BGSzC Pestszentlőrinci Közgazdasági és Informatikai Szakgimnáziuma

1184 Budapest Hengersor 34.

1. Záró dolgozat

Online kódmegosztó

Konzulens tanár: Készítette:

Péter-Szabó Richard Czékus Máté

Tartalom

[Tartalom 2](#_Toc4948125)

[1 Bevezetés 3](#_Toc4948126)

[1.1 Feladat leírás 3](#_Toc4948127)

[1.1.1 Általános ismertetés 3](#_Toc4948128)

[1.2 A felhasznált ismeretek 3](#_Toc4948129)

[2 Felhasználói dokumentáció 4](#_Toc4948130)

[2.1 A program általános specifikációja 4](#_Toc4948131)

[2.2 Rendszerkövetelmények 4](#_Toc4948132)

[2.2.1 Hardver követelmények 4](#_Toc4948133)

[2.2.2 Szoftver követelmények 5](#_Toc4948134)

[2.3 3. A program telepítése 6](#_Toc4948135)

[2.4 A program használatának a részletes leírása 9](#_Toc4948136)

[3 Fejlesztői dokumentáció 16](#_Toc4948137)

[3.1 Az alkalmazott fejlesztői eszközök 16](#_Toc4948138)

[3.1.1 Adatmodell leírása 17](#_Toc4948139)

[3.2 Részletes feladatspecifikáció, algoritmusok 18](#_Toc4948140)

[3.3 Tesztelési dokumentáció 20](#_Toc4948141)

[3.4 7. Továbbfejlesztési lehetőségek 21](#_Toc4948142)

[4 Felhasznált irodalom 23](#_Toc4948143)

# Bevezetés

## Feladat leírás

### 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.

Az oldal, a szélesebb közönség megszólítása érdekében, ANGOL nyelvű, ezáltal használatához az angol nyelv használata elengedhetetlen.

## 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

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:

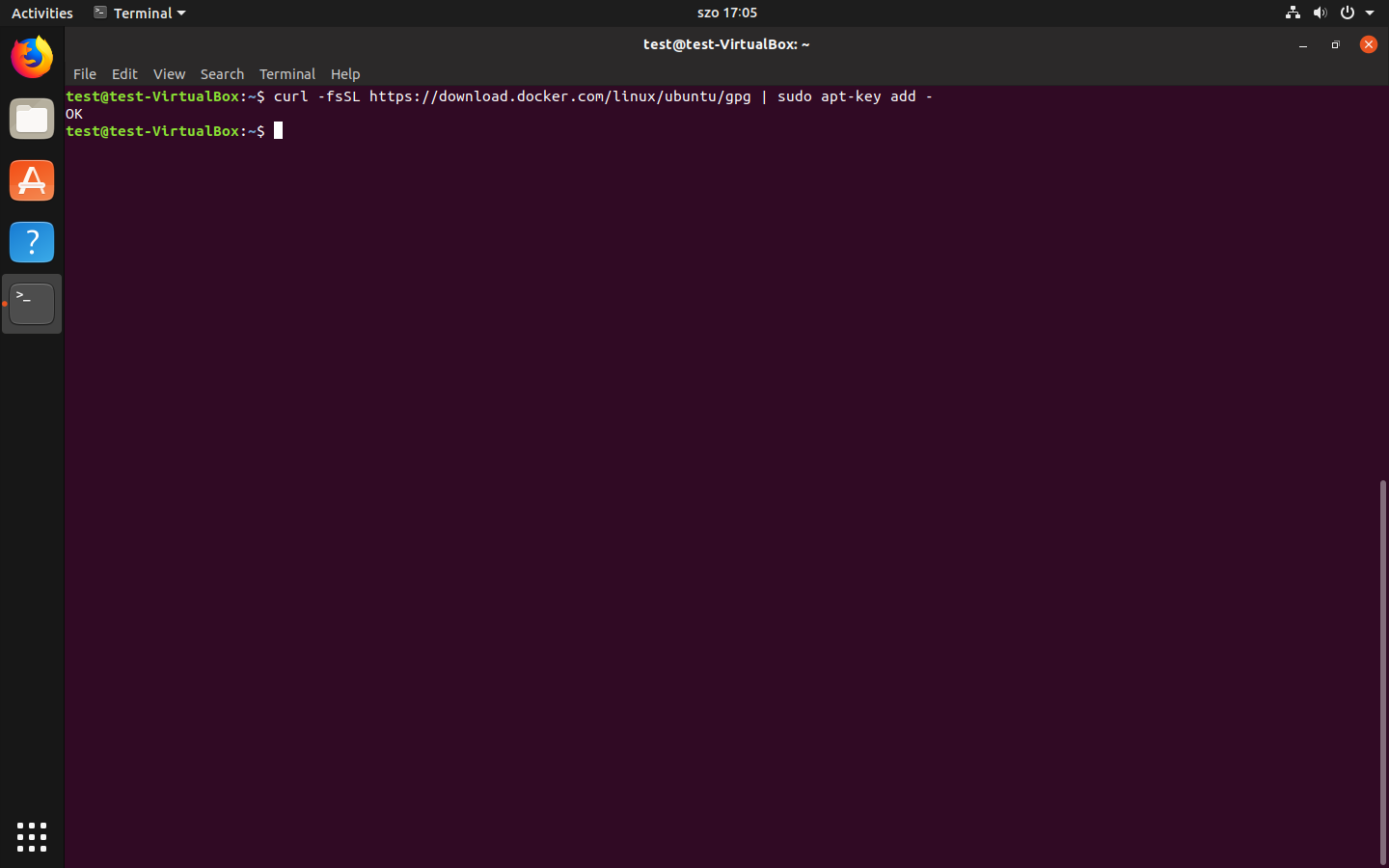
sudo apt-get update

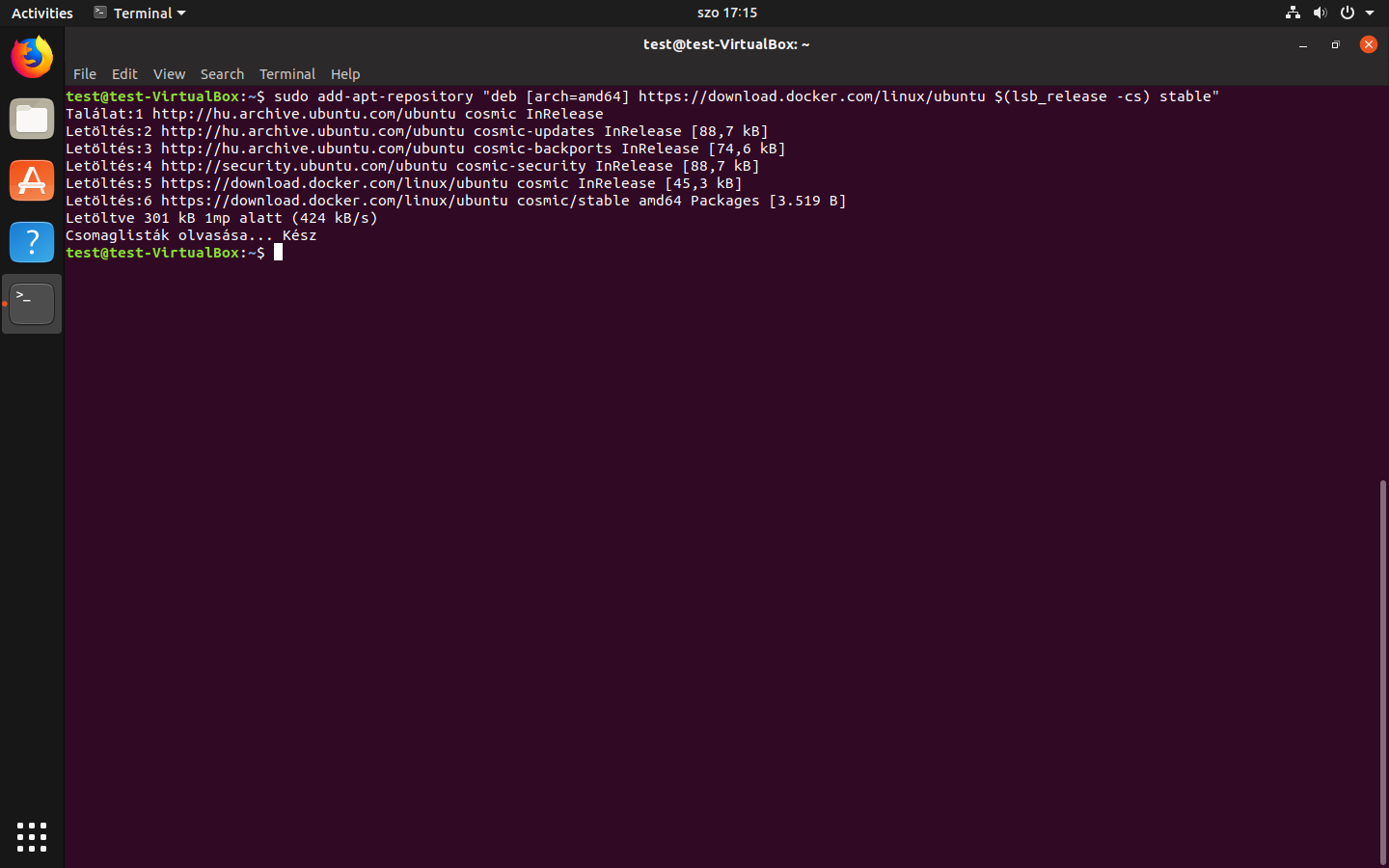
Ez után a parancssor elfogja kérni tőlünk a rendszergazda jelszót, ennek ha nem vagyunk birtokában kérjünk rendszergazdai segítséget, illetve engedélyt. A lefuttatott parancs lekérdezi a telepített naprakészségét, és a hozzátartozó információkat, ne ijedjünk meg akkor sem ha ennek a parancsnak hatására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

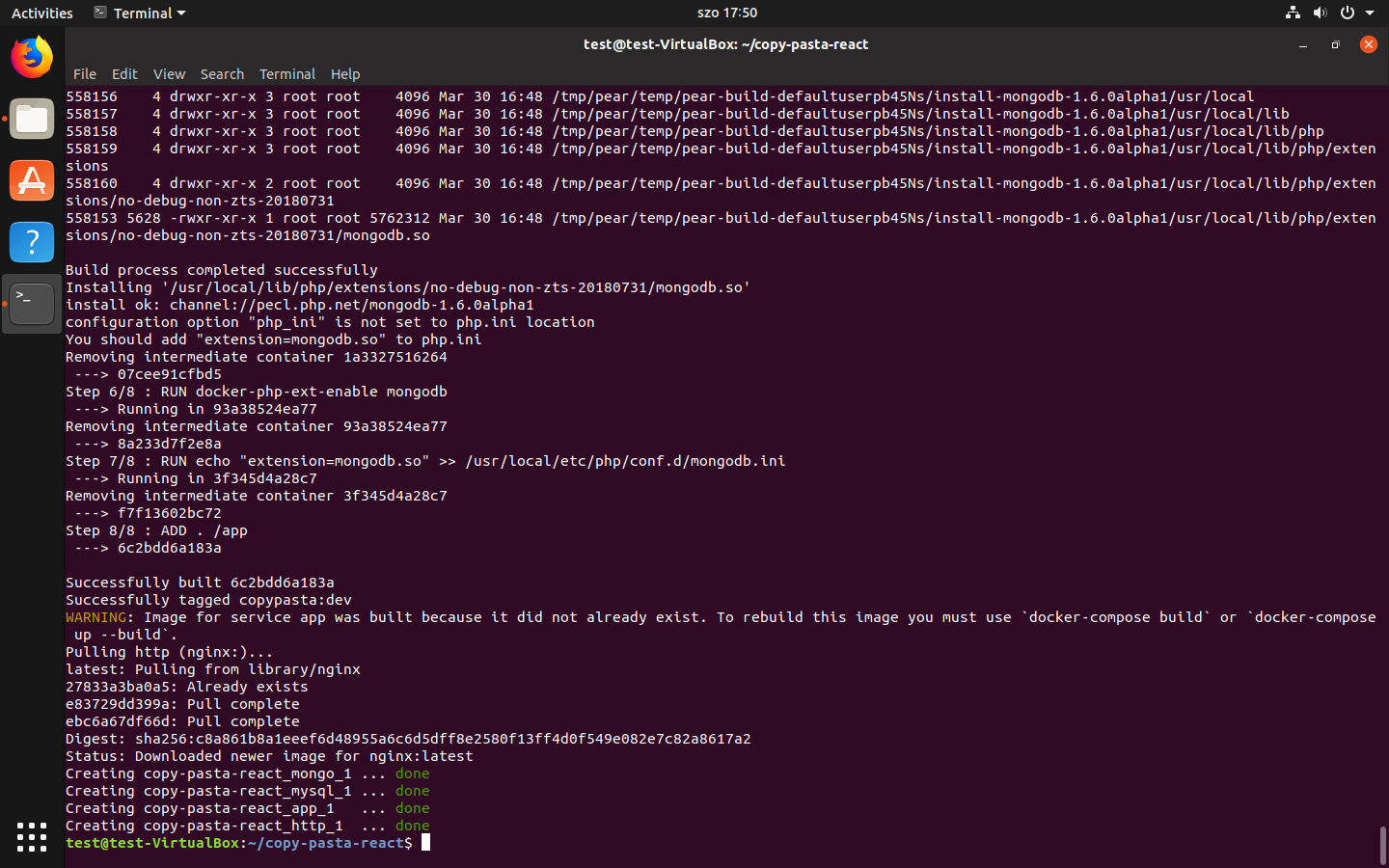
sudo usermod -aG docker $USER

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.

Alkalmazásunk ezek után leállítható a program gyökérkönyvtárában kiadott paranccsal:

docker-compose down

## A program használatának a részletes leírása

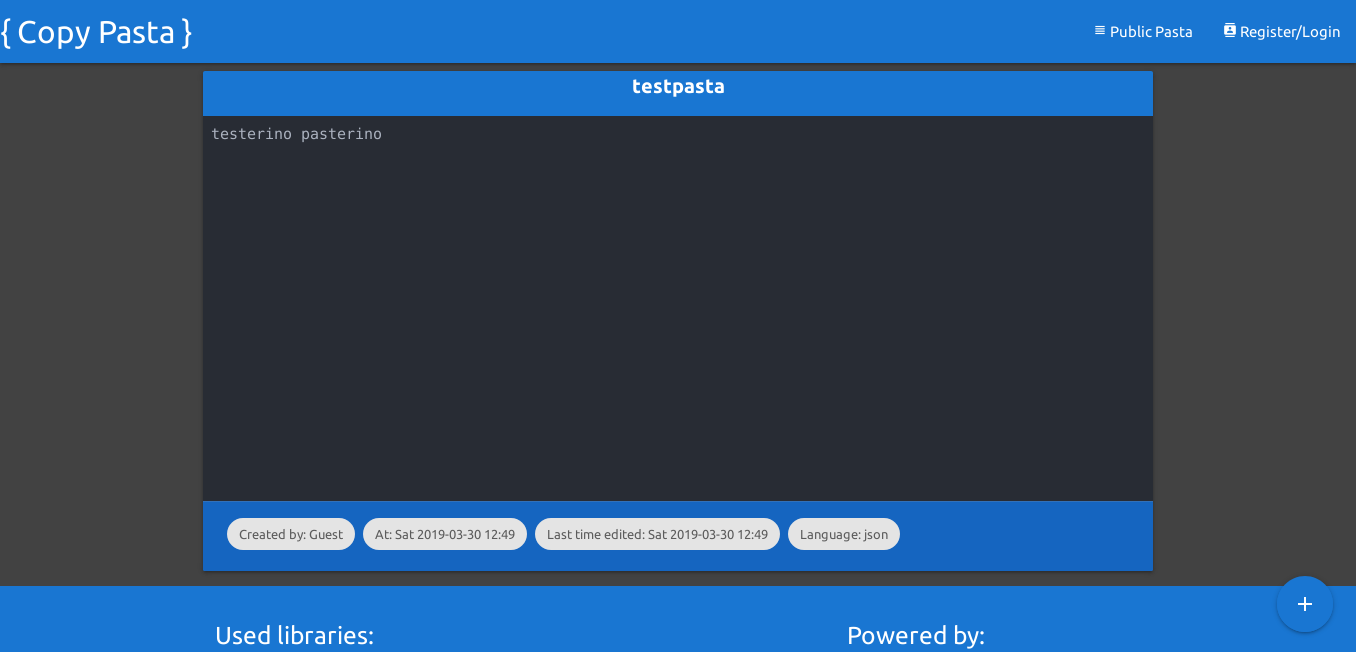
**Általános és minden oldalon megtalálható funkciók**

Felül található egy navigációs sáv, tartalmaz egy logót mely számítógép és nagyobb táblagépeken a képernyő bal felső sarkában, kis méretű táblagépeken, mobiltelefonokon ez a logó a képernyő közepén található, ezt megnyomva a kezdőlapra navigálhatunk. Szintén ezen a navigációs sávon nagyobb felbontásokon, a képernyő jobb felső sarkában, kijelentkezett állapotban található két darab menüpont ezek közül az egyik a „Public Pasta” menüpont mellyel a nyilvánosan publikált tartalmak megtekintésére szolgáló kezdő oldalra lehet vissza navigálni, a másik menüpont a „Register/Login” nevet viseli, ez a nevéből is adódóan a regisztrációra és bejelentkezésre szolgáló oldalra hivatott átirányítani, ezek a menüpontok mobil nézeten, a képernyő bal oldalán a logótól balra találhatóak.

Bejelentkezett állapotban ezen menüpontok egyike, pontosabban a „Register/Login” menüpont megváltozik egy „Profile” nevezetű menüpontra, erre a menüpontra kattintva átirányít minked a saját profilunk megtekintésére szolgáló oldalra, asztali nézetben szintén ezen az oldalon, mobil nézetben a logó jobb oldalán, található egy sötétebb színű kijelentkezés gomb.

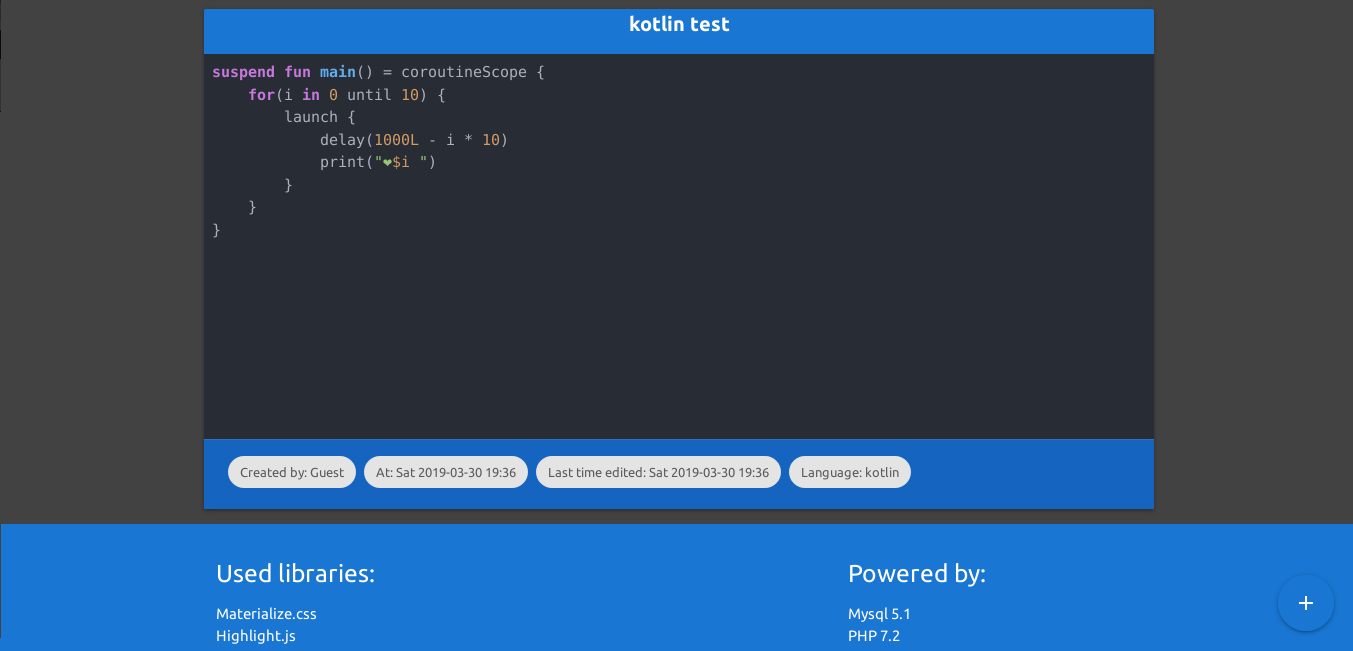
Ha adminisztrátori jogkörrel rendelkezünk ezen a felső sávon található egy menüpont az adminisztrációs feladatok ellátására is „Admin View” néven.

Az oldal alsó részén található egy lábléc, mely kijelzi az oldalon használt programozási eszközöket és keretrendszereket a bal oldalon, mobilnézetben felül, és egy, az oldal üzemeltetése során használt technológiákat jegyzet található ezen lábléc jobb oldalán, mobil nézetben ez az előbb említett rész alatt található. Az oldal legalsó részén sötétebb színnel jelzett rész található, ez az szoftver létrejöttének dátumát, licencét és létrehozójának nevét látjuk.



Public Pasta – Kezdőlap

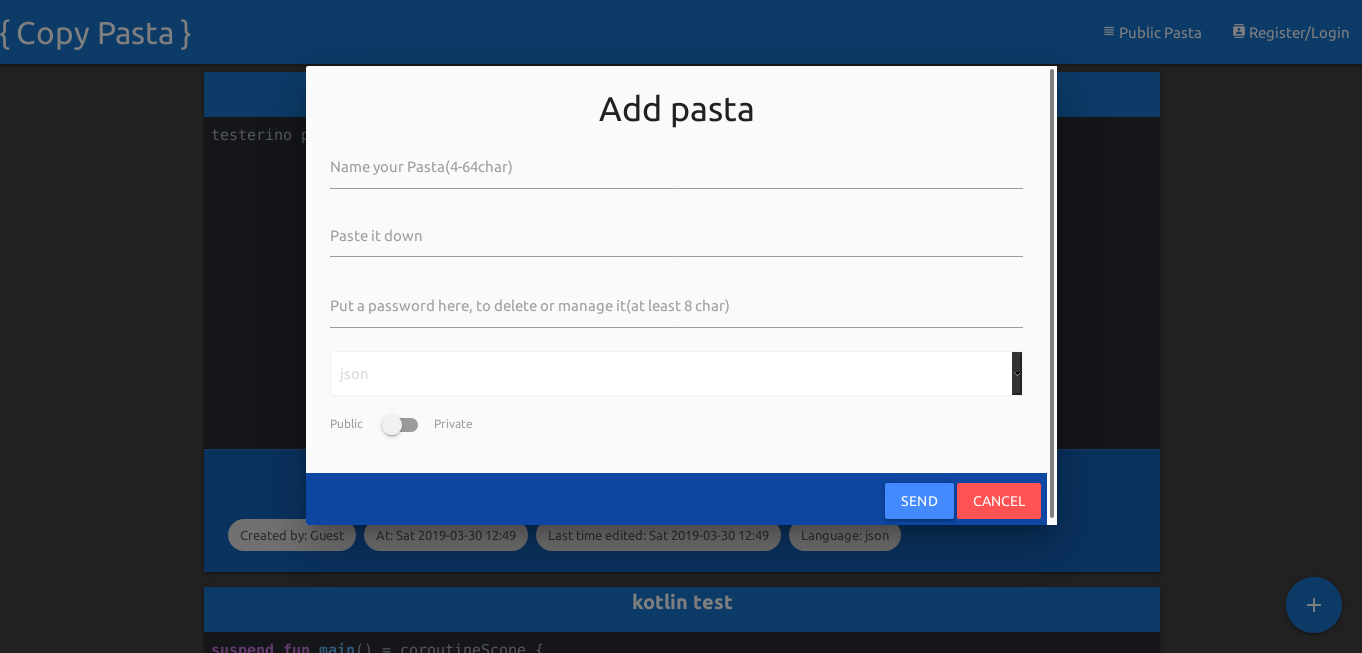
Ez az oldal kezdőlap része, illetve a Public Pasta gomb megnyomása is erre az oldalra irányít, itt lehet látni minden olyan feltöltött, publikációt mely nem privát formában került feltöltésre a weboldalra. A szóban forgó szövegek, mindegyike egy-egy kis kártya szerű tartalom megjelenítésre szolgáló modulon tekinthető meg. Ezek a felületek felül egy címsávval rendelkeznek, itt tekinthető meg a cím, illetve az erre való kattintás szolgál a jegyzet közelebbről való megtekintésére.

Középen egy sötét kékes szürke mező található, mely magát a létrehozott kód- vagy szövegrészletet jeleníti meg, amennyiben a kivonat egy az oldal által is támogatott programozási nyelven íródott, érvénybe lép a weboldal szintaxis megkülönböztető funkciója, mely az adott programnyelv kulcsszavait írásjeleit és tagoltságát hivatott megkülönböztetni a sztenderd szövegektől. Ezeken a tulajdonságokon felül, bármelyik középső tartalom dobozra kétszer kattintunk tartalma azonnal a számítógép vágólapjára kerül ezek másolásának megkönnyítésének érdekében, ezen funkcionalitásra való felhívás megtalálható minden komponens alsó részében.

Minden egyes modulunk alján található még egy záró rész mely az adott publikáció legfontosabb információit hivatott leírni, tartalmazza a textus létrehozójának nevét, ha ez a mező a „Guest” nevet érinti, abban az esetben egy vendég hozta létre. Mellette szorosan található a létrehozás ideje és dátuma, emellett az utolsó módosítás időpontja, majd a programozási nyelv melyben az adott kódrészlet íródott, általános szöveg esetében ez a „plain text” feliratot hordozza magában. Ezen „kártyák” egymás alatt jelennek meg az utolsó módosítási dátum régiségének függvényének növekvő sorrendjében.

Az oldal jobb alsó sarkában található az új tartalom felvitele gomb, ez a komponens egy kék körben található fehér plusz jel, úgynevezett lebegő gomb, követi a kijelzőt pozícióját így a kezdőlapon mindig elérhetjük és a vele felnyíló Új tartalom feltöltése menüt is, a jobb alsó sarokban, mobilnézeten a gomb ugyan ezekkel a tulajdonságokkal rendelkezik, csak jóval kisebb kontextusban, ezáltal nem kitakarva egyetlen fontos részletet sem, ezzel zavartalanná téve a tartalomfogyasztást.

Új tartalom feltöltésére szolgáló felugró ablak

Ez a felugró ablak tartalmaz egy címsávot, alatta közvetlen 3 darab bemeneti mező található, ezekre kattintva a címsáv felúszik a szövegdoboz fölé így be tudjuk írni az általunk kívánt értékeket.

Az első mezőben „Name your Pasta” egy címet kell megadnunk ami legalább 4 de legfeljebb 64 karakter hosszú lehet.

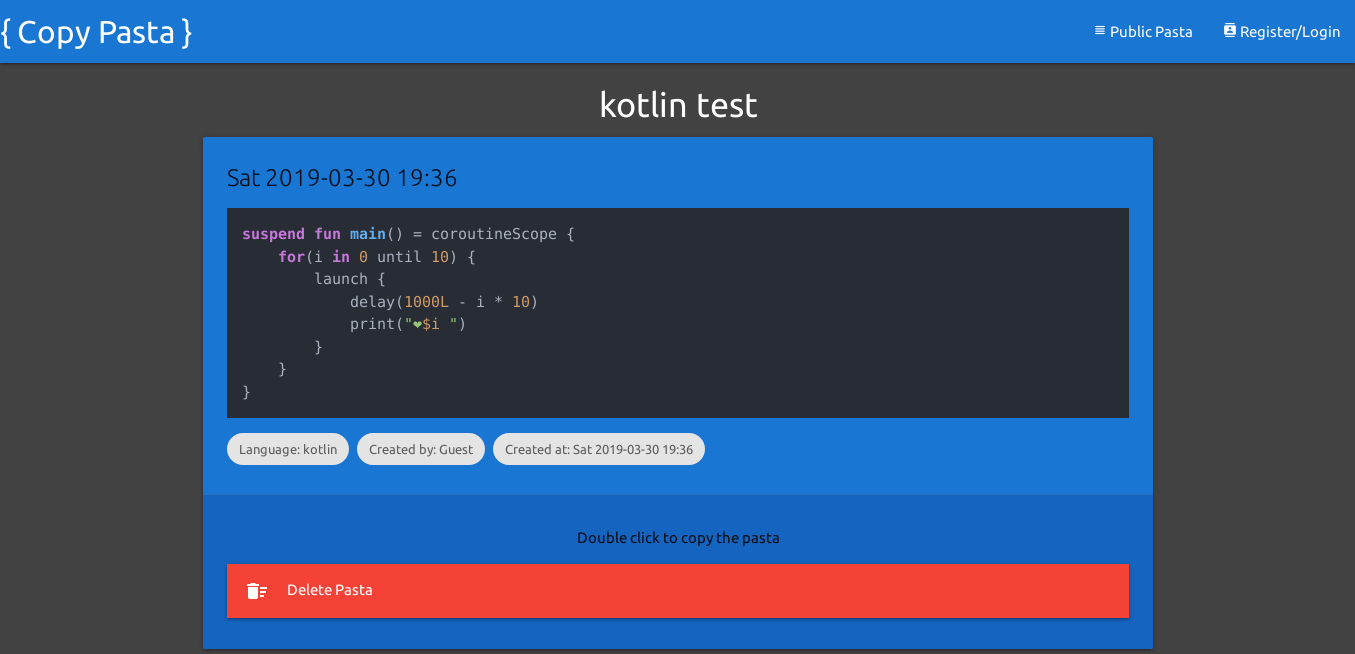
A második mezőbe „Paste it down” a megosztásra kívánt szövegünk kerül, ez a mező nem lehet üres.

A harmadik mezőbe a jelszó kerül „Put a password here”, abban az esetben ha vendégként zajlik a folyamat, abban az esetben ha bejelentkezett felhasználóval rendelkezünk ez a mező nem látszik amivel később törölhetjük az általunk feltöltött tartalmat, így jegyezzük meg vagy esetleg írjuk fel, később ennek módosítására nincs lehetőség, ennek legalább 8 karakter hosszúnak kell lennie.

A harmadik pontban kiválaszthatjuk milyen programozási nyelven íródott a bemásolt tartalom, ez egy jelen pillanatban 64 darab, a napjainkban legtöbbet használt programozási nyelvekből és konfigurációs fájlok jelölő nyelveiből álló választék.

Az utolsó menüpontban kiválaszthatjuk egy két irányú kapcsolóval, hogy publikus „public” vagy privát „private” a feltöltött tartalom, az esetben ha a kapcsolót jobbra azaz a privát oldalra kapcsoljuk, csak a link birtokában lesz megtekinthető az általunk feltöltött tartalom, illetve regisztrált felhasználóként a felső navigációs sávban a profilunkra kattintva is elérhetjük, cím alapján.

A tartalmak közelebbi megtekintésére szolgáló oldalaira

Ezen az oldalon, középen található az általunk kiválasztott kód címét, alatta ismét egy „kártya” komponenst találhatunk, ezúttal a címsávban az utolsó módosítás dátumát, hiszen ha erre az oldalra navigáltunk ez egy fontos információ lehet számunkra, középen szintén egy sötét-kékes-szürke szövegmezőt találunk a már fentebb említett szintaxis kiemelő funkcióval, alatta a nyelvet, készítőjét és a készítés dátumát.

Ezúttal rendelkezünk egy kicsit sötétebb lábléccel is, mely tartalmaz egy felhívást a dupla koppintásos másolásra, illetve egy kicsit nagyobb, a tartalom törlésére szolgáló gombot.

Ezt a gombot megnyomva, lenyílik egy piros menüsáv mely az esetben ha az adott tartalmat egy vendég hozta létre kér egy jelszót mely megadásával törölhető lesz a tartalom. Abban az esetben ha a tartalmat létrehozó felhasználó és a belépett felhasználó, egy és ugyan az a személy akkor a tartalom jelszó nélkül törölhetővé válik, a menü lenyitása utána a törlés gombot megnyomva. Ha nem rendelkezünk a jelszóval a vendég tartalomhoz és nem is a tartalmat létrehozó felhasználó vagyunk az esetben ez a gomb funkcióját veszíti. Amennyiben az alkalmazásban található egyetlen moderátor felhasználóval jelentkezünk be ezek jelszó és a létrehozó felhasználó nélkül is törölhetőek az oldal, adminisztrációjának érdekében.

Regisztráció és Bejelentkezés

Ez az oldal hivatott ellátni a regisztrálás és bejelentkezés folyamatát.

Az oldal felső részén a navigációs sáv alatt található a bejelentkezési felület, itt két mező található, egy felhasználónév „Username” és egy jelszó „Password”. Feltétel, hogy már regisztrált felhasználóval kell rendelkezned.

A bejelentkezés során több-féle hiba léphet fel, az oldal egy hibaüzenettel reagál erre majd a hibaüzenet után visszairányít a regisztrációs és bejelentkezési oldalra.

Ez a hibák a nem létező felhasználó „The user doesn't exits”.

A jelszónak legalább 8 karakternek kell lennie „The password must be at least 8 char logn”

Illetve a nem megfelelő jelszó lehet „Wrong password”.

A sikeres bejelentkezés esetén szintén egy üzenet fogad minket átirányítás előtt.

A felület alsó részén a bejelentkezés „Login” rész és a lábléc felett helyezkedik el a regisztrációs „Register” rész, melyben új felhasználót tudunk létrehozni. Itt 4 mező található, egy a felhasználónév számára „Username”, mellette jobbra egy az email cím számára fent tartott mező található „Email”, alattuk két darab jelszó mező található „Password”, „Verify Password”, ezekből a baloldalt található mezőbe a tőle jobbra elhelyezkedő bemeneti szövegdoboz a jelszó megerősítésére szolgál, hogy biztosak legyünk benne hogy jól írtuk be a jelszavunkat.

Regisztráció során szintén több hibaüzenettel találhatjuk szemben magunkat, ezek következőek lehetnek:

„The user is already exist.” azaz a felhasználó már létezik.

„The email address isn't valid” Az email cím nem valós.

„The two password doesn't match” a két jelszó nem egyezik.

„You should fill all the fields” azaz minden mező kitöltése kötelező.

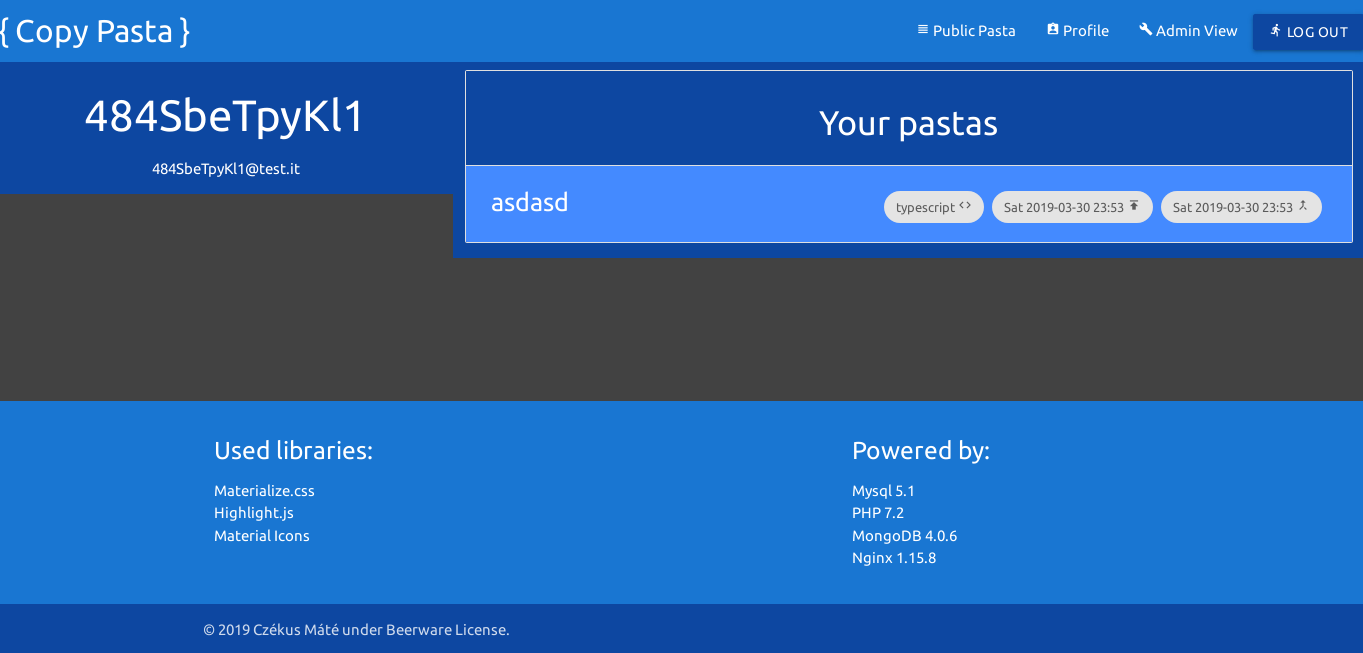
„The password must be atleast 8 char long” a jelszónak legalább 8 karater hosszúnak kell lennie.

Illetve ezen a szakaszon is a sikeres regisztrációt követi egy üzenet, illetve a rendszer automatikusan beléptet az új felhasználóba.

Profil oldal

A profil oldal, az alkalmazásban regisztrált felhasználónk adatainak áttekintésére szolgál.

A szóban forgó weblap két részre osztható, az egyik része a bal oldalt található a személyes adatok megjelenítésére szolgáló mező, a jobb oldala a „Your pastas” nevet viseli ami a felhasználónk oldal használatán belüli feltöltési tevékenységét hivatott dokumentálni számára, egy átlátható listaszerű nézetben. Ebben a listában az általa az oldalon létrehozott tartalmak jelennek meg növekvő időrendi sorrendben, az elem bal oldalán a nevével mely egy kattintható link ami a teljes tartalomra mutat jobb oldalán pedig a legfontosabb tulajdonságaival..

Ha a felhasználó még nem hozott létre tartalmat az oldalon, egy kékes szürke lista elem található benne mely tartalma „You don't have any pastas, its time to create one!” biztató üzenetet tartalmazza.

# 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 program létrehozása során a VSCodium nevezetű nyílt forráskódú fejlesztői környezetben dolgoztam. - https://github.com/VSCodium/vscodium

Melyben három fontos, a munkámat megkönnyítő kiegészítővel dolgoztam.

One Dark Pro - https://marketplace.visualstudio.com/items?itemName=zhuangtongfa.Material-theme

PHP IntelliSense - <https://marketplace.visualstudio.com/items?itemName=felixfbecker.php-intellisense>

Material Icon Theme - https://marketplace.visualstudio.com/items?itemName=PKief.material-icon-theme

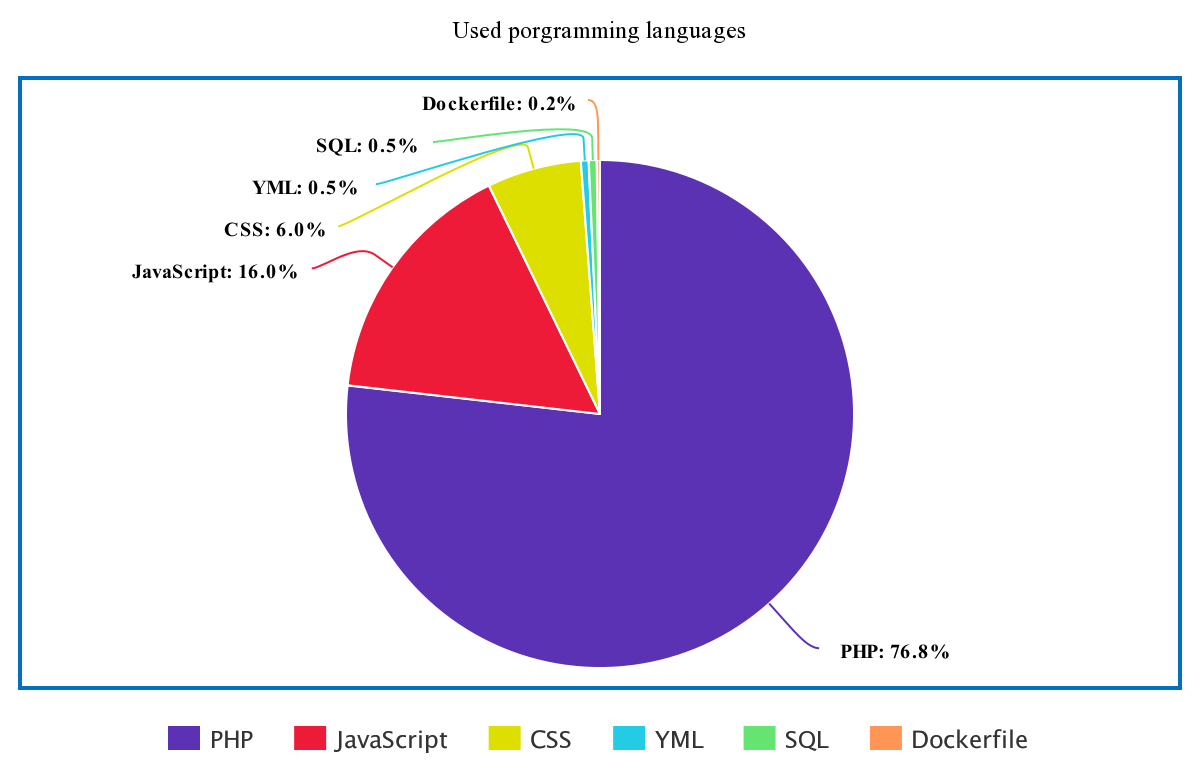
Dokumentáció írás tekintetében a LibreOffice nevű ingyenes, nyílt forráskódú irodai csomagot használtam. - https://www.libreoffice.org/

Adatbázis kezeléshez a parancssort, illetve a MongoDB Compass Community nevű szoftverét alkalmaztam.

Üzemeltetés és a program futtatása során a Docker Community Edition került felhasználásra, a docker-compose nevű nyílt forráskódú Python nyelven írt kiegészítővel.

Verzió kezelést git-el végeztem a fejelsztői környezet beépített Git kezelővel illetve a parancssoros interfésszel. Külső tárolóként egy prémium github privát repository-t alkalmaztam.

A használt programozási nyelvek PHP, JavaScript, CSS, SQL. A konfigurációs fájlokhoz a következőeket használtam: Dockerfile, yml.

Adatbázis rendszerek tekintetében kettő került felhasználásra:

MongoDB mely egy noSQL, objektum tároló.

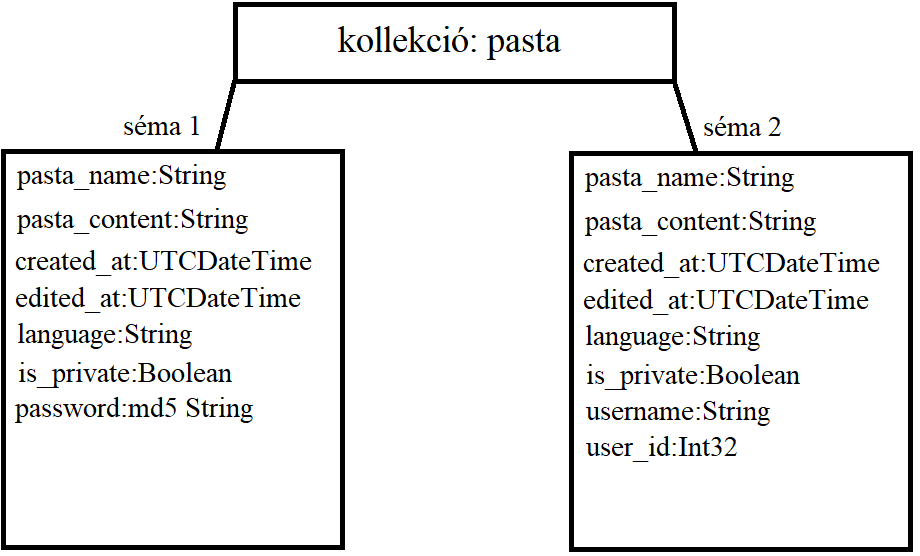
MySQL mely egy relációs adatbázis kezelő rendszer.

Továbbá a mások által készített és felhasznált könyvtárak és keretrendszerek listája:

MaterializeCSS, HighlightJS, Animate.css, jQuery, md5.js.

### Adatmodell leírása

MongoDB adatbázis motorban, objektumokat tárolunk, az összetartozó adatokat úgynevezett kollekciókba gyűjtve. Nem rendelkezik fixen kiállított adatstruktúrával, ha egy bizonyos sémát szeretnénk használni az adatbázisunkban azt a programban kell meghatároznunk ezáltal egy sablont előállítva kollekciónk számára. Bár ezen sablon használata nem kötelező, a rendszer szintén a fejlesztőre bízza ennek használatát. A program létrehozása során a következő két, egymástól eltérő, de sok tulajdonságában megegyező sémát hoztam létre az egy kollekcióba tartozó adatok részére:

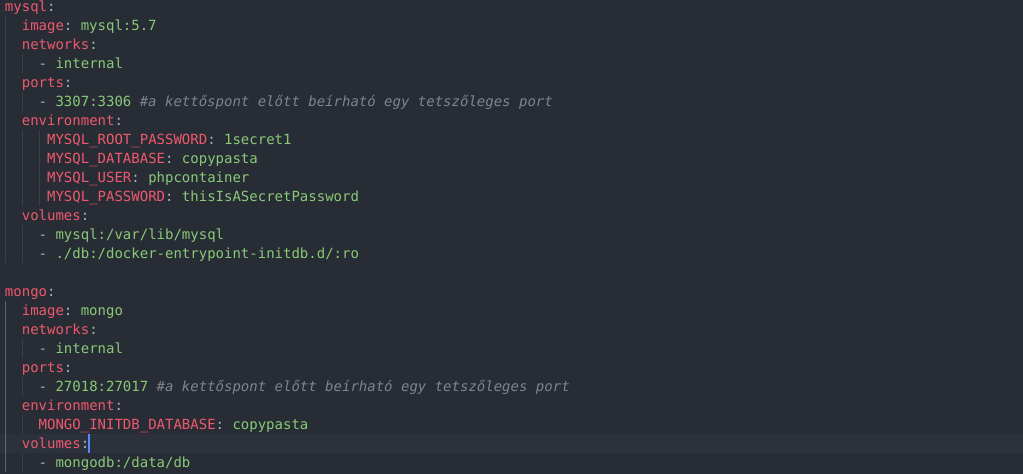
Emellett, a vezérlő, minden feltöltött objektum példányhoz hozzá rendel egy objektum azonosítót \_id néven, mely egy 12byte-on tárolt bináris érték, 24 karakteres hex string reprezentációja. Az első 4 byte a unix idő kezdete óta eltelt másodpercek száma, a következő 3 a szerver azonosítója, az utolsó előtti 2 a folyamat azonosítója, az utolsó 3 byte pedig egy számláló, mely egy pszeudo véletlen számot generál. Ez minden esetben az adott példány egyedi azonosítója.

Ezen felül MySQL-ben egy ettől jelentősen egyszerűbb felhasználó tábla található az adatbázisunkon belül. Egy egyszerű kulcsmezővel ami a user\_id, ezeken kívül kettő darab unique mezővel melyek azt a célt szolgálják hogy ne szerepelhessen két felhasználó ugyan azon a felhasználónéven.

## Részletes feladatspecifikáció, algoritmusok

A program karbantatrásához szükséges eszközök közé tartozik egy adminisztrátor, ez nem előre definiált, ezáltal először egy regisztráció során jön létre, a következő felhasználónak mindenre rálátása van az alkalmazáson belül illetve bármit törölhet, ez pedig a 484SbeTpyKl1 felhasználó név.

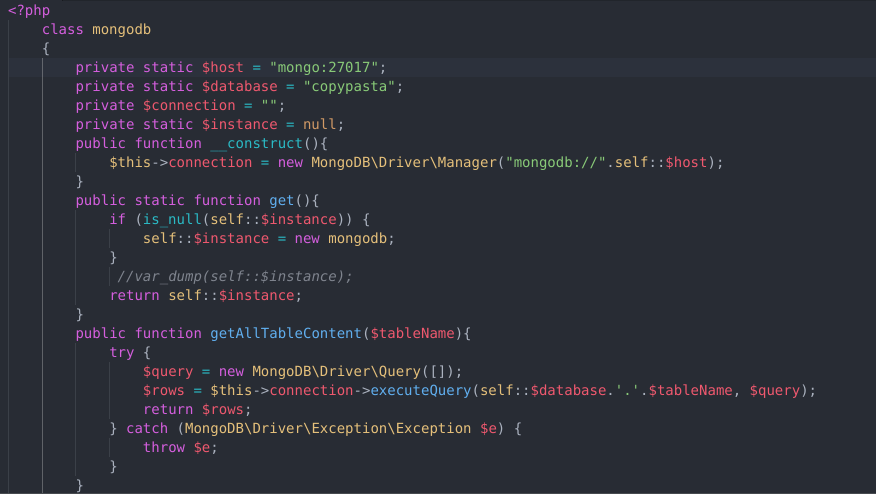
A fejlesztés és karbantartás során továbbá két fontos lépés lehet még, az egyik az adatbázisok menedzselése, a másik az alkalmazás belső kommunikációs hálózati rétege ehhez fontos tudnuk hogy működik a konténerek közötti belső kommunikáció, a Docker a „docker-compose up ” vagy „docker-compose up -d” parancs kiadása után felépíti a belső hálózati infrastruktúránkat, ez egy csak a konténerek között elérhető egyedi ID-vel rendelkező maszkolt belső virtuális hálózat, ez az ID jelen esetben az „internal” a konténerek jelen esetben a két adatbázis a HTTP szerver és a PHP alkalmazás szerver ezen belül kommunikálnak egymással, ebből a hálózatból a konténer számára külső, számitógépünk számára belső hálózaton csak és kizárólag az NGINX webszerver képes kommunkálni, ezáltal az adatbázisok a szoftver módosítása nélkül nem vezérelhető, ezek biztonsági okokból nem lettek forward-olva a konténer csomagból.



Amennyiben elérésük mégis szükséges lenne a „docker-compose down” parancs kiadásával leállítva a konténer csomagot a „docker-compose.yml” fájl módosításásával tehető meg, ebben a fájlban a konténerek konfigurációja található egyesével felbontva, ha a külső hálózatra szeretnénk forgatni a mysql vagy mongo adatbázisunkat, létre kell hoznunk a „networks” beállításai alatt egy új beállítást „ports” névvel mely alá egy kötőjelet kivetően, a mysql-nél a 3306:tetszőleg port, mongodb esetén pedig a 27017:tetszőleges port, által az általunk megadott porton lesz elérhető a szerverünkön az adatbázis kezelő számára az a program által alkalmazott táblák és kollekciók.

Az alkalmazás továbbá az objektum orientáltsága miatt további adatkezelő osztályokal rendelkezik, ezek közül legjobb példaként a Mongo adatbázist vezérlő interfészt tudom hozni. Ezt az osztályt saját magam készítettem korábban, ez a programon kívül megtekinthető a publikus Github felhasználómon.

Az osztály alkalmazásakor a feljebb említett részek alkalmazása nem figyelmen kívül hagyható, tehát az adatbázis motor továbbra is UTCDateTime értékekkel dolgozik dátum tekintetében, egyedi azonosító értelmében pedig Objektum \_id-kkel dolgozik.



Ez egy úgynevezett „singleton” osztály azaz a program során csak egy példány létezhet belőle, ezt úgy érjük el, hogy példányosítása csak a statikus „get” metóduson keresztül végezhető el, ez a függvény visszatér a már példányosított adatbázisunkkal hogyha már létezik. Rendelkezik továbbá 4 privát mezővel, melyek a kapcsolat felépítését szolgáló információk átadására készültek. Továbbá rendelkezik 4 publikus függvénnyel melyek mindegyike legalább egy kollekció nevet vár értékül, ennek külső válotózba való szeparálása, a több kollekciós műveletek végzésének érdekében nem történt meg, hogy az osztály képes legyen újra pldányosítás, vagy kollekció léptető függvény nélküli általánosított munkára.

A metódusok felépítése lehetővé teszi a az amúgy nehézkes, külső, C nyelvben íródott PHP nyelv kiegészítőjét.

Rendelkezik egy teljes tábla tartalom lekérésére szolgáló függvénnyel, ez a „getAllTableContent” nevet viseli, paramétereit tekintve csak egy kollekció nevet vár a feladat végrehajtásához, a visszatérési értéke egy asszociatív tömb melyenk minden eleme egy adatbázisból lekért objektum, saját mezőivel és metódusaival.

Egy „getFilteredContent” részegységgel mely bemeneti értéknek egy kollekció nevet illetve, egy filter-t vár, a szűrés tömb szűrésként működik, tehát létre kell hozni egy asszociatív tömböt, ennek első mezőjének nevét be kell állítani azon mező értékére amelyik értéke alapján szürni szeretnénk a tartalmat illetve ennek az értékét is be kell állítani.

Rendelkezik továbbá egy „inserObject” nevű függvénnyel is mely feladata, egy objektum beszúrása a megadott kollekcióba. Ez is tömb beszúrás alapon működik tehát a bemeneti értékeink a kollekció néven kívül egy előre be”mappolt” tömböt vár, a mezőértékek típusosságát betartva.

Végül, de nem utolsósorban rendelkezik továbbá egy deleteObject funkcióval, mely egy kollekció nevet illetve a törölni kívánt objektum egyedi azonosítóját várja el. Ezzel a függvénnyel törölhetjük az objketumokat kollekciónkból.

Bármelyik függvény tér vissza hibával az egy angol nyelvű mongodb hibaüzenet.

## Tesztelési dokumentáció

A program fejlesztésének a tesztelés szerves részét képezte a kezdetektől egészen a dokumentáció írás pillanatáig. A esetleges hibák a teszteken dokumentálva lettek üzenet formájában, az akkori tények állása szerint egy lehetséges megoldás is vázolva lett mellette melyek konzultáció során vagy konzultációk után kerültek bele a jegyzetbe. A véglegesítés pillanatában ez a fájlok megsemmisítésre kerültek.

Az osztályok és azok vezérlőinek létrehozása után dedikált „unit” tesztek voltak futtatva. Az integrációk előtt, integrációs tesztek is voltak is voltak implementálva a programkódba a funkciók egymással való inetrakciójának vizsgálata érdekében. Több esetben mikor felhívtam a figyelmet az adatbázis motor típus konvencióira, azok a teszthibákból visszatérő tapasztalatok megosztása volt. Főleg a dátum hibák kerültek elő legtöbbször, hiszen a UTCDateTime egy olyan típus mely az 1970-01-01 óta eltelt idővel tér vissza miliszekundumkoban, ennek konvertálása csak nehezen volt kivitelezhető mert a PHP más számolási metódust használ ezen formátum számítására. Ezt a hibát olvasásnál az 1000-el való osztással illetve ennek szorzásával a felvitelnél lehetett javítani, emellett egy TimeStamp konvertálásra is szükség volt mely egy dátumot a PHP által számolt TimeStamp-re vált. Ezen algoritmus 1 oldallal előbb megtekinthető.

Továbbá a szintaxis kiemelés rész esetén funkció integrálási hibák léptek fel, hiszen az általam írt másoló függvény több ponton is összeakadt ennek az osztálynak a függvényeivel.

Mivel ez is egy singleton függvény így inicializálással kapcsolatos hibák is léptek fel, rosszul tért vissza a már létező példányával, illetve példányon kívülre eső függvényhívások is történtek. Ezeket a hibákat egy teszter jeleztevissza aki a program korai verzióját kapta meg teszetlésre. Ezt a hibát könnyen orvosolhattam a függvényhívások és példányosítás sorrendbe rendezésével.

Léptek fel továbbá üzemeltetési hibák melyeket menet közben sikerült feltárni, webszerver átirányítási hibák melyeket egy félregépelés okozott a konfigurációs fájlban. Mellette fellépett egy probléma az adatbázis kommunikáció során hiszen a Docker virtuális hálózatán belül a dhcp-vel kiosztott IP címek használata nem kötelező, ezek helyett alkalmazható a konténer neve mely belső domain név átirányításként funkcionál a docker Swarm-on belül. Ezen hibák megoldását a hivatalos Docker dokumentáció, páratlan minőségben magyarázza meg.

A fellépő hibák listája csak bővült amikor észrevettem hogy a felhasználókezelésnél nincsenek visszatérő üzenetek a felhasználók számára a hibákról, ha nincsen kézzel fogható magyarázata annak mit csinál a program, könnyen nagyon sok felhasználó elpártolhat tőle, ezáltal a regisztrációba és a bejelentkezésbe bekerült egy hibakezelő modul. Itt a felhasználóknak ahogy a felhasználó dokumentáció lépéseinél is látszik, minden egyes probléma rendelkezik egy egyszerűen feldolgozható vizuális reprezentatívával, ezek megtekinthető a felhasználói dokumentációban a regisztráció és bejelentkezés résznél.

## 7. Továbbfejlesztési lehetőségek

További fejlesztési lehetőségnek véleményem szerint egy hibajegy feladási funkciót lehetne implementálni amelynél a regisztrált felhasználók hibajegyeket adhatnak fel a tartalom publikálójának ezzel elősegíteni a kooperatív munkát és elkerülni a hibalehetőségeket, vagy elősegíteni kezdő felhasználók fejlődési periódusának lépéseit.

Ezek mellett a program legnagyobb részére ráferne egy teljes refaktoráció a kód átláthatóságának és a kollaborációs tevékenység elősegítése érdekében.

MongoDB vezérlő osztályban bár indokoltan nem található kollekció léptetésre predeffiniált funkció, az osztály hosszú távú több projektre kiterjedő használatakor lehet hogy hasznos lenne, feltételekkel. A hibakezelés idő hiányában elég puritán ezáltal ezt is lehetne restaurálni, illetve külön funkcióba emelni.

Az objektum ID típusossága miatt célra vezető lenne egy objektum egy string-ből erre fordító funkció.

A dátum kiszámításának külső funkcióba emelése is növelhetné a produktivitást és átláthatóbbá tenné az osztály használatát kezdő felhasználók számára.

Egy tartalom módosító funkcióra is szükség lehet a későbbi felhasználás során ezáltal megkönnyítve feladatunkat.

Szükséges lehet továbbá a lekérdezések és visszatérési értékeik formázására, illetve refaktorálására, hogy levegyük az osztályt használó programozó válláról azt a kurzor tömbbé való iterálásának terhét ezáltal csökkentve a hibalehetőséget.

A MongoDB kötetlen, ezáltal sok hiba előfordulhat, a jövőben egy séma definíciós funkció is bevezetésre kerülhet az egy kollekciókban található adatok túlzott eltérésének elkerülése érdekében, ez enyhítheti az adatbázisunkat karbantartó személy munkakörülményeit, lehetővé téve a kötetlen munkát, ha ezt a feladatot két külön személy tölti be, bár szem előtt tartva azt hogy az adatbázis rendszer legnagyobb erőssége ez a kötetlenség, sémán felüli plusz mezők hozzáadását továbbra is lehetővé tenné.

# Felhasznált irodalom

PHP hivatalos dokumentáció - https://secure.php.net/manual/en/

Docker hivatalos dokumentáció - https://docs.docker.com/

MongoDB getting started - https://resources.mongodb.com/getting-started-with-mongodb

Github chear sheet - https://github.github.com/training-kit/downloads/github-git-cheat-sheet.pdf

Materialize CSS hivatalos dokumentáció - https://materializecss.com/

Ubuntu csomag lista - https://packages.ubuntu.com/

Ubuntu hivatalos dokumentáció - https://help.ubuntu.com/stable/ubuntu-help/index.html

MongoDB használata PHP alatt - http://zetcode.com/db/mongodbphp/

Docker-compose dokumentáció - https://docs.docker.com/compose/

Animate.css github oldala - https://github.com/daneden/animate.css

Highlight JS dokumentáció - https://highlightjs.readthedocs.io/en/latest/

Beerware licenc feltételek - https://fedoraproject.org/wiki/Licensing/Beerware

MySQL Docker - https://hub.docker.com/\_/mysql

PHP Docker - https://hub.docker.com/\_/php