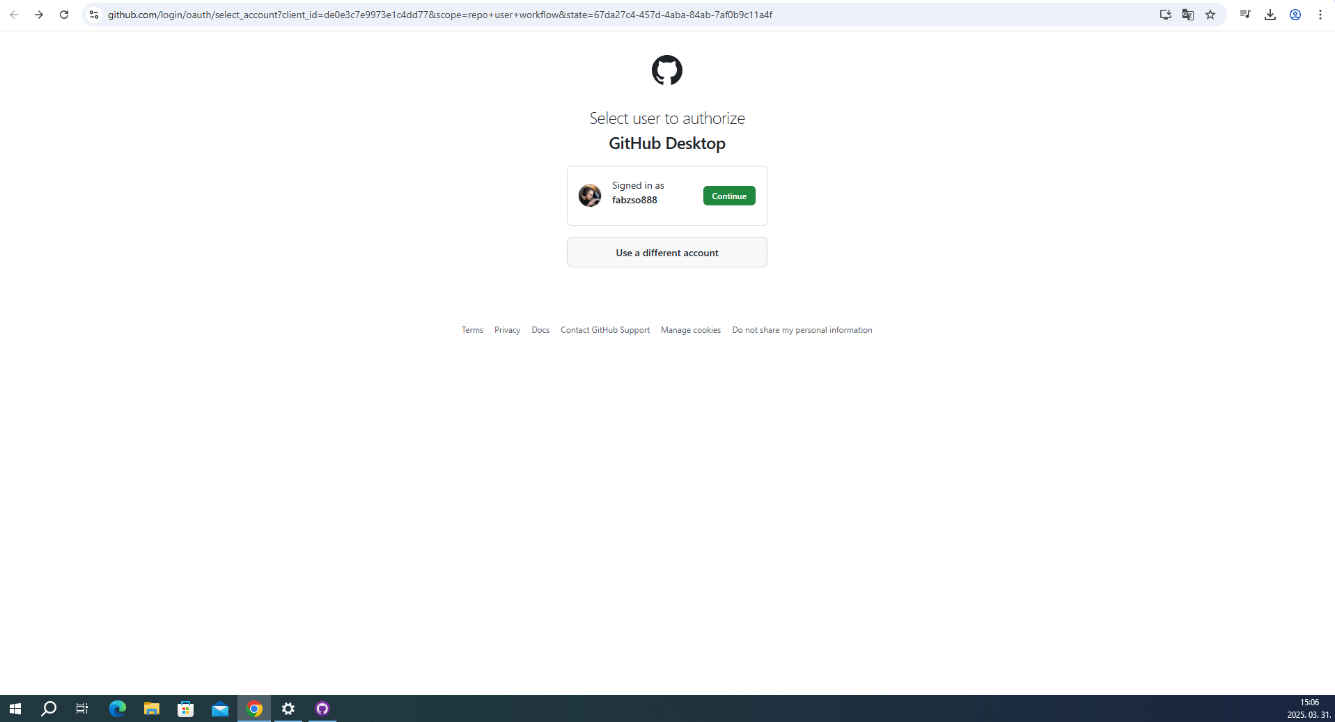
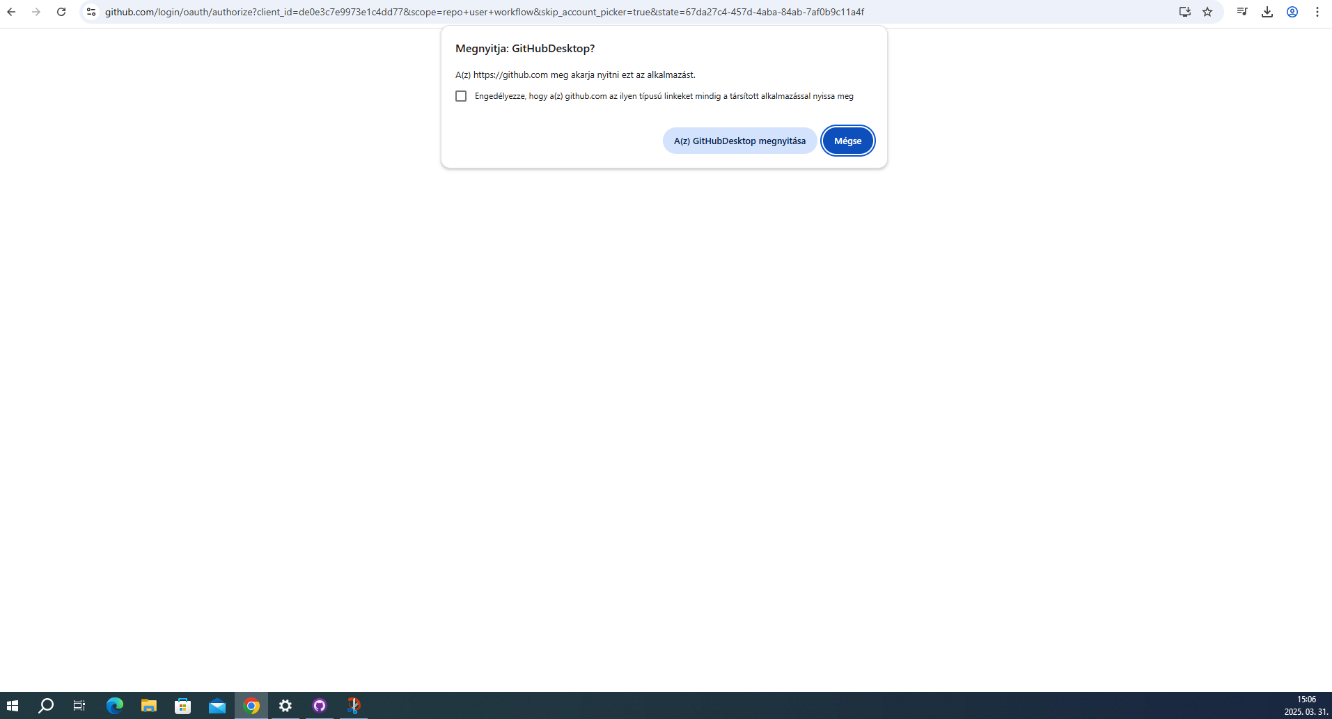
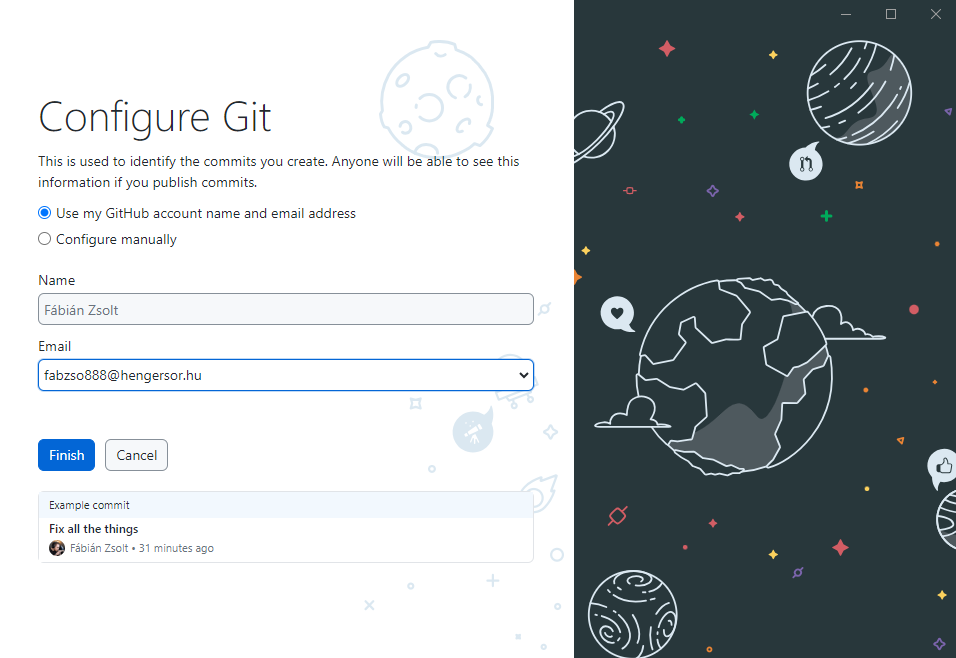
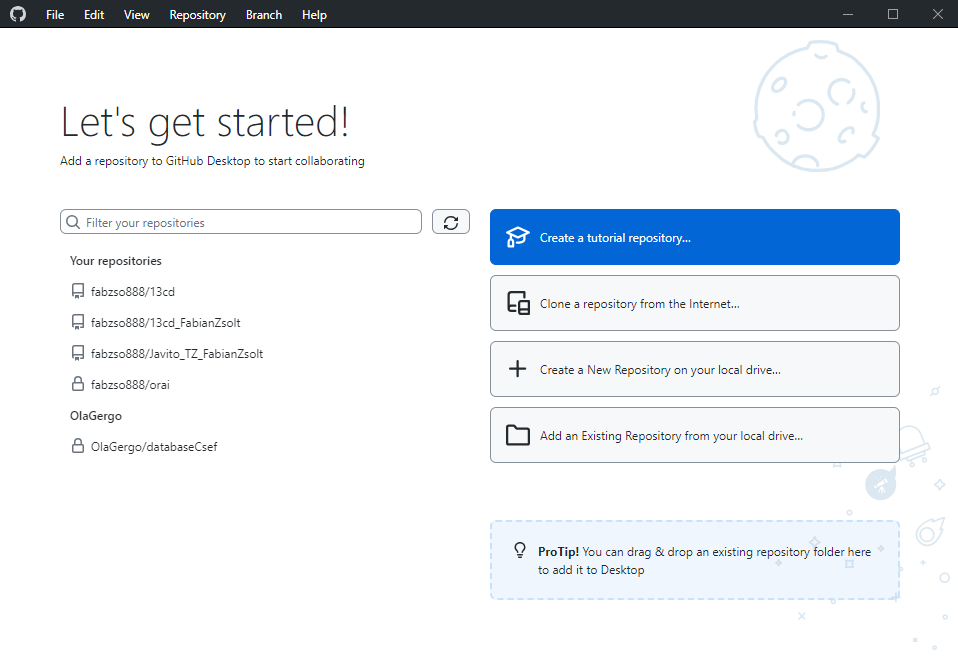
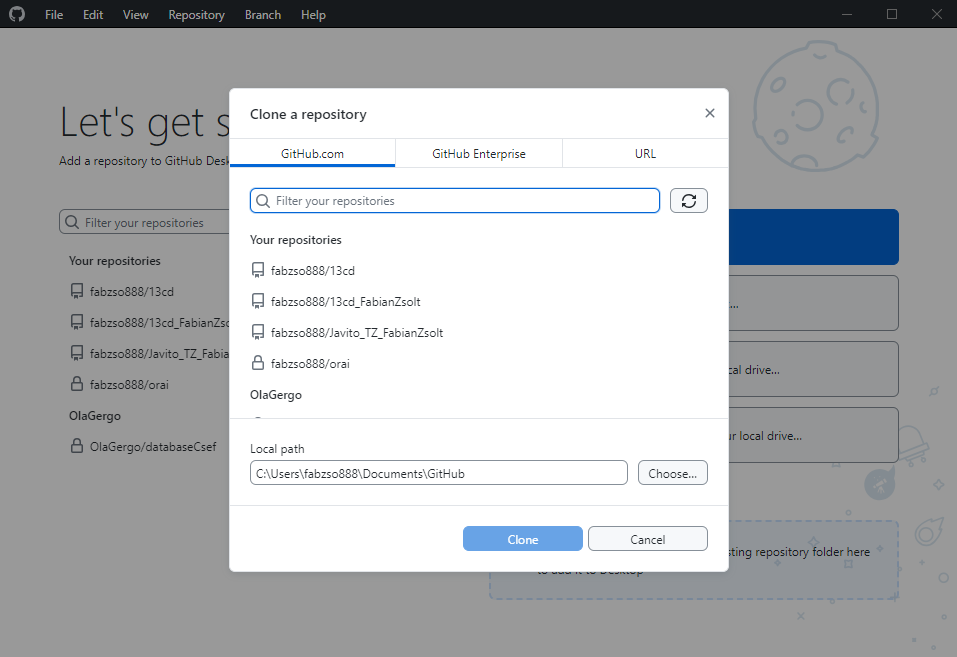
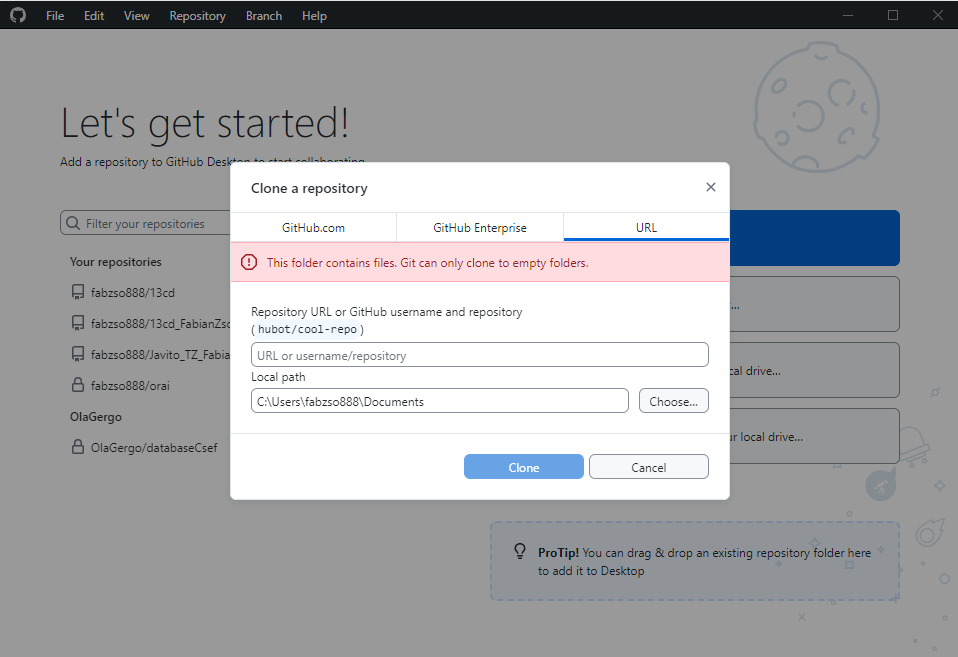
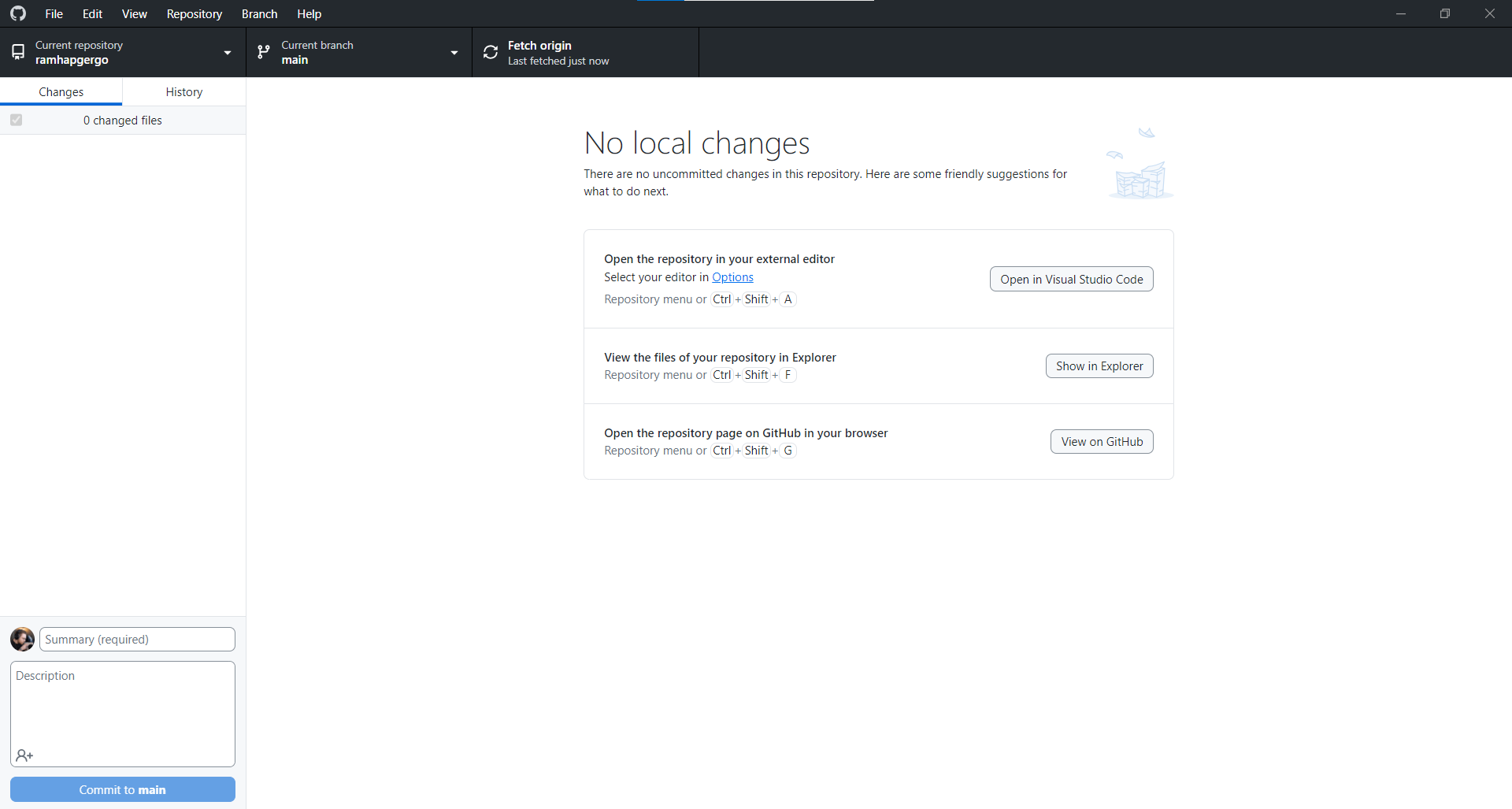
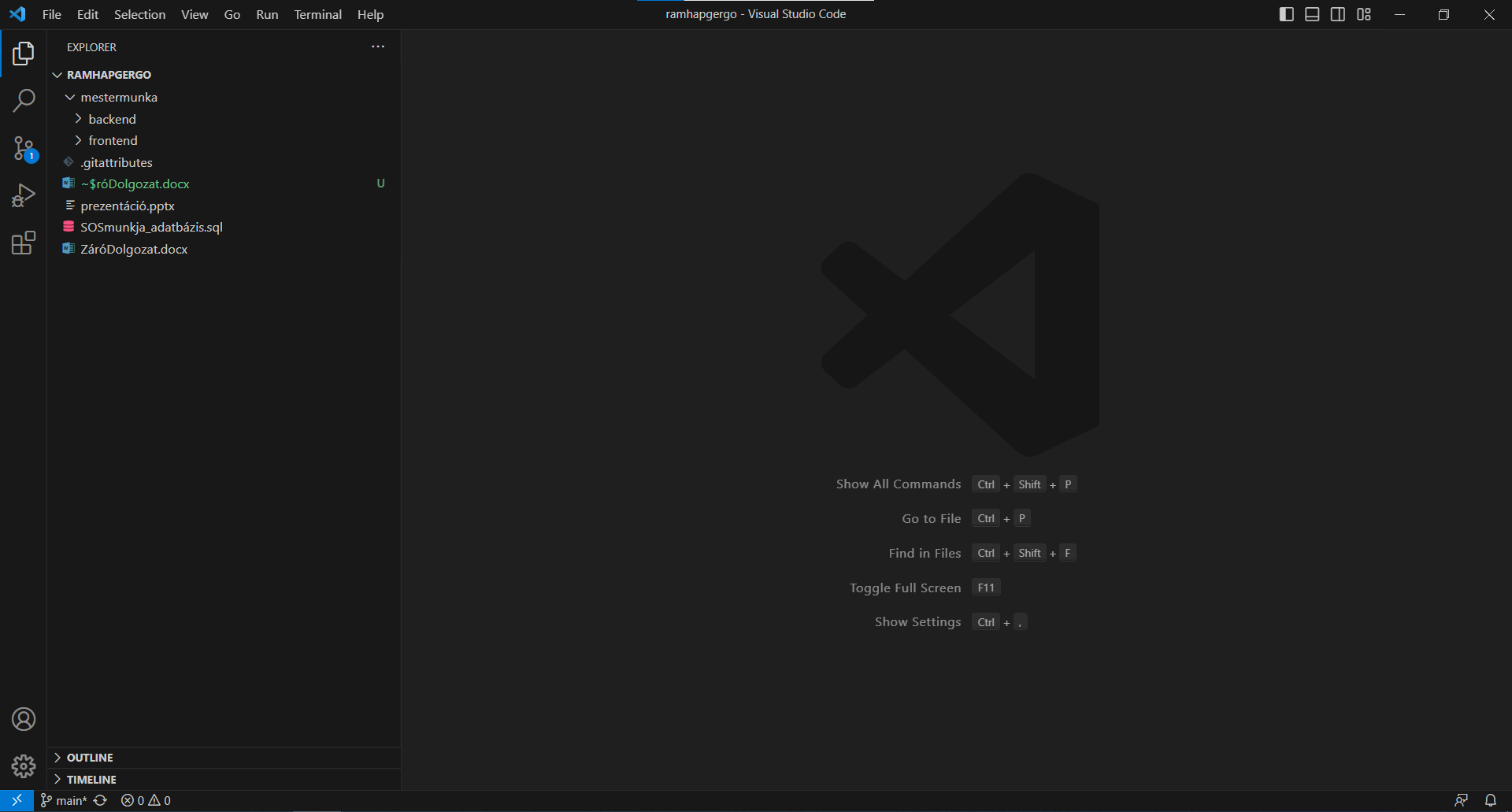
Képekkel illusztrált, részletes leírás a program telepítésének a menetéről.

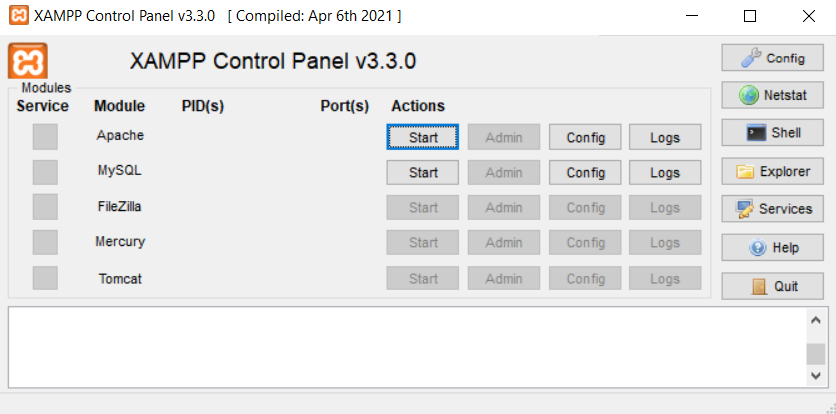
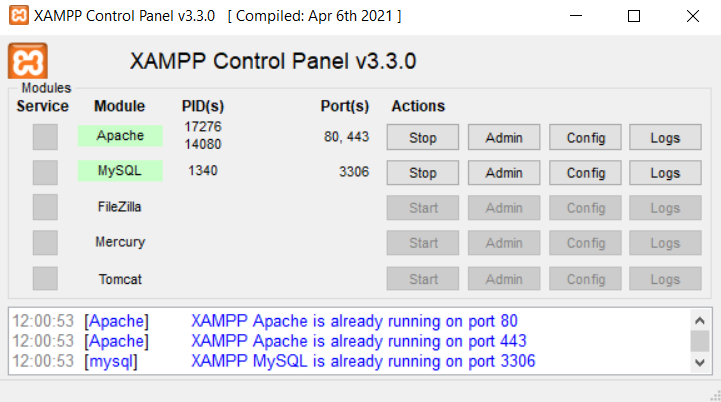
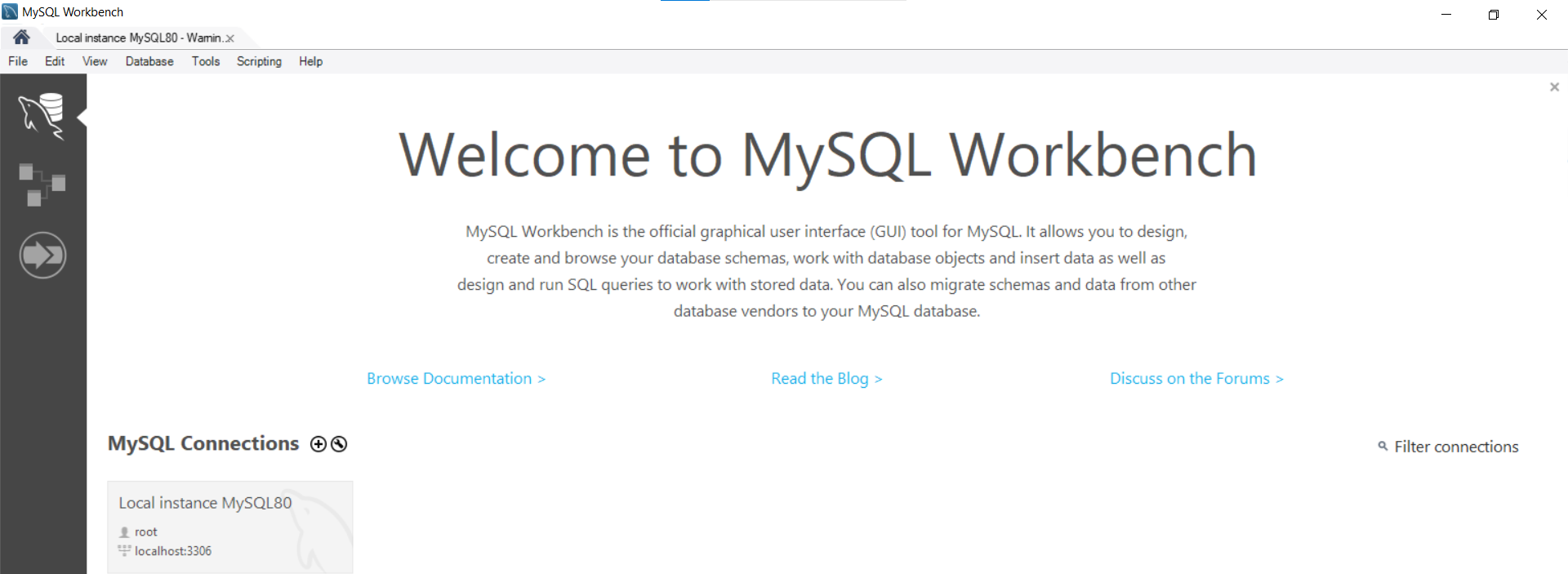
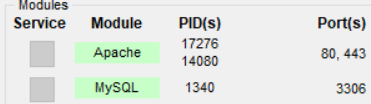
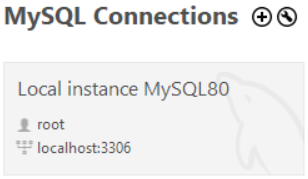
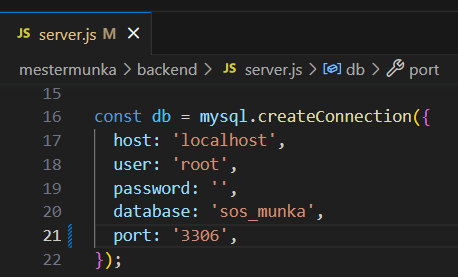
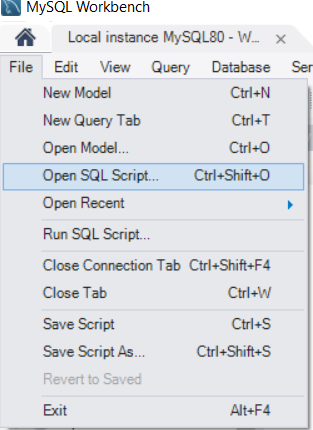
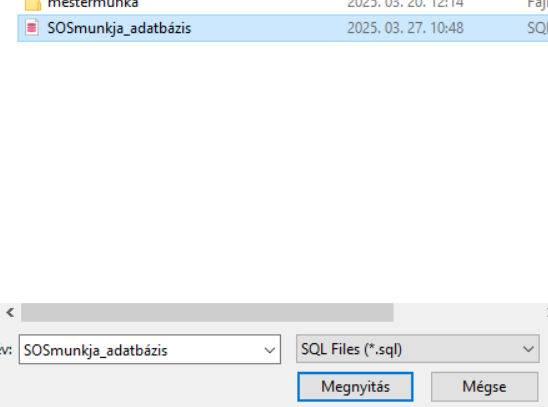
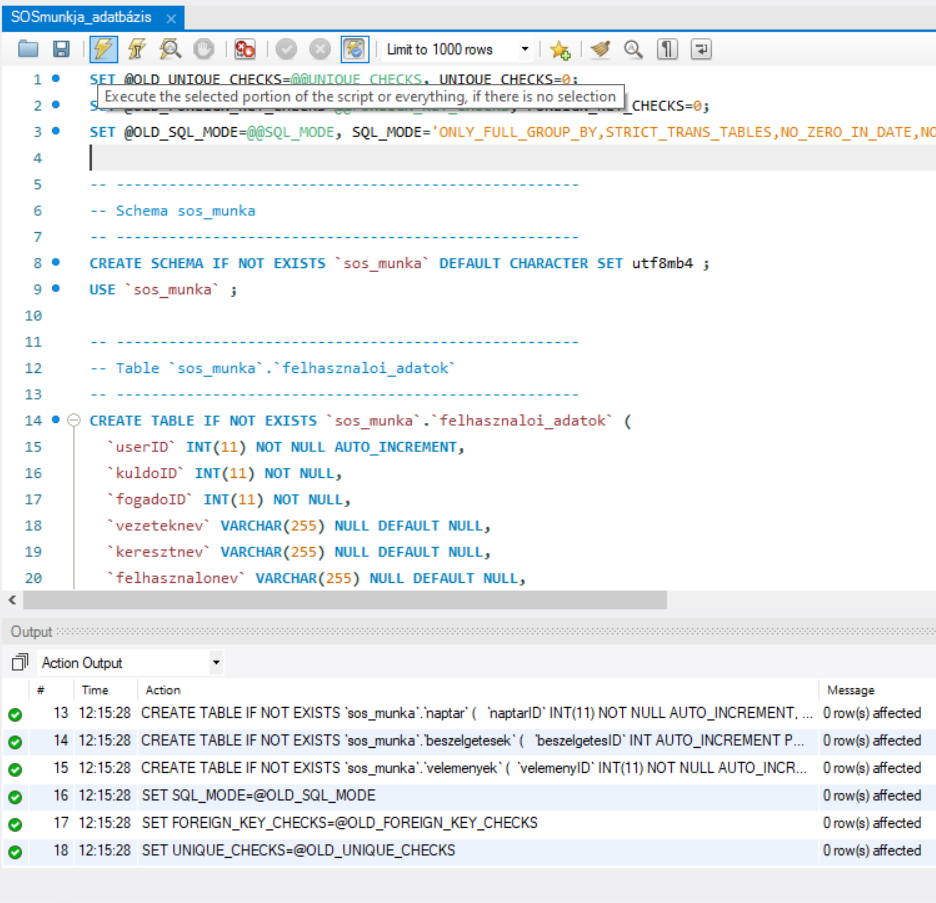
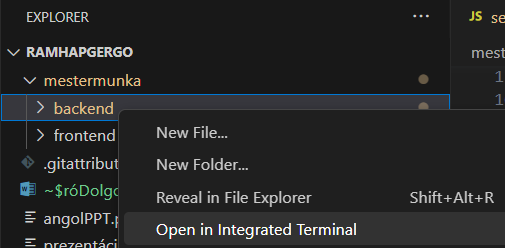
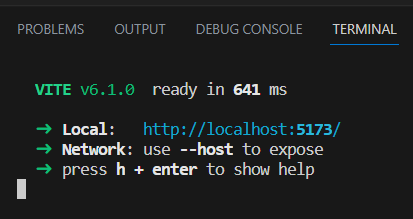
* A leírás alapján a felhasználónak hiba nélkül telepíteni kell tudni a programot. A leírásnak ki kell térnie a telepítés során kiválasztható opciókra is.
* Ha esetleg nincs telepítőprogram, akkor kellő részletességgel le kell írni, hogy mely fájlokat, pontosan hova kell felmásolni, és hogy lehet a programot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

1. kép Ez a módosító ablak a programban

Ajánlott terjedelem: 2 -4 oldal, ábrákkal együtt.  
A program telepítéséhez szükség lesz a következő alkalmazásokra:

* GitHub Desktop <https://desktop.github.com/download>
* Visual Studio Code <https://code.visualstudio.com/Download>
* XAMPP Control Panel <https://www.apachefriends.org/download.html>
* MySQL WorkBench <https://dev.mysql.com/downloads/workbench>

A linkeken keresztül telepítsük őket.  
  
Futtassuk a GitHub Desktop-ot.  
  
Ha még nincs GitHub fiókunk, nyomjunk rá a „Create your free account.” kék szövegre majd a regisztrációs mezőket kitöltve regisztrálunk egy fiókot.  
Ha már van GitHub fiókunk, akkor a kék „Sign in to GitHub.com” gombra rányomunk, majd a böngészőben megnyíló ablakban bejelentkezünk.  
  
Itt a zöld „Continue” gombra nyomunk, majd az oldal megkérdezi hogy átirányíthat-e a GitHub Desktop alkalmazásba.  
  
Itt a világoskék „A(z) GitHub Desktop megnyitása gombra nyomunk, majd megnyílik az alkalmazás. Ekkor ezt az ablakot látjuk:  
  
Érdemes a képernyőn látható beállításokat használni. (Természetesen itt a saját E-mail címedet és nevedet fogod látni.)  
A kék „Finish” gombra nyomunk, majd átirányít a „Let’s get started!” oldalra  
 Itt a jobb oldalon lévő szürke „Clone a repository from the Internet…” gombot választjuk.  
Itt pedig a jobb oldalt lévő „URL” gombra nyomunk.  
A felső kitöltendő mezőbe a GitHub Repository linkjét kell beilleszteni.  
Az alsó mezőbe pedig kiválasztjuk hogy a számítógépen hova szeretnénk menteni.  
Végül a kék „Clone” gombra nyomunk és a GitHub Repository átmásolódik a kiválasztott tárhelyre.  
Ekkor ezt kellene látnunk az alkalmazásban:  
  
Az átmásolt fájlok megtekintéséhez nyomjunk rá a szürke „Open in Visual Studio Code” gombra  


Most futtassuk az XAMPP Control Panel alkalmazást  
  
Nyomjuk meg a „Start” gombot az Apache sorban és a MySQL sorban  
  
Futtassuk a MySQL WorkBench-et  
  
Fontos hogy az XAMPP alkalmazásban a MySQL port egyezzen a MySQL szerver portjával és a leklónozott GitHub repository backend mappájában lévő server.js fájl portjával, és a többi csatlakozási adat is egyezzen egymással.  
  
A MySQL WorkBench-ben duplaklikkel csatlakozzunk a lokális szerverre.  
A bal felső sarokban a „File” és „Open SQL Script”  
   
A megnyíló ablakban nyissuk meg a mappát ahova klónoztuk a GitHub Repository-t és válasszuk ki az SOSmunkja\_adatbázis fájlt. Majd a „Megnyitás” gombra nyomunk.  
A bal oldalt lévő sárga villámmal lefuttatjuk az adatbázist létrehozó scriptet. Ha jól csináltuk a képernyő alján zöld pipáknak kéne lenni  
  
Menjünk vissza a Visual Studio Code alkalmazásba és itt jobbklikkel rányomunk a backend mappára. Ekkor ezeket az opciókat kapjuk:  
  
Az „Open in Integrated Terminal” gombra nyomunk. Ez a képernyő alján megnyit egy terminál ablakot. Azt írjuk be hogy „npm i” és a billentyűzeten enterrel futtatjuk. Utána „npm run dev” parancsot írjuk be és enterrel futtatjuk.  
Ezután a frontend mappára is jobbklikk. „Open in Integrated Terminal” majd az „npm i” és „npm run dev” parancsokat futtatjuk.  
  
A terminál ablakban megjelenő linket nyissuk meg.  
Az oldal használatra kész!

## A program használatának a részletes leírása

Mindenre kiterjedő, részletes leírás a program használatáról. Alapszabályok:

* 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.

A screenshot of a computer

AI-generated content may be incorrect.

A főoldalon 6 animált elemünk van:

* A felül látható navigációs sávon a kis logónk. Ez az elem egy <NavLink>-en belül egy <img>. Az img-nek a logo-img1 osztálynév van megadva, amit a Navbar.css stíluslappal van rárakva az animáció: animation: float 3s infinite ease-in-out;
* A középen látható cím és alcím <motion.div> elemekkel van meganimálva. A Home.jsx fájlban vannak megadva a kezdő pozíciójuk és a végső pozíciójuk „kordinátái”. A <motion.div> elemeken belül <p> paragrafus elemekben van megírva a szöveg, és <h2> valamint <h4> elemeket használunk a címek kiírásához.
* Az alul lévő 3 szürke doboz szintén <motion.div> elemek, benne <h1> tartalmazza a címet, <p> tartalmazza a szöveget.

A főoldalról a felül látható navigációs sávon lévő elemek kattinthatók.

A posztok feliratra kattintva a React-Router átirányít a Posztok.jsx oldalra. Itt láthatod a sikeresen létrehozott/feltöltött posztokat.

A posztok oldalon a szűrő segítségével tudsz a posztok között tetszés szerint keresni, szűrni, kilistázni.  
A lenyíló Kategória és Megye menüből választható opciók lokális változóból (tömbből) töltődnek be. A Posztok.jsx fájl tetején const categories =[ ] és const locations = [ ].

A kedvenc csillag kattintható, „filled” vagy „” tulajdonsága van, ez kattintásra változik. A tulajdonsággal együtt változik a színe „gold”-ról „gray”-re (arany és szürke).

A keresőmezőbe kézzel írható szöveg pedig a poszt címét vagy a posztot létrehozó felhasználó nevét vizsgálja. A nagyító gombra kattintva lépnek érvénybe a beállított szűrők, vagy egyből a kedvenc csillagra kattintva egyből frissül.  
A screenshot of a computer

AI-generated content may be incorrect.

Így néz ki a posztok oldal, és látszik hogy a szűrő tökéletesen működik.

Itt hagytam abba\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# 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.

* Microsoft Word (a dokumentáció megírásához)
* Visual Studio Code (a React és Javascript telepítéséhez, a HTML, CSS és JavaScript kód fejlesztéséhez)
* MySQL Workbench (az adatbázis létrehozásához)
* framebox (a felhasználói felület megtervezéséhez)
* draw.io (az egyed-kapcsolat diagramhoz és az adatbázis megtervezéséhez)

## 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.

Az adatbázisunk neve „*s.o.s. munka*”

A csevegési funkció működéséhez van a „beszelgetesek” tábla.

* beszelgetesID – int(11) Auto Increment, Primary Key: két felhasználó közötti beszélgetés egyedi azonosítója
* feladoID – int(11) Egyedi azonosító ami a jelenlegi felhasználót összekötteti a kiválasztott címzettel az üzenetküldéshez
* cimzettID – int(11) Egyedi azonosító ami az üzenetet fogadó felhasználó kap
* tartalom – text – Ebben tároljuk magát az üzenetet
* kuldesIdopont – datetime – Az üzenetek elküldésének idejét tároljuk el
* olvasott – tinyint(1) Minden üzenetre külön állapot van hogy olvasott / olvasatlan. Ha az állapot olvasatlan, azt kiírja az oldal  
  Ez a funkció egyenlőre nem működik

A posztokat értékelni lehet egy 1-tól 5-ig terjedő skálán csillagokkal. Ennek az adatait az „ertekelesek” táblában tároljuk.

* ertekelesID – int(11) Auto Increment, Primary Key – Automatikusan generált egyedi azonosító amely minden egyes értékeléshez van
* post\_id – int(11) Egyedi azonosító, az adott értékelés melyik poszthoz tartozik
* user\_id – int(11) Egyedi azonosító, az adott értékelést melyik felhasználó adta
* rating – int(11) 5-ből hány csillaggal értékelte a felhasználó a posztot
* created\_at – datetime – Időadat, mikor lett értékelve a poszt

A felhasználói adatok tárolására van a „felhasznaloi\_adatok” nevű táblánk.

* userID - int(11) Auto Increment, Primary Key: egyedi azonosító, amit a háttérben legenerálunk a felhasználónak
* kuldoID – int(11) Az üzenetek küldéséhez szükséges azonosító
* fogadoID – int(11) Az üzenetek fogadásához szükséges azonosító
* vezeteknev – varchar(255) A felhasználó vezetéknevét tartalmazza
* keresztnev – varchar(255) A felhasználó keresztnevét tartalmazza
* felhasznalonev – varchar(255) A felhasználó által választott felhasználónév
* jelszo – varchar(255) A felhasználó által választott jelszót tárolja bycrypt titkosítással
* emailcim – varchar(255) A felhasználó email-címe
* telefonszam – varchar(20) A felhasználó telefonszámát tárolja
* telepules – varchar(255) A felhasználó települése
* letrehozasDatum – date ; A felhasználói profil létrehozásának ideje
* modositasDatum – date ; A felhasználói profil szerkesztésének ideje
* admin – bool / tinyint(1) A felhasználónak van admin joga / nincs
* munkasreg – bool / tinyint(1) A felhasználó szakmunkás profilként regisztrált-e
* profilkep – varchar(255) A felhasználó profilképének fájlnevét tárolja.
* lastActive – datetime – Azt tárolja a felhasználó mikor volt utoljára elérhető

A kategóriák tárolására van a „kategoriak” nevű táblánk

* kategoriID – int(11) Auto Increment, Primary Key ; A kategóriáknak azonosító
* megnevezes – varchar(255) A munka típusát tartalmazza

A kedvenc posztok tárolására van a „kedvencek nevű táblánk.

* kedvencID – int(11) Primary Key
* userID – int(11) A felhasználó azonosítója aki becsillagozta a posztot
* posztID – int(11) Annak a posztnak az azonosítója ami be lett csillagozva
* datum – date ; Az a dátum amikor be lett csillagozva a poszt

A posztra foglalt időpontokat a „naptár” táblában tároljuk.

* naptarID – int(11) Auto Increment, Primary Key
* posztID – int(11)
* userID – int(11)
* nap – varhar(20)
* ora – varchar(5)
* foglalasDatum - datetime

## 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.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tesztesetek | Tesztelő | Tesztlépés leírása | Elvárt eredmény | Hibás eredmény | Megjegyzés | Dátum | Javítva? |
| Adatbázis kapcsolódás | Rámháp Gergő | a server.js fájlba létrehozni az SQL kapcsolatot, majd console log-gal kiíratni egy tesztadatot | A console-ban megkapjuk a sikeres kapcsolódás üzenetet és az adatot is megjeleníti | Hibátlan volt |  | 2020.02.20. | Nem volt szükség javításra |
| Regisztráció | Rámháp Gergő | Létrehoztunk egy Regisztráció.jsx fájlt. Először a beviteli mezőkbe lévő adatokat lokális változókban tároljuk, majd sikeres regisztrációnál Axios.post művelettel feltöltjük az adatbázisba | Az adatbázis kapcsolat sikeres és az adatok a küldés gombbal feltöltődnek az adatbázisba | Hibátlan volt | A bejelentkezést is Axios.post művelettel fogjuk megírni.  Szeretnénk ha sikeres regisztráció után átirányít a bejelentkező felületre | 2025.02.20. | Nem volt szükség javításra |
| Bejelentkezés | Rámháp Gergő | Bejelentkezes.jsx ugyanazzal a logikával működik mint a regisztráció | Ha a bejelentkezési adatok megegyeznek a regisztrációs adatokkal akkor sikeresen bejelentkezik | Hibátlan | A bejelentkezés működik Axios.post-tal | 2025.02.20. | Nem volt szükség javításra |
| Posztok | Wágner János |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

# 

# Összefoglalás

## Önértékelés

***Kitűzött célok elérése***

A csapatnak volt közös kitűzött célja: hogy egy működő weboldalt hozzunk létre a vizsgakövetelményeket elérve, határidőn belül.

Büszkék vagyunk a projektünkre, a projekt végtermékre, a csapatmunkára és egymásra.

A projekt közben felmerült problémák leginkább a GitHub Desktop-pal voltak és a MySQL Workbench-csel.

Néha a commit-olás (fájlok feltöltése, felülírása) rosszul sült el. Például: egy egész napnyi munka elveszett, felülíródott. A server.js fájlban párszáz sor eltűnt.

A MySQL Workbench iskolai hálózaton nem engedett módosításokat végezni (nem volt felhasználói engedély). Egyed-kapcsolat diagramot nem tud legenerálni az adatbázisunkról.

***Önértékelés***

Fábián Zsolt Ferenc:  
A projekt ideje alatt megtanultam hogyan kell társaimmal hatékonyan kommunikálni, fejlődött az előadási stílusom, valamint tanultam a React keretrendszer felállításáról és használatáról. A CSS tudásomat pedig frissítettem, gyakoroltam a weblapok és elemek formázását. A GitHub Desktop-ról és a verziókezelésről is tanultam. A commitolás és a régebbi munka megtekintése jobban megy.

Wágner János:

A projekt ideje alatt fejlődött a kommunikációs képességem, én is megtanultam a React keretrendszert telepíteni és használni. Rengeteg új parancsot tanultam CSS kódolás közben. Én is jobban megtanultam a GitHub Desktop-ot kezelni.

Rámháp Gergő:

Megtanultam SQL kéréseket írni, adatbázis kapcsolatot létrehozni, adatokat feltölteni, JSON token-t használni és alkalmazni a projektben. GitHub repository-t létrehozni és kezelni. Én voltam úgymond a „csapat vezére”, ellenőriztem csapattársaim commit-jait és munkáját. Ha elakadtak, segítettem nekik. Szerintem a munka gördülékenyen ment, eredményes projekt volt.

## 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

Ötletek, amelyeket meg akartunk valósítani, de nem sikerült / nem fért bele az időbe:

* Poszt létrehozásnál több kép feltöltése, és szép megjelenítése
* Poszt szerkesztése (nem volt rá idő)
* A posztok oldalon keresztül a saját posztodra tudsz szöveges véleményt hagyni
* A posztok oldalon keresztül a saját posztodra tudsz csillagos értékelést adni
* A posztok oldalon keresztül a saját posztodra tudsz időpontot foglalni

# 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.

Fórum ahonnan a drótváz készítő programra találtunk rá:  
<https://kisshonlapkeszites.hu/3-hasznos-drotvazkeszito-online-szoftver>  
Legutóbbi elérés ideje: 2025.02.03. (Hétfő) 15:06

Drótváz készítő program:  
http://framebox.org/  
Legutóbbi elérés ideje: 2025.02.03. (Hétfő) 15:07

Adatbázis és egyed kapcsolat tervező program:  
https://app.diagrams.net/  
Legutóbbi elérés: 2025.02.03. (Kedd) 13:14

Bootstrap Grid System felosztás  
<https://www.w3schools.com/bootstrap/bootstrap_grid_system.asp>  
Legutóbbi elérés: 2025.02.06. (Csütörtök) 11:24

# Ábrajegyzék

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