BGSzC Pestszentlőrinci Közgazdasági és Informatikai Szakgimnáziuma

1184 Budapest Hengersor 34.

1. Záró dolgozat

Online kódmegosztó

Konzulens tanár: Készítette:

Péter-Szabó Richard Czékus Máté

Tartalom

[1 Bevezetés 3](#_Toc529273086)

[1.1 Feladat leírás 3](#_Toc529273087)

[1.1.1 Általános ismertetés 3](#_Toc529273088)

[1.2 A felhasznált ismeretek 3](#_Toc529273089)

[2 Felhasználói dokumentáció 4](#_Toc529273090)

[2.1 A program általános specifikációja 4](#_Toc529273091)

[2.2 Rendszerkövetelmények 4](#_Toc529273092)

[2.2.1 Hardver követelmények 4](#_Toc529273093)

[2.2.2 Szoftver követelmények 4](#_Toc529273094)

[2.3 3. A program telepítése 4](#_Toc529273095)

[2.4 A program használatának a részletes leírása 5](#_Toc529273096)

[3 Fejlesztői dokumentáció 6](#_Toc529273097)

[3.1 Az alkalmazott fejlesztői eszközök 6](#_Toc529273098)

[3.1.1 Adatmodell leírása 6](#_Toc529273099)

[3.2 Részletes feladatspecifikáció, algoritmusok 6](#_Toc529273100)

[3.3 Tesztelési dokumentáció 7](#_Toc529273101)

[3.4 7. Továbbfejlesztési lehetőségek 7](#_Toc529273102)

[4 Felhasznált irodalom 8](#_Toc529273103)

# 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.

### Általános ismertetés

Ez a program azért készült mert a már piacon lévő versenytársai nem rendelkeznek elég funkcióval illetve elég intuitív kezelőfelülelttel az egységsugarú fellhasználók számára. A nehéz kezelhetőségük mellett a publikálás ezen oldalakon elég nehézkes, emlette a szintaxis kiemelésük sem túl kiemelkedő. Az általam készített program erre a problémára reflektálva próbál megoldást nyújtani intuitív és modern kezelőfelületével, kiemelt szempontban tartva az online magánszférát illetve az névtelen és nevesített kód publikációt. Emellett magán, privát kódok megosztására is kiváló. Üzemeltetési szempontból, előtérben tartva a legújjabb trendeket felhő környezetre felkészítve, könnyen skálázható eldobható egységekből épül fel, emellett egy előre elkészített konfiguráció miatt egy átlag felhasználó is könnyen felépítheti a weboldalhoz kellő speciális környezetet, melyben futtathatja az oldalból készült saját replikáját. Amely a kódom licensz feltételei alapján szabadon megtehető.

## A felhasznált ismeretek

Front-end tervezés https://github.com/thedaviddias/Front-End-Checklist

Ikonkészlet és használata https://google.github.io/material-design-icons/

MongoDB kezelés PHP-ban http://zetcode.com/db/mongodbphp/

Materialize keretrendszer https://materializecss.com/

MongoDB ismeretek https://www.mongodb.com/

Docker https://docs.docker.com/

Verziókezelés https://git-scm.com/

MySQL ismeretek https://dev.mysql.com/doc/

Syntax highlight https://highlightjs.org/

Webszerver dokumnetáció <https://nginx.org/en/docs/>

# Felhasználói dokumentáció

## A program általános specifikációja

Ez a szoftver kiváló azon felhasználók számára, akik programozással, szoftverfejlesztéssel foglalkoznak és bárki másnak akik számára kimondottan fontos nagy mennyiségű szöveg gyors és hatékony továbbítása más felhasználók részére. A szoftverfejlesztőknek és programozók számára azért ajánlom, mert a kód kiemelés funkciója kimondottan felhasználóbaráttá és ergonómikussá teszi a megosztott tartalom áttekintését és az azzal továbbiakban történő szoftverevolúciós folyamatokat és elősegíti a fejlesztők közötti produktív együttműködést. Ezen felül tökéletesen szolgál egyszerű pár soros kódok, vagy akár csak egyszerű szövegek/szövegrészletek nyilvánossá tételéhez. Mindezek mellett ez a széles funkcionalitás nem jár a személyes adataink és magánszféránk kockáztatásával, hiszen az oldal használatához nem kötelező a regisztráció így mint névtelen felhasználó is létrehozhatunk szöveg vagy kódrészleteket az oldalon, azonban ha mégis a regisztrációt mellett döntenénk ez esetben sem kell lemondanunk privátszféránk védelméről, hiszen a regisztrációhoz mindössze egy felhasználónév és egy email cím szükséges a jelszón felül, a regisztrált profilunk ezen felül mivel privát profil mások által nem megtekinthető, egyedül a felhasználónév a publikációink alatt.

## Rendszerkövetelmények

### Hardver követelmények

Az általam készített szoftver, mivel egy felhőkörnyezetre felkészített alkalmazás így napjainkban használt bármelyik populáris operációs rendszeren működőképes a ezek elérik a következő minimális követelményeket:

Microsoft Windows-t használó számítógépek esetén:

-A számítógép BIOS rendszerében a Virtualizációnak bekapcsolva kell lennie (Virtualization: enabled)

-Legalább 4Gigabyte RAM( memória)

-64bit-es legalább Intel „Nehalem”, ”Westmere”, „Sandybridge” architektúrára épülő processzorok, illetve ettől újabb architektúrára épülő szériák.

-64bit-es legalább AMD „Barcelona” architektúrára épülő processzor vagy újabb.

-Legalább 2GB szabad hely a merevlemezen.

-Széles sávú internet hozzáférés

Apple macOS esetén:

-Legalább egy 2010-es vagy újabb Mac modell

-Legalább 4Gigabyte RAM (memória)

-Legalább 2GB szabad hely a merevlemezen.

-Széles sávú internet hozzáférés

Linux disztribúciót futtató számítógép esetén:

-A számítógép BIOS rendszerében a Virtualizációnak bekapcsolva kell lennie (Virtualization: enabled)

-Legalább 4Gigabyte RAM( memória)

-64bit-es legalább Intel „Nehalem”, ”Westmere”, „Sandybridge” architektúrára épülő processzorok, illetve ettől újabb architektúrára épülő szériák.

-Legalább 2GB szabad hely a merevlemezen.

-Széles sávú internet hozzáférés

### Szoftver követelmények

-Windows 10 64bit: Pro, Enterprise vagy Education (Ez esetben a buildszámnak legalább 14393-nak kell lennie vagy nagyobbnak)

-Legalább Mac OS Sierra vagy újabb operációs rendszer

-Legalább 64bit-es verziója az alábbi az alábbi disztribúcióknak:

CentOS 7

Debian 9 vagy Raspbian Stretch

Fedora 28

Ubuntu Xenial 16.04

Arch Linux

Ezeken felül:

Az operációs rendszernek megfelelő Docker CE vagy Docker EE szoftvercsomag, a program Docker Toolbox működhet de felléphetnek kisebb hibák, ezért ezt a verziót nem ajánlom.

## 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.

Ajánlott terjedelem: 2 -4 oldal, ábrákkal együtt.

Telepítés Windows operációs rendszer alatt:

Első lépésben kezdjünk a Docker nevű programmal hiszen ez elengedhetetlen az alkalmazás futtatásához, fontos feltétel, hogy a szoftverhez bekapcsolt állapotban kell lennie Windows operációs rendszerünkön a Hyper-V nevű virtualizációs eszköznek, ezt a Vezérlőpult → Programok és szolgáltatások → Windows-szolgáltatások be- és kikapcsolása menüpont alatt érhetjük el, bekapcsolása újraindítást igényel. Amennyiben sikeresen bekapcsoltuk vagy bekapcsolt állapotban találtuk a szolgáltatást kezdhetjük is az üzemeltetésre szolgált programunk telepítését.

Ehhez nincs más teendőnk mint elindítjuk a lemezen található Docker For Windows Intaller.exe futtatható állományt.

Telepítés Linux disztribúciók alatt:

Ebben a szakaszban, csak egy példányra térek ki pontosan, a legelterjedtebb disztribúció, ha a listában nem szerepel az ön által használt, kérem látogasson el a [Docker hivatalos dokumentációjának oldalára, további telepítési információért.](https://docs.docker.com/)

Az installáció lépései Ubuntu rendszerek alatt:

Indítsuk el a számítógépünkön található terminál emulátort (ez szerepelhet Terminal, Konsole, Xfce Terminal néven a telepített alkalmazásaink listájában), ebbe gépeljük be a következő parancsokat ügyelve a sorrendre az írásjelekre és a szóközök helyére:

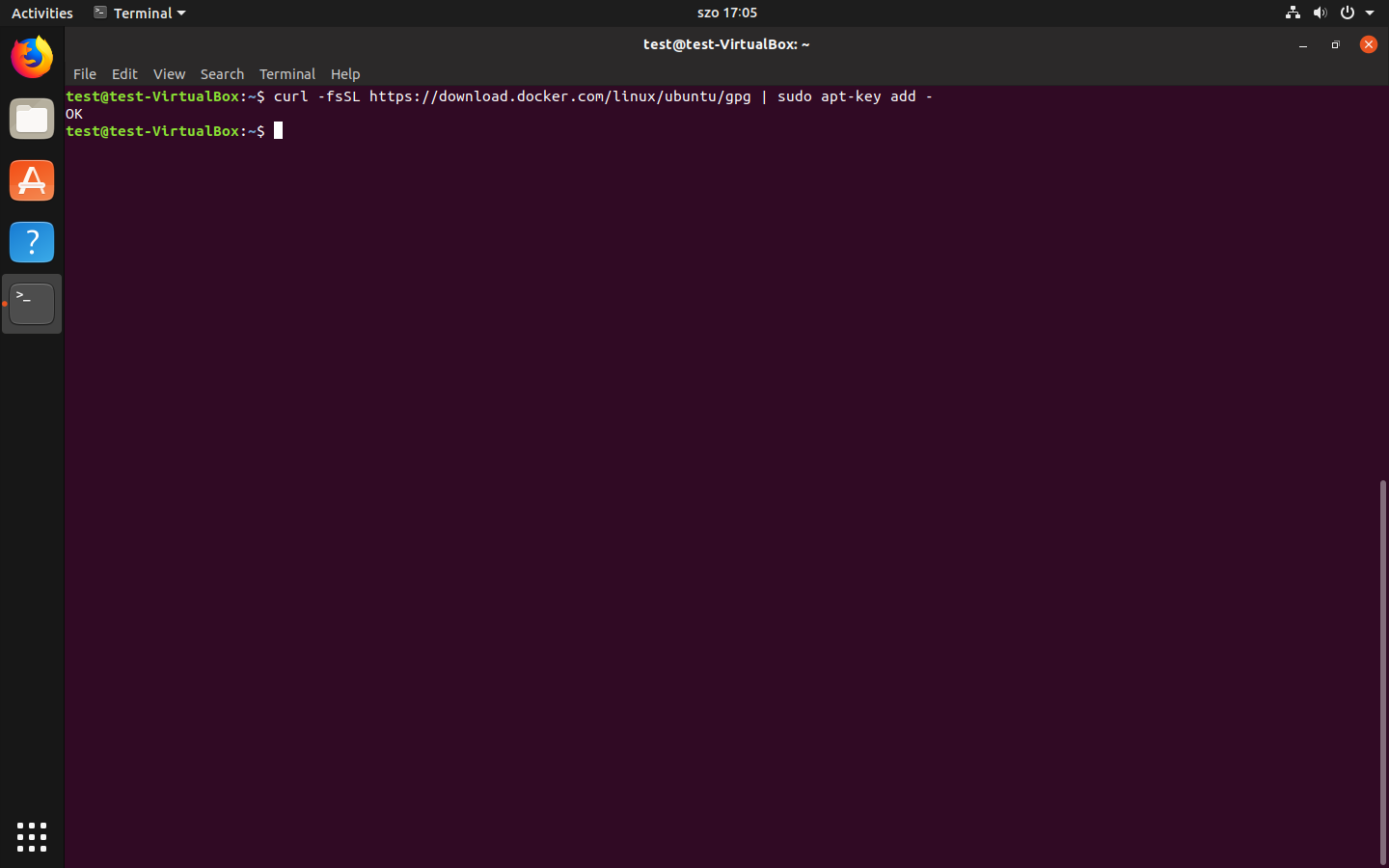
sudo apt-get update

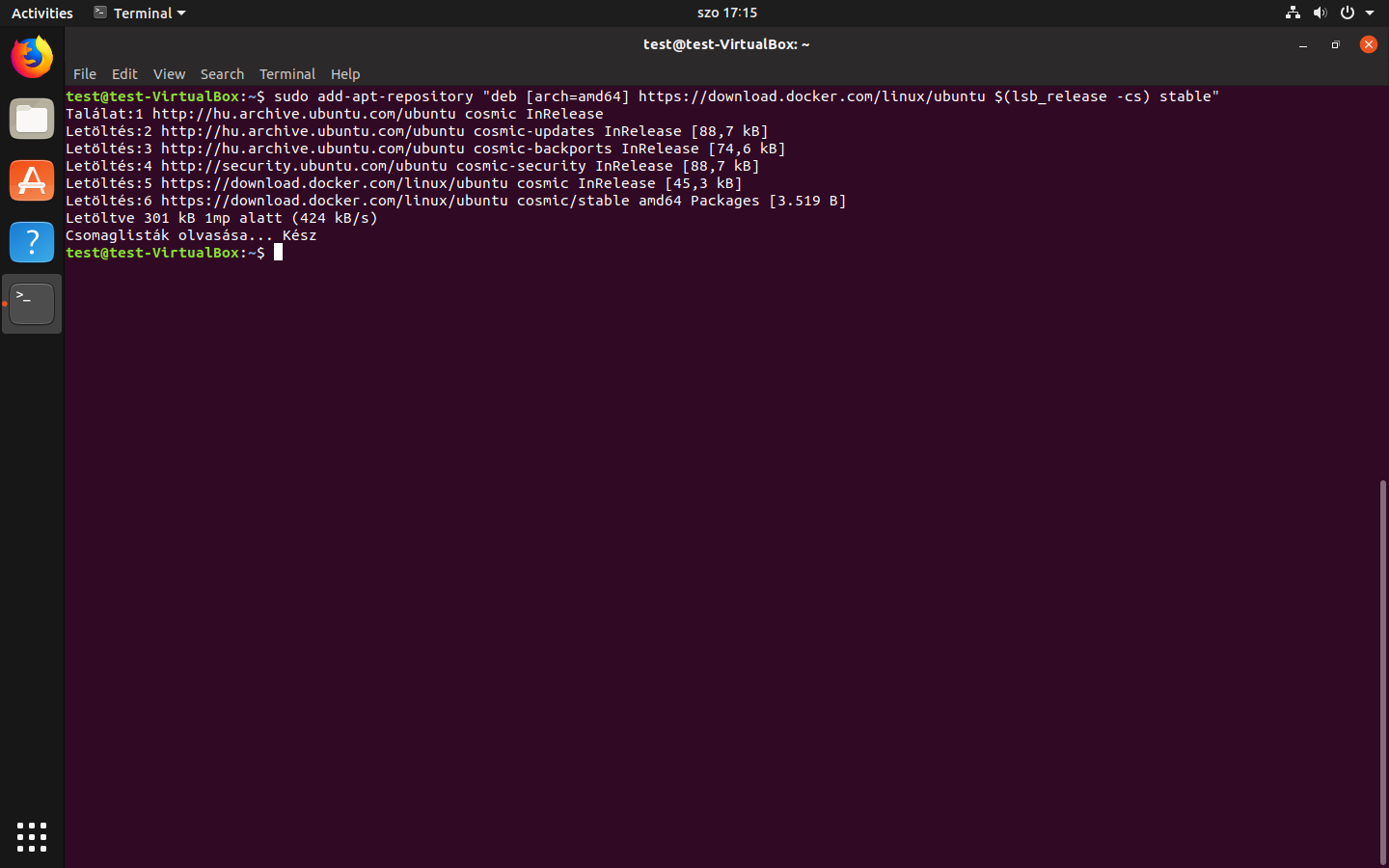
Ez után a parancssor elfogja kérni tőlünk a rendszergazda jelszót, ennek ha nem vagyunk birtokában kérjünk rendszergazdai segítséget, illetve engedélyt. A lefuttatott parancs lekérdezi a telepített naprakészségét, és a hozzátartozó információkat, ne ijedjünk meg akkor sem ha ennek a parancsnak hatására rengeteg frissítés történik ez nem probléma. Ez után lefuttatjuk a következő parancsot:

sudo apt-get install apt-transport-https ca-certificates curl gnupg-agent software-properties-common -y

Ez a parancs lehetővé teszi a csomagkezelőnk számára hogy egy titkosított csatornán keresztül tölthessünk le csomagokat. Ezután a következő parancs lefuttatása szükséges:

curl -fsSL <https://download.docker.com/linux/ubuntu/gpg> | apt-key add -

Ennek az utasításnak köszönhetően bekerül a letöltendő csomagok listájába a Docker üzemeltetési környezetünk, telepítését kiszolgáló szerverének titkos kulcsa amivel meghívható lesz csomagkezelőnkbe a következő utasítás után. Beírása után egy „OK” feliratot kell látnunk a képernyőn. Ezután a következő összetett utasítást kell lefuttatnunk, itt fokozottan felhívnám a kedves felhasználó figyelmét a szóközök írásjelek és kis nagy betűk különbségének odafigyelésére, ha átláthatatlannak találjuk terminálunk, töröljük le a képernyő tartalmát a „clear” parancs használatával!

 sudo add-apt-repository "deb [arch=amd64] <https://download.docker.com/linux/ubuntu> $(lsb\_release -cs) stable"

Ahogyan az ábra is szemlélteti sikeresen hozzá adtuk csomagkezelőnk frissítendő listájához, az üzemeltetési környezetünk. Ezután megint gépeljük be a csomag lista frissítésére szolgáló parancsot amit a konzol megnyitása után is tettünk.

sudo apt-get update

Ha lefutott és visszakapjuk a vezérlést gépeljük be a következő telepítő utasítást:

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose -y

Amennyiben helyesen gépeltünk be mindent visszakapjuk a vezérlést, ezennel a Docker környezetünk telepítése sikerrel zárult. Ebben az esetben azonban számítógépünk helyi felhasználója nem jogosult ennek használatára, ehhez meg kell hívni helyi felhasználónkat az úgynevezett „docker” felhasználó csoportba. Melyet a következő utasítással végzünk el:

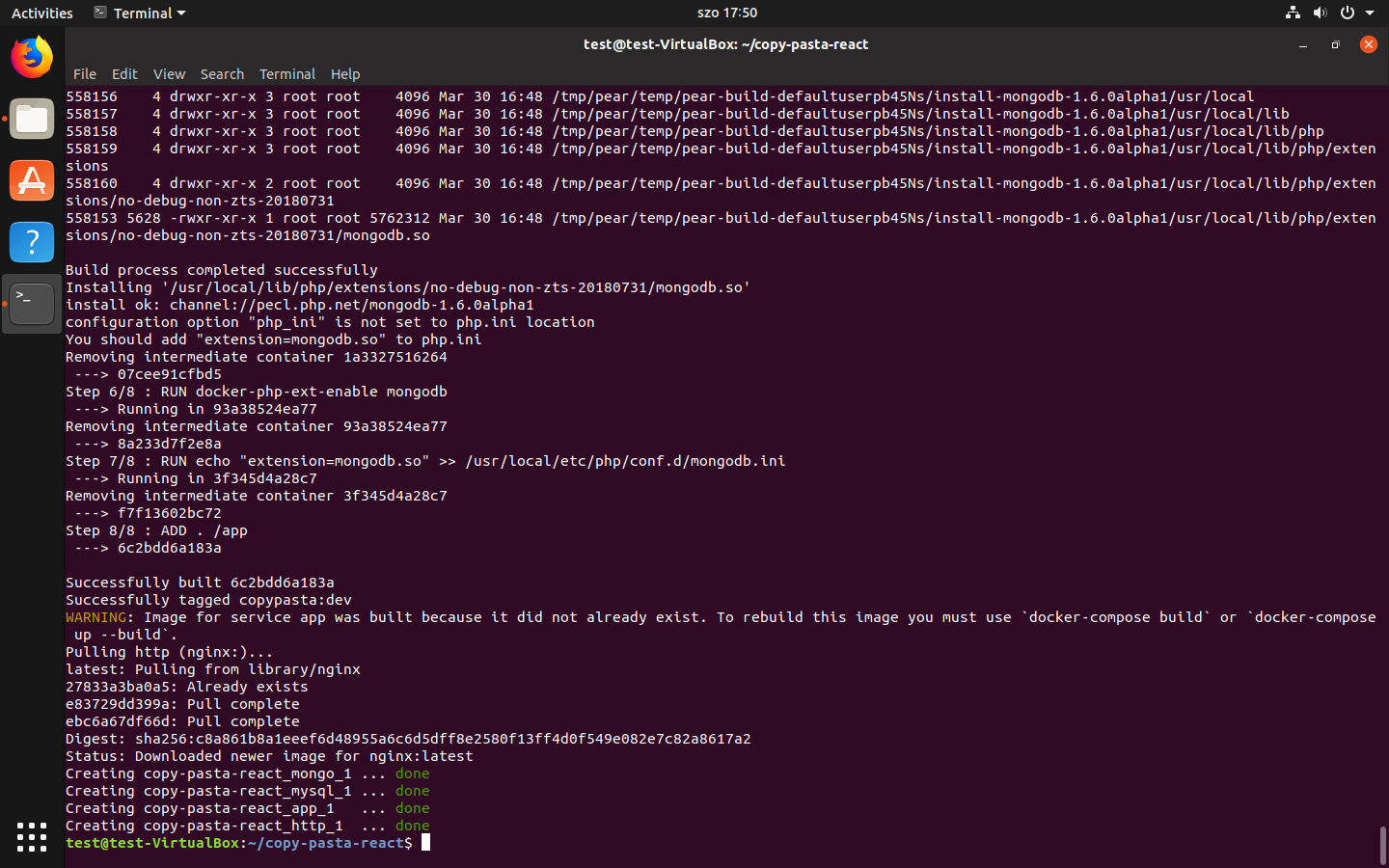
sudo usermod -aG docker $USER

Ezután indítsuk újra számítógépünket, hogy a felhasználói jogkör módosítások életbe lépjenek. Ez a lépés kötelező!

Most másoljuk fel magát az alkalmazást bárhova a számítógépre, ezután nyissuk meg a mappát ahova felmásoltuk majd a fájlkezelőben egy üres helyre való jobb egérgomb kattintással hozzuk elő a menüt és válasszuk ki az „Open in Terminal” menüpontot ezután megint szemben találjuk magunkat a terminál ablakunkkal azonban mint látjuk most van egy útvonal felhasználó nevünk után ez általában más színnel van jelölve, innen tudjuk hogy egy másik mappába adjuk ki az utasításokat. Ezek után nincs más teendőnk mint beríni a következő parancsot:

sudo docker-compose up -d

Ezután mint látjuk elkezdődik egy letöltési folyamat. Ilyenkor a Docker környezetünk beszerzi azokat a konténereket melyek szükségesek számára az alkalmazás felépítéséhez, ez több percig is eltarthat internet kapcsolattól függően. Ezután kezdetét veszi egy úgynevezett „build” azaz felépítési folyamat melyben Docker környezetünk testre szabja magának az előző lépésben letöltött konténerek listáját, ez a lépés szintén eltarthat több percig számítógépünk teljesítményétől függően. Miután a fentebb említett két lépés lefutott látni fogjuk hogy alkalmazásunk szolgáltatásai elindultak, ezeket zöld „done” feliratok jelzik minden egyes szolgáltatás mellett.

Ezután az alkalmazásunk elérhető a [http://localhost:8000](http://localhost:8000/) címen.

## 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 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.

# 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.

### 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.

## 7. 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: 1-2 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

Elektronikus dokumentum. Meg kell adnod a szerzőt, a letöltés helyét

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.