BGSzC Pestszentlőrinci Közgazdasági és Informatikai Szakgimnáziuma

1184 Budapest Hengersor 34.

1. Záró dolgozat

SiteEz

Konzulens tanár: Készítette:

Bencze István Hulitza Csaba Máté

Tartalom

[1 Bevezetés 3](#_Toc100233878)

[1.1 Feladat leírás 3](#_Toc100233879)

[1.2 A felhasznált ismeretek 3](#_Toc100233880)

[1.3 A felhasznált szoftverek 3](#_Toc100233881)

[2 Felhasználói dokumentáció 5](#_Toc100233882)

[2.1 A program általános specifikációja 5](#_Toc100233883)

[2.2 Rendszerkövetelmények 5](#_Toc100233884)

[2.2.1 Hardver követelmények 5](#_Toc100233885)

[2.2.2 Szoftver követelmények 5](#_Toc100233886)

[2.3 A program telepítése 6](#_Toc100233887)

[2.4 A program használatának a részletes leírása 8](#_Toc100233888)

[3 Fejlesztői dokumentáció 13](#_Toc100233889)

[3.1 Az alkalmazott fejlesztői eszközök 13](#_Toc100233890)

[3.2 Adatmodell leírása 13](#_Toc100233891)

[3.3 Részletes feladatspecifikáció, algoritmusok 15](#_Toc100233892)

[3.4 Tesztelési dokumentáció 17](#_Toc100233893)

[4 Összefoglalás 21](#_Toc100233894)

[4.1 Önértékelés 21](#_Toc100233895)

[4.2 Továbbfejlesztési lehetőségek 21](#_Toc100233896)

[5 Felhasznált irodalom 23](#_Toc100233897)

[6 Ábrajegyzék 25](#_Toc100233898)

# Bevezetés

## Feladat leírás

A SiteEz című weblapomat azért hoztam létre, hogy tesztelni tudjam a mostani nyelvek rugalmasságát és kezelhetőségét. A weblapommal egyszerű weblapokat lehet készíteni, azzal a céllal, hogy próbálja ki a átlag ember a weboldal készítés varázslatosságát vagy esetleg hogy kipróbáljon egy új program nyelvet.

## A felhasznált ismeretek

Bootstrap - <https://getbootstrap.com/docs/5.1/getting-started/introduction/>, a CSS keretrendszer, amit használok, segítségével előre elkészített HTML osztályokkal lehet formázni könnyedén a weblapot, annak dokumentációja.

ReactJS - <https://reactjs.org/docs/getting-started.html>, a Javascript könyvtár, amit használok, ami nagyban elősegíti a Javascript alapú front-end webkészítést, annak dokumentációja.

React Router - <https://reactrouter.com/docs/en/v6>, a ReactJS egy könyvtár toldása, amiben a Router funkció elérhető, annak dokumentációja.

Stack Overflow - <https://stackoverflow.com/>, weboldal, ahonnan a kódomhoz nyújtottak segítséget emberek az interneten.

## A felhasznált szoftverek

Visual Studio Code - <https://code.visualstudio.com/>, a Microsoft ingyenes kódfejlesztő programját használtam, egyszerű és átlátható, könnyen kezelhető és sok bővítmény letölthető hozzá.

XAMPP - <https://www.apachefriends.org/index.html>, XAMPP egy szoftver gyűjtemény, amit álltalában PHP fejlesztéshez használnak, de én csak a MySQL szerver részét használtam.

GitHub Desktop - <https://desktop.github.com/>, a GitHub álltal fejlesztett asztali alkalmazás, amivel hozzá tudsz férni a repository-hoz, le tudod klónolni, fetchelni az adatait, és még commitolni és pusholni a mostani verziót, a GitHub weboldal bővítménye.

NodeJs - <https://nodejs.org/en/>, a NodeJS alapján működik az egész letöltés, telepítés és a szerver is. Létfontosságú komponens és szoftver.

# Felhasználói dokumentáció

## A program általános specifikációja

A weboldal képes kevés programozói tudással egy basic statikus WEB 1.0 weboldalt készíteni. Kipróbálni a weblapfejlesztés varázsát vagy gyakorlásra tökéletes. A folyamathoz nem kell más, mint egy kis szöveg szerkesztés tudás, és egy kis fantázia vagy cél. Ha van egy kis Markdown tudás is, akkor az is jó, mert tudniillik arra a nyelvre van alapozva. A weboldal használata egyszerű, minden le van írva az oldalon. Van FAQ oldal is, ha a felhasználónak lenne kérdése vagy kéne segítség, és még gyors segítségek is elérhetőek.

## Rendszerkövetelmények

### Hardver követelmények

Windows:

* Windows 7, 8, 8.1, 10 vagy későbbi verzió,
* Intel Pentium 4 vagy újabb processzor, ami támogatja az SSE3-as technológiát.

Mac:

* Mac OS X El Capitan 10.11 vagy újabb verzió.

Linux:

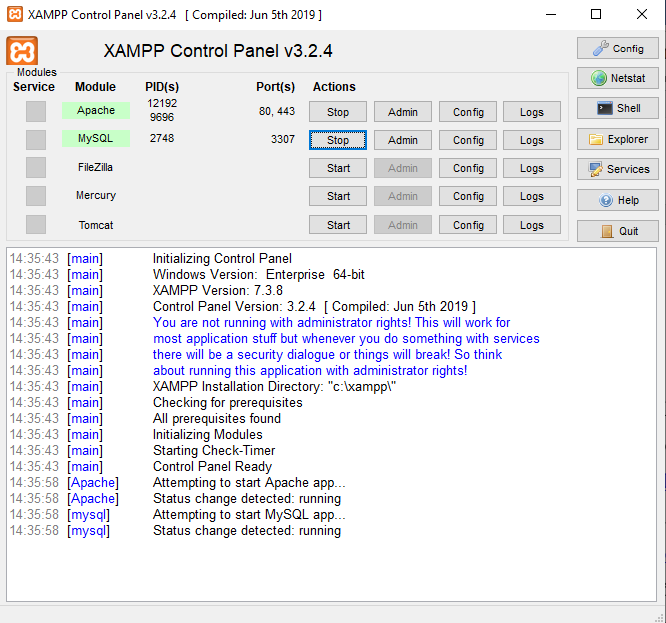
* 64-bites Ubuntu 18.04+, Debian 10+, openSUSE 15.2+, vagy Fedora Linux 32+,
* Intel Pentium 4 vagy újabb processzor, ami támogatja az SSE3-as technológiát.

### Szoftver követelmények

* Újabb verziójú webböngésző, ami támogatja a Javascriptet. (pl. Google Chrome, Microsoft Edge)
* XAMPP, Apache és MySql ebből. Ezek szükségesek a szerver hozzáféréshez.
* NodeJS programrendszer a számítógépen.
* Minden komponens az apphoz és szerverhez, minden benne van a „client” és „server” mappákban, letöltés lejjebb.
* „siteez.sql” fájl, ami tartalmazza az üres adatbázist és táblákat.

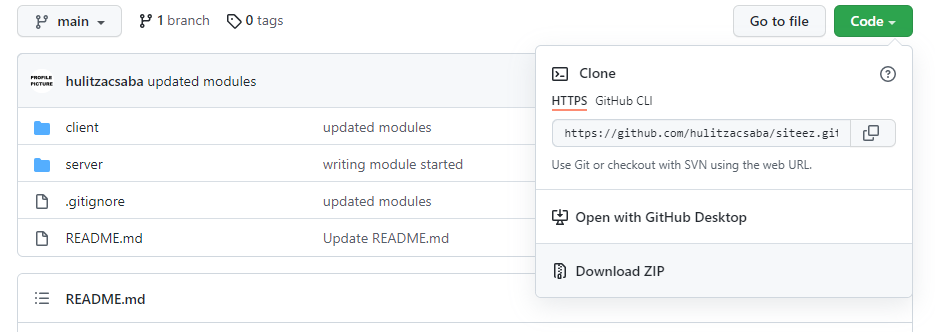
## A program telepítése

Az első lépések közé tartozik a csatolmány, függő programok feltelepítése. Itt a következő programokra gondolok. A Node.js kötelező a program zökkenőmentes használatához. Letöltés innen érhető el (<https://nodejs.org/en/download/>), az LTS kiadás bőven elég a célunkra, az operációs rendszernek megfelelő verziót töltsük le. A részeletes telepítését itt lehet megtalálni (<https://nodejs.dev/learn/how-to-install-nodejs>), módosítás nem kell. Következőnek, a XAMPP letöltése és telepítése szükséges, ez kötelező a weboldal és adatbázis kapcsolatához. Innen letölthető (<https://www.apachefriends.org/download.html>), a legújabb verzió tökéletes. A telepítés folyamata közben mikor kérdezi, hogy mely kiegészítőket kérjen a telepítés mellé, akkor a Xampp, MySQl és Apache komponenseket kell legalább feltelepíteni. Az adatbázis kapcsolathoz indítsa el a XAMPP Control Panelt.



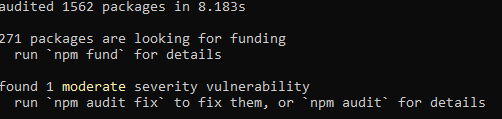
1. kép A XAMPP Control Panel megfelelő beállításai

Ezeket kéne, látnia mikor rendesen van elindítva a MySQL szerver. Ezután menjen a MySQL sor Admin gombjára, és megnyílik majd az sql admin felület. Mikor megnyílik, a felület akkor menjen az importálás fülre. Tallózza be a „siteezdummy.sql” fájlt majd az indítás gombra.

Pár zöld ablakot kéne, látnia a főoldalon, ahova visszadob. Ezzel készen áll az adatbázis. A következő lépés a weblap kicsomagolása vagy letöltése a GitHub-ról. Ha ez első módszert választjuk, akkor a lemezen be lesz csomagolva egy tömörített mappába. Ezt kell kicsomagolni egy mappába a számítógépen. Ha a másikat, akkor meglátogatjuk a GitHub linket és a Code> Download ZIP gomb által letöltjük tömörítve.

2. kép GitHub oldal a letöltő linkkel

Miután kicsomagoltuk, megnyitjuk a CMD-t vagy a Windows PowerShell-t, esetleg szimulált parancssort is használhatunk, mint például a parancssort a Visual Studio Code-ban. Navigáljunk a mappában vagy nyissuk meg a mappában a Shift+Jobb klikk> PowerShell ablak megnyitása itt opcióval. Utána benavigálunk a ’cd’ paranccsal a ’client’ vagy ’server’ mappába, attól függően, hogy melyiket akarjuk elindítani. Ha a kliens oldalt akarjuk, akkor miután benavigáltunk a mappájába, akkor a ’npm install’ paranccsal feltelepítjük a NodeJS modulokat, amik szükségesek a weboldal futásához. Ezt a műveletet elég egyszer elvégezni a kicsomagolás után. Ha a letöltés sikeresen lefutott így:



3. kép Sikeres modul telepítés

Ha sikeresen lefutott, akkor ugyan ebben a mappában, a ’npm start’ paranccsal elindíthatjuk a kliens oldalt. Ezután nem kell nyúlni a parancssorhoz, ha sikeresen elindult, akkor meg kellett volna nyílnia az alapértelmezett böngészőben a ’localhost:port(3000)’ linkkel ellátott weboldal. Így készen áll a kliens rész, de még nem funkcionálna a weboldal, erre kell elindítani a szerver oldal scriptet is. Nyissunk egy új parancssort és navigáljunk be a szerver mappájába. Itt is futtassuk a ’npm install’ parancsot, megvárjuk, még itt is telepíti a modulokat (referáljunk a 3. képre). Ezután futtassuk a ’node index.js’ parancsot. Ha sikerese volt a parancs, akkor fut a szerver és azt kéne kiírnia, hogy „SERVER RUNNING…”. Ha ezt írja ki egyik parancssort se zárja be. Ha minden sikerült, akkor a weboldal készen áll a használatra.

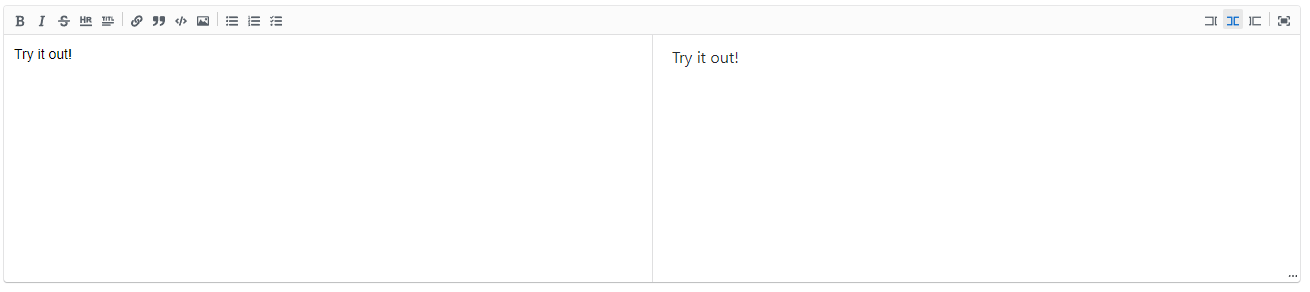
## A program használatának a részletes leírása

A képen szöveg látható

Automatikusan generált leírásMiután megnyitottuk a weboldalt, láthatjuk a weblap felületét, főoldalt, fölül a navigációs menüt és pár funkciót. Megjegyzem hogy az egész weboldal angolul van.

4. kép Főoldal

A főoldalon lehet elérni a weblap fő funkcióját, a weblap készítő komponenst. A weblapkészítő komponens által lehet készíteni egy weblapot, ez a hirdetett funkciója is ennek a projektnek. A főoldal 3 részre tagolódik: a „Head options”-ra, a „Body text editor”-ra és az alsó „Function” részre. Most elmagyarázom mi melyik mit csinál. Az első rész, a „Head options”, a készülendő weblap fej egységének beállításai. A title a weblap címe a böngészőben, a lapok között lehet majd látni az itt beállított szöveget. A noScript, az egy olyan szöveg, ami akkor jelenik meg a böngészőben mikor nincs bekapcsolva a Javascript, általában ez be van kapcsolva minden böngészőben, szóval nem kell nagy figyelmet fordítani felé, de érdemes megadni valamit. A language elég egyértelmű, a weboldal szövegének és keresési elérhetőségének nyelvét adhatjuk meg, ezen a nyelven lesz a weboldal javítása, szövege, keresési beállításai beállítva. Pár nyelv elérhető, a legelterjedtebbeket raktam bele. Az utolsó a character set, ami annyit takar, hogy a weboldal milyen karakterekkel lesz kompatibilis. A legelterjedtebb az UTF-8 de ván pár más opció, ha nem tudod melyiket válaszd akkor válaszd az UTF-8-at.

Ezzel végeztünk a fej beállításokkal, jöhet a fő része a weblapnak, a test. A test, ami megjelenik a képernyőn, a tartalma a weblapnak. Ennek szerkesztéséhez van itt a „Body editor” -ban a Markdown alapú szöveg szerkesztő.

5. kép MD editor

A Markdown az egy szöveg alapú formázható formátum, amit elég sok helyen használnak. A szintaxról olvashatsz itt (<https://www.markdownguide.org/basic-syntax/>). Ha van elképzelésed, akkor próbálhatsz szöveget írni, leírni amit szeretnél a weblapon. Az editor kettő részre van osztva, bal része ahova írsz, a jobb oldalon meg megjelenik majd a formázott szöveg. A billentyűzettel írhatunk a mezőbe, dupla enterrel meg tudunk új sort kezdeni. Ha írtál szöveget, akkor kipróbálhatod a formázásokat. Az editor ablak tetején találod a formázásokat. 2 féle képpen tudod használni. Vagy megnyomod a formázáshoz megfelelő gombot és a megjelent jelek közé írod a formázni kívánt szöveget, vagy kijelölsz egy szövegrészletet és utána nyomod meg a gombot. Ezáltal a szöveg formázva lesz, a jobb oldalon láthatod is. Ezután csak megírhatod a szöveged, közben nézheted a dokumentációt, hogy sikerüljön egy remek weblapot kreálni. Ha a szöveg elkészült, akkor végezd el az utolsó simításokat, és menjünk az utolsó részre.

Az utolsó rész, hogy végig nézd a szöveget, beállításokat, biztos így akarod-e, biztos jó-e stb. Ha nem tetszik akkor változtass rajta, mert még lehet. Ha készen állsz akkor nyomd meg a kék gombot a weblap alján. Ilyenkor a szerver elvégzi a konvertálást, és majd felkínálja a fájlt letöltésre, vagy a linket a weblap fájlához. Készen is van a weblap.

A weboldalon van még kettő lap amit meg kell néznünk. Az első a FAQ oldal, elérhető a navigációs menüből fönt, csak rá kell kattintani.

A képen szöveg látható

Automatikusan generált leírásA FAQ oldal egy olyan oldal, ami nem lát semmi mechanikai funkciót, viszont szöveget tartalmaz. A FAQ vagy Frequently Asked Questions vagy GYIK egy olyan oldal, ami a legtöbbször kérdezett kérdésekre ad vagy próbál választ adni. A szöveg egy pár percnyi olvasni való csak és fontos információkat tartalmaz, szóval érdemes elolvasni.

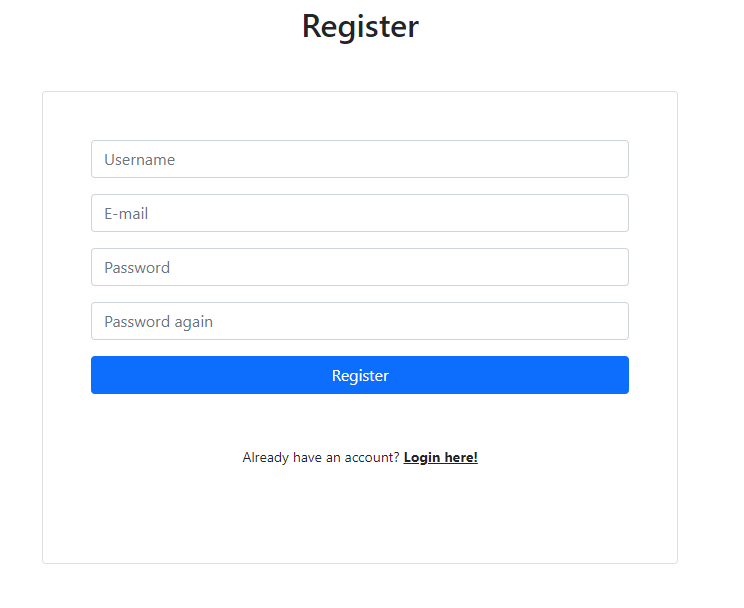
6. kép FAQ Oldal

A képen szöveg látható

Automatikusan generált leírásA harmadik oldalak a Profile opció kattintásával érhető el. Ez az oldal egy bejelentkezés lapot nyit meg, ahol be lehet jelentkezni a weblapra, hogy elérjük a weblap mentési funkciót, ami elmenti a weblapot egy adatbázisba, későbbi hozzáféréshez.

7. kép Login oldal

Itt kettő mezőt és egy gombot találunk. A felső menőbe kell írni a felhasználónevet, a másodikba pedig a jelszót. Ha beírtuk mindkettőt akkor menjünk a nagy kék gombra. Ilyenkor be kéne jelentkeztetnie a lapnak és ki kéne írnia a felhasználónevet, amivel bejelentkezett. Ha nem így történt akkor próbálja meg újra és ha megint nem sikerült, akkor írjon egy hibajelentést. De ha sikerült akkor kiélvezheti az extra funkciót. Ha a szöveg ott marad a mezőkben ne ilyedjen meg. Csak frissítsen az oldalra és eltűnnek, de nem jelentenek problémát, se funkcionalitásban, se biztonságban. Ha viszont nincs fiókja akkor regisztrálni kell egyet. A regisztrációs lapra el tud menni a „Create an account” gombbal.

8. kép Register oldal

A regisztrációs oldalon lehet regisztrálni fiókot. A regisztráció rendkívül egyszerű. Írjuk be a kívánt felhasználónevet felülre, utána az e-mail címet, utána a jelszavat és újra jelszavat. Ha megnéztük és beírtuk kívánt adatainkat nyomjunk a nagy gombra. Ezután a rendszer regisztál minket. Ha a szöveg ott marad a mezőkben ne ilyedjen meg. Csak frissítsen az oldalra és eltűnnek, de nem jelentenek problémát, se funkcionalitásban, se biztonságban. Ha megtette, akkor rámehet a „Login here” szövegre, hogy visszamenjen a bejelentkezésre, és hogy meg tudja próbálni a bejelentkezést. Ha úgy látja, hogy nem sikerült akkor írjon egy hibajelentést.

Ha sikeres a bejelentkezés akkor a Login oldalon és a főoldalon is írnia kéne a bejelentkezett felhasználót.

A weblap funkciójai ezek, minden könnyen navigálható, de ha van valami hiba/javaslat akkor lehet írni hibajelentést GitHub-on.

Sok szerencsét a weblap készítéshez.

# Fejlesztői dokumentáció

## Az alkalmazott fejlesztői eszközök

Fejlesztéshez legfőbbképpen a Visual Studio Code-t (<https://code.visualstudio.com/>) használta, ebben írtam az összes kódom, ami megtalálható a projektben, továbbá a virtuális parancssor funkcióját is előszeretetten használtam. Programozási nyelvemnek a Javascipt-et használtam, abból is az JSX és ES6-os kiadást. A legtöbb prototype parancs ebben található, amik nagy mennyiségben segítettek a projektben. A Javascript-re jött a ReactJS (<https://reactjs.org/>) nevű keretrendszer, ami az egész front-end működését biztosítja. A back-end/server működését a NodeJS (<https://nodejs.org/en/>) biztosítja. Az adatbázis kapcsolat és fenntartását a XAMPP (<https://www.apachefriends.org/hu/index.html>) szoftvercsomag segíti elő, ebben a MySQL szerver funkcionalitás miatt fut az adatbázis és a back-end – adatbázis kapcsolat. Ebben a csomagban a PHPMyAdmin felületet használtam az adatbázis létrehozásához.

## Adatmodell leírása

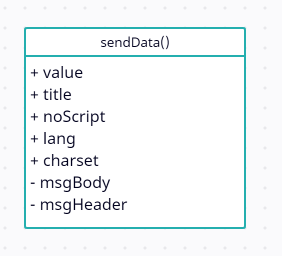
A képen szöveg látható

Automatikusan generált leírásA weboldal pár funkcionalitása alatt egy MySQL adatbázis található, ami számos adatot tárol a felhasználóról és a weblapokról, amit készítenek. Az adatbázis kettő táblából áll.

. kép Adatbázis felépítése kapcsolattal

Az első tábla „user” egy adatbázis, ami a regisztrált felhasználókról tárol adatokat, mint például felhasználó nevet, jelszót (bcrypt), email-t. Továbbá egy id-t, amit használ egy másik tábla. A weboldal innen kéri le a felhasználók listáját bejelentkezéskor, hogy be tudjon jelentkeztetni. A másik tábla a „sitesave” egy tábla, ami az elmentett weboldalakat fogja tárolni. Ez azt jelenti, hogyha egy felhasználó be van jelentkezve a felhasználói fiókjába, és csinál, egy weboldalt akkor a weboldal automatikusan elmenti a weboldalát a táblába, névvel és linkkel együtt, ami egy link, ami egy Pastebin link lesz, feltöltve a weboldal tartalmával. Tehát ez a tábla kötve van a másikkal egy userId foreign key-en keresztül, hogy a weboldal le tudja kérni a felötlött weboldalakat a profil fülön.

Egy másik adatszerkezet, amit alkalmazok az a weboldal adatainak elküldéséhez alkalmazott kapcsolat és adatszerkezet, amit elküld a front-end a back-endnek.



. kép Készülendő weblap adatszerkezete

A sendData funkció küldi az adatokat a szervernek, de most nem is ez fontos. A fontos rész, az adatok, amiket küld. A title, value, noScipt, charset, lang az mind publikus a front-end részen, ezek a változóknak mindegyiknek megvan a set fele is, amin keresztül változik értékük. Mikor a küldés történik, akkor az adatokat abból a publikus változókból begyújti. A title, noScipt, charset és lang változók a weboldal meta tag-jeit képviselik majd a kész lapon. Ezeket a változókat összefűzi a msgHeader változóba, ami persze privát ezért nem lehet hozzáférni, továbbá a msgBody-ba meg a value értéke kerül. Utána ezt a kettő változót küldi tovább a program összekapcsolva, a kettő változó végül a készülendő weblap adatait tartalmazza, tehát egy adatszerkezet.

Még kettő adatszerkezet van, de ezek kisebbek és nem teljesen szóra méltóak. A bejelentkezés és regisztrációnál van kettő adatstruktúra, ami 2-3 változóból ál, amik vagy a felhasználónév, jelszó vagy pluszban az e-mail címet tárolja, és ez megy a szerverre, a regisztráció vagy bejelentkezés funkciónál.

## Részletes feladatspecifikáció, algoritmusok

A program számos függvény építi fel, mind szinkronba működve a weboldal működéséhez megfelelően.

A képen szöveg látható

Automatikusan generált leírásRegister() függvény:

11. kép Register függvény

A Register függvény 4 változóval rendelkezik. A username és email azok egyértelműek, a regisztrációnál bekért felhasználónév és e-mail tárolására van. A másik kettő, a password és password2, a két jelszó, amit szintén tárol, és összehasonlítja, hogy ugyan azok e. A függvény csak akkor fut le, ha ugyan azok, ha nem akkor egy errort kap a felhasználó. Ha lefut, akkor az Axios elküldi egy POST paranccsal a szerver felé a három adatot, ami utána küld vissza egy választ, amit logol a weboldal.

Login() függvény:

A Login függvény, hasonlóan működik, ezért csak egy gyors magyarázat. Itt csak kettő változó van, a username és password. Mindkettőt elküldi az Axios egy POST paranccsal, és utána vár egy választ, amit felhasznál a függvény. Eltárolja a választ, és ha van ilyen felhasználó, akkor azt is eltárolja.

A képen szöveg látható

Automatikusan generált leírásuseEffect() függvény:

12. kép useEffect függvény

A useEffect függvény, ami a ReactJS része, egy függvény, ami lefut, ha valami történik az oldalon. Ez a példánya annak a függvénynek mindig lefut, mikor betölt az oldal, ahol elhelyezkedik. A függvény Axios-sal küld egy GET parancsot, ami vár vissza egy választ a login úton. És ha a visszakapott válasz, ami egy bejelentkezés adatszerkezet, akkor megnézi, hogy be van e jelentkezve valaki.

A képen szöveg látható

Automatikusan generált leírássendData() függvény:

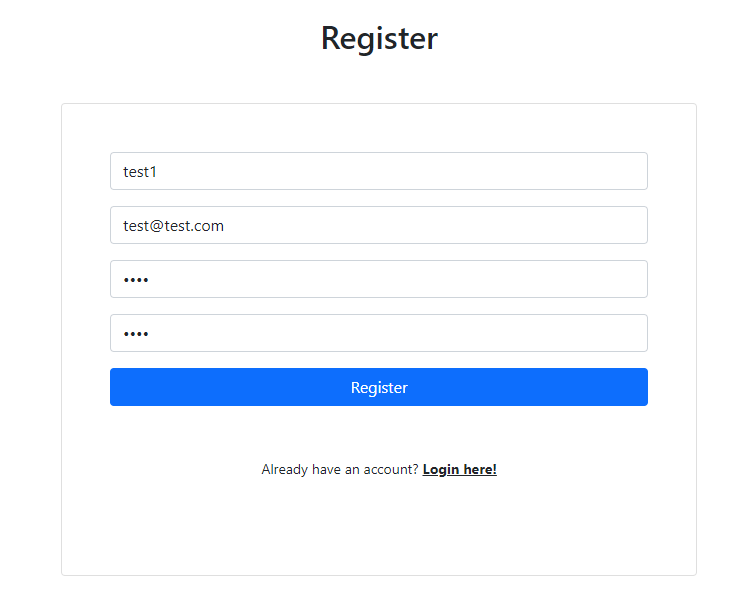
13. kép sendData függvény

A sendData függvényről már volt szó az adatszerkezet leírási fejezetnél. Itt viszont a kód lényegét fogom elmagyarázni. Először is az adatokat bekéri is összefűzi, mint ahogy volt róla szó korábban. A függvényben van egy másik privát függvény, a send(), ami a következőt csinálja. Axios használatával elküldi az összefűzött adatokat a getData úton, utána vár egy választ a szervertől. Ha kap egy választ, ami esetlegesen ’undefinied’ akkor futassa a függvényt újra, de ha nem ’undefinied’ akkor csak logolja a választ. A függvény addig nem fut le, míg előtte nem megy át egy error check-en a parent függvény. Megnézi, hogy minden bekért adat nem üres, vagy null vagy ’undefinied’. Ha mindegyik érték megfelelt, akkor megnézi, hogy ezeknek az adatoknak nem ’undefinied’ e típusuk, mert akkor használhatatlanak lennének a szerver számára. Ha jó volt, akkor fűz. Utána megnézi, hogy az összefűzött msgHeader változó nem üres vagy null, vagy ’undefinied’ e. Ha az, akkor dobjon hibát, ha nem akkor futassa a send() függvényt.

## Tesztelési dokumentáció

A tesztelést kettő részre bontom. A bejelentkezés/regisztráció részre, és a weboldal készítő komponens részre.

Regisztráció tesztelés: normális körülmények, készítő álltal.

Kimenetel: sikeres.

A bal oldalon látható az elküldött adat, és alul a megkapott adat az adatbázisban. Hiba nélkül lefutott, bár az adatok ott maradtak a form-ban, de nem probléma. A jelszó titkosítva megy.

14. kép Reg-teszt

Regisztrációs teszt: hatalmas adatokkal, készítő álltal.

Kimenetel: sikeres. Kép nem kell.

Az adatbázis tudta fogadni az adatokat, mind a jelszó, felhasználónév, és email rendkívül hosszú volt, 50+ karakter. A jelszó titkosítva ment, ezt is tárolta.

A képen szöveg látható

Automatikusan generált leírásBejelentkezés teszt: normális körülmények, készítő álltal.

15. kép Login-teszt

Kimenetel: sikeres

Fölül az elküldött adat és a kapott válasz, sikeres volt a bejelentkezés. Szintén maradt az adat a form-ban de nem probléma itt sem.

Bejelentkezés teszt: hatalmas adatokkal, készítő álltal.

Kimenetel: sikeres. Kép nem kell.

A weboldal be tudta jelentkeztetni a rendkívül hosszú felhasználó nevet, és el tudta tárolni.

Ez volt a profil része a tesztelésnek, most jön a weblap komponens.

Weblap tesztelés: normális szöveg, készítő álltal.

A képen szöveg látható

Automatikusan generált leírás

16. kép Weblap-teszt

Kimenetel: sikeres.

Minden adat beleépült a weboldalba és a szöveg is megjelenik a sima HTML stílusban. A title, noScript szöveg is tökéletesen működik. A szöveg maradt, de lehet, hogy ez nem is hiba, hátha akarja szerkeszteni a felhasználó.

Weblap tesztelés: brutálisan nagy szöveg, készítő álltal.

Kimenetel: sikeres, hibákkal.

A legtöbb adat átalakult a kívánt formátummá, mivel a szöveg, amit használtam az a CommonMD kiegészített Markdown formátumban volt, ezért a konvertáló komponens egy kicsit várhatóan, kihagyott/nem értett részeket. De a java része rendkívül sikeresen beépült a weblapba. Elfogadható a kereteken belül, mert a konvertáló más syntax-ra lett állítva, a weboldal funkcióját el tudja látni ahogyan elképzeltem.

# Összefoglalás

## Önértékelés

Összeségben ez a projekt segített megérteni, hogy milyen dolgozni egy olyan projekten, aminek minden részét én magam terveztem, készítettem, teszteltem, írtam, satöbbi. Az idő attól függetlenül, hogy pár hónap terjedelmű volt, az idő ettől függetlenül véges volt, amit véleményem szerint jobban be tudtam volna osztani. A projektből, amit eredetileg akartam, ki tudtam hozni, az extrákat, amit terveztem, és próbáltam beletömni, annak csak egy rész valósult meg sajnálatos módon. A design is kérhetett volna még magára egy kis figyelmet, de sajnos nem tudtam rá szánni. Ha értékelném kéne a munkámat, akkor nem a legjobb, lehetett volna több erőt fektetnem bele, de sajnos más okok és programozással kapcsolatos kiégés miatt csak ennyi sikerült.

## Továbbfejlesztési lehetőségek

Dinamikus profil és link mutatás:

Az egyik célja a profil oldalnak nem csak a bejelentkezés lett volna. A profil fül egy teljesen személyes, gyűjtemény lett volna a bejelentkezett felhasználónak. A weboldal menti a weblapokat, amit csináltak, és el is küldi a felhasználónak, de eredetileg a terv az lett volna, hogy feltölti a linkeket egy adatbázisba, és ha a felhasználónak vannak feltöltött linkjei, akkor ki rakta volna az összes előző munkáját a profil fülre, ami dinamikusan megjelenítette volna.

Felújított design:

A bootstrap alapvetően egy letisztult, egyszerű használható stíluslap gyűjtemény, ami segítségével könnyen készíthető reszponzív weblap. A design is elég jó, elvárt szinten van, és én ezt használom a weblapomon. Akartam volna egy saját design-t viszont mellé, ami egy kicsit könnyebbé és átláthatóbbá tette volna a weblapot, de sajnos nem fért bele az időbe, ezért ebbe a kategóriába tökéletes.

Sima, hiba mentes bejelentkezés/regisztráció:

A bejelentkezés és regisztrációs rendszer működőképes, minimális mennyiségű hibával. Sajnos a rendszer, mégis, néha, megoldja és dob egy hibát. Az első dolog, amit ezzel csinálnék, hogy visszairányítana a main oldalra, vagy csak kitörölné az adatokat a form-ból, hogy egy kicsit szebb legyen. Ezt mind a kettő esetben, a bejelentkezés és regisztrációnál is megcsinálnám. Továbbá, hibamentessé csinálnám az egészet, mert a bejelentkezés legtöbb esetben hibát ír ki, Bár le van kezelve, és további utasításokat ír a felhasználónak, örülnék, ha egyáltalán, vagy csak ritkán történne meg ez az eset. Ezért ez is egy javítandó lehetőség.

Több opció:

A legtöbb opció, ami fontos elérhető a weboldal készítésnél. Bár igazat megvallva, a mostani legalapabb weblapon is több dolgot állítanak be, mint például a stíluslapot, alap scripteket, satöbbi. Ez lenne az egyik fejlesztési opció, hogy a weboldal készítésnél, több, jobb és érthető opciók lennének, hogy még jobban személyre tudjad szabni a weblapokat, amit készítész. Továbbá hozzá lennének adva a kísérleti opciók, amik akár katasztrofális hatással lennének a weblapra, ezek az opciók eredetileg regisztrációnál lettek volna lehetővé téve a felhasználó felé.

# Felhasznált irodalom

<https://reactjs.org/docs/getting-started.html> - ReactJS dokumentáció

<https://reactrouterdotcom.fly.dev/docs/en/v6> - React Router dokumentáció

<https://axios-http.com/docs/intro> - Axios dokumentáció

<https://nodejs.org/en/docs/> - NodeJS dokumentáció

<https://getbootstrap.com/docs/5.1/getting-started/introduction/> - Boostrap dokumentáció

<https://stackoverflow.com/> - Stack overflow kód megosztó weboldal

<https://uiwjs.github.io/react-md-editor/> - React MD editor by uiwjs dokumentáció

<https://www.npmjs.com/package/pastebin-ts> - Pastebin-Ts by j3lte dokumentáció

<https://www.youtube.com/watch?v=W-sZo6Gtx_E> – PedroTech - Register and Login Page Tutorial | ReactJs, NodeJS, MySQL (Beginner) videó

<https://www.youtube.com/watch?v=sTHWNPVNvm8> – Pedro Tech – Register and Login Tutorial | ReactJs, NodeJS, MySQL | Cookies, Sessions, Hashing videó

<https://nodejs.org/api/fs.html> - NodeJS FileStream dokumentáció

<https://nodejs.dev/learn/the-nodejs-fs-module> - NodeJS Fs modul dokumentáció

<https://www.w3schools.com/js/default.asp> - Javascript segítség

<https://www.w3schools.com/mysql/default.asp> - MySQL segytésg

<https://www.w3schools.com/react/default.asp> - React segíttség

# Ábrajegyzék

[1. kép A XAMPP Control Panel megfelelő beállításai 6](#_Toc99719951)

[2. kép GitHub oldal a letöltő linkkel 7](#_Toc99719952)

[3. kép Sikeres modul telepítés 7](#_Toc99719953)

[4. kép Főoldal 8](#_Toc99719954)

[5. kép MD editor 9](#_Toc99719955)

[6. kép FAQ Oldal 10](#_Toc99719956)

[7. kép Login oldal 11](#_Toc99719957)

[8. kép Register oldal 11](#_Toc99719958)

[9. kép Adatbázis felépítése kapcsolattal 13](#_Toc99719959)

[10. kép Készülendő weblap adatszerkezete 14](#_Toc99719960)

[11. kép Register függvény 15](#_Toc99719961)

[12. kép useEffect függvény 16](#_Toc99719962)

[13. kép sendData függvény 16](#_Toc99719963)

[14. kép Reg-teszt 17](#_Toc99719964)

[15. kép Login-teszt 18](#_Toc99719965)

[16. kép Weblap-teszt 19](#_Toc99719966)