

BGSzC Pestszentlőrinci Közgazdasági és Informatikai Szakgimnáziuma

1184 Budapest Hengersor 34.

Záró dolgozat

SiteEz

Konzulens tanár:

Bencze István

Készítette:

Hulitza Csaba Máté

Tartalom

1	Bevezetés	3
1.1	Feladat leírás	3
1.2	A felhasznált ismeretek	3
1.3	A felhasznált szoftverek	3
2	Felhasználói dokumentáció.....	5
2.1	A program általános specifikációja.....	5
2.2	Rendszerkövetelmények	5
2.2.1	Hardver követelmények	5
2.2.2	Szoftver követelmények.....	5
2.3	A program telepítése.....	6
2.4	A program használatának a részletes leírása	8
3	Fejlesztői dokumentáció	13
3.1	Az alkalmazott fejlesztői eszközök	13
3.2	Adatmodell leírása	13
3.3	Részletes feladatspecifikáció, algoritmusok.....	15
3.4	Tesztelési dokumentáció.....	17
4	Összefoglalás	21
4.1	Önértékelés	21
4.2	Továbbfejlesztési lehetőségek	21
5	Felhasznált irodalom	23
6	Ábrajegyzék	25

1 Bevezetés

1.1 Feladat leírás

A SiteEz című weblapomat azért hoztam létre, hogy tesztelni tudjam a mostani nyelvek rugalmasságát és kezelhetőségét. A weblappal 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.

1.2 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.

1.3 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 általában PHP fejlesztéshez használnak, de én csak a MySQL szerver részét használtam.

GitHub Desktop - <https://desktop.github.com/>, a GitHub által fejlesztett asztali alkalmazás, amivel hozzá tudsz férni a repository-hoz, le tudod klónozni, 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.

2 Felhasználói dokumentáció

2.1 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.

2.2 Rendszerkövetelmények

2.2.1 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:

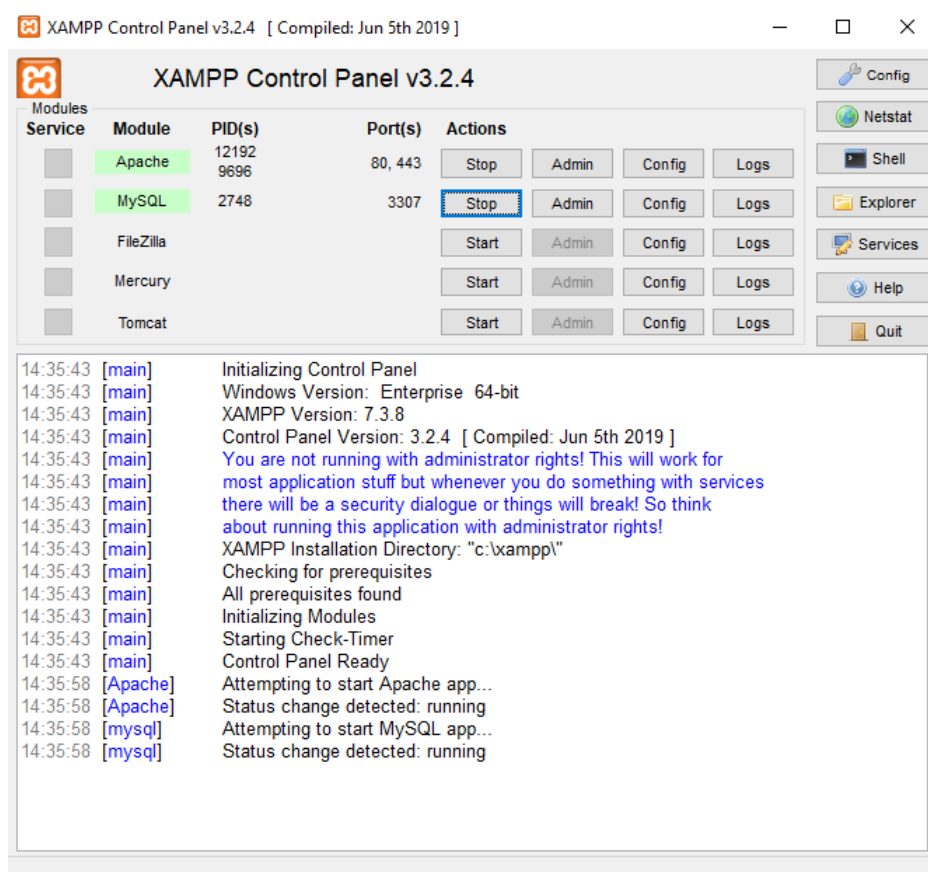
- 64-bites Ubuntu 18.04+, Debian 10+, openSUSE 15.2+, vagy Fedora Linux 32+,
- Intel Pentium 4 vagy újabb processzor, ami támogatja az SSE3-as technológiát.

2.2.2 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.

2.3 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 részletes 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 el kell indítani a XAMPP Control Panelt.

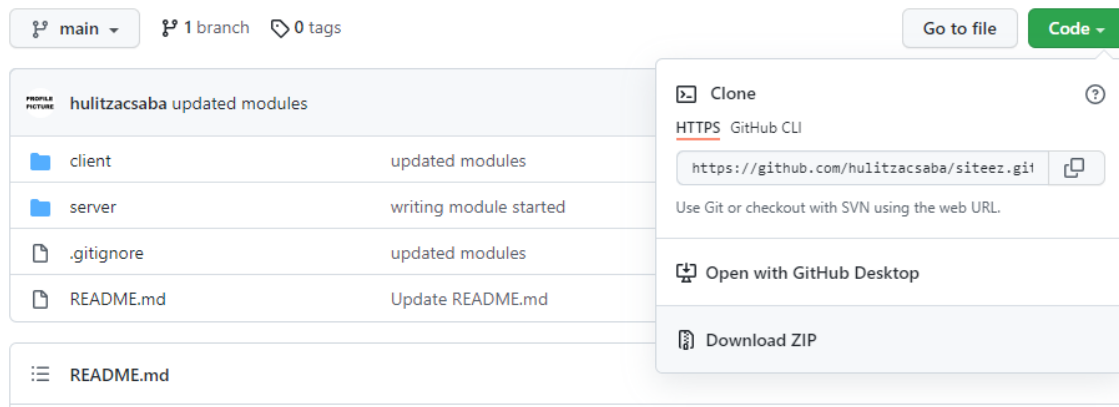


1. kép A XAMPP Control Panel megfelelő beállításai

Ezeket kéne, látnia mikor rendesen van elindítva a MySQL szerver. Ezután menjünk a MySQL sor Admin gombjára, és megnyílik majd az sql admin felület. Mikor megnyílik, a

felület akkor menjünk az importálás fülre. Tallózzuk be a „siteezdummy.sql” fájlt majd az indítás gombra.

Pár zöld ablakot kéne, látnunk 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:

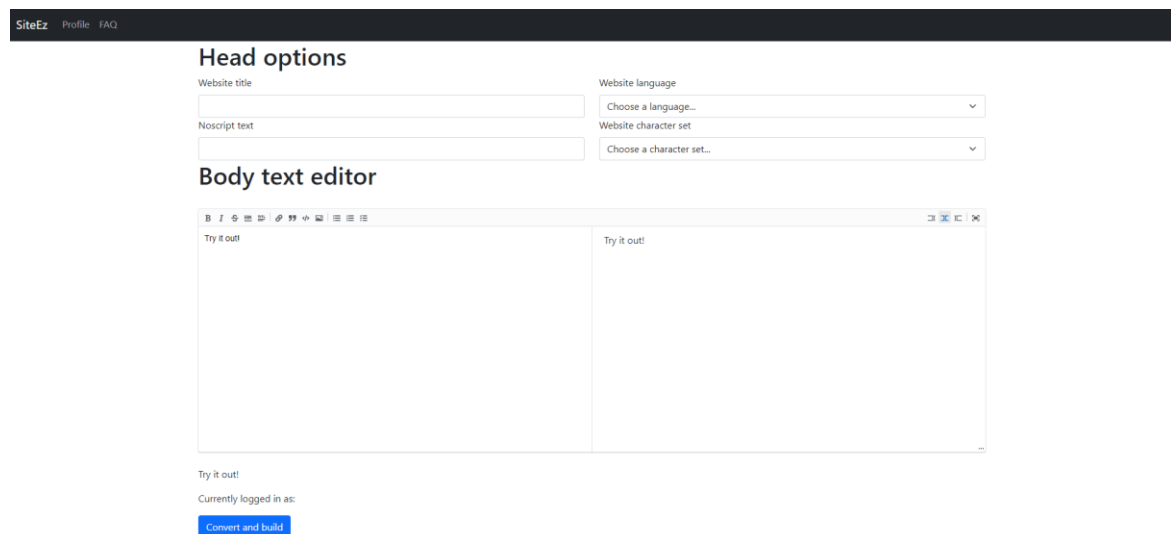
```
audited 1562 packages in 8.183s
271 packages are looking for funding
  run `npm fund` for details
found 1 moderate severity vulnerability
  run `npm audit fix` to fix them, or `npm audit` for details
```

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: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 sikeres 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.

2.4 A program használatának a részletes leírása

Miután megnyitottuk a weboldalt, láthatjuk a weblap felületét, főoldalt, fölül a navigációs menüt és pár funkciót. Az említett funkciók az opciók, a szerkesztő és a navigációs menü. A weboldal angol nyelvű, mert a fogalmazás és készítése így egyszerűbb volt számomra.

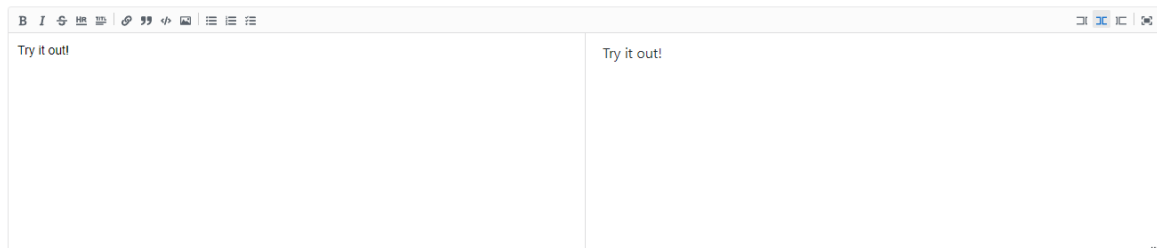


4. kép Főoldal

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 van pár más opció, ha nem tudjuk, 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

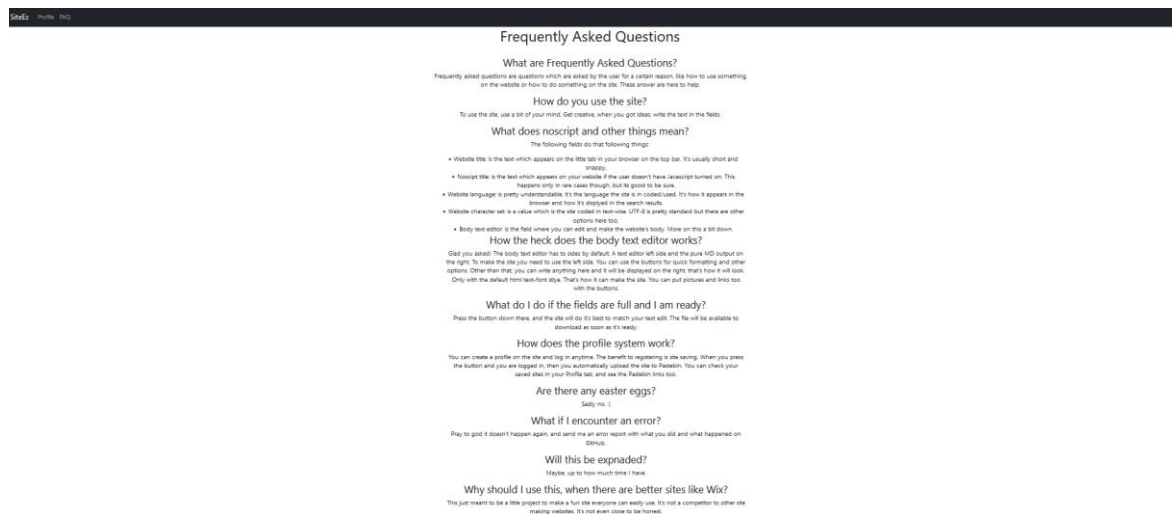
A Markdown az egy szöveg alapú formázható formátum, amit elég sok helyen használnak. A szintaxról olvashatsunk itt (<https://www.markdownguide.org/basic-syntax/>). Ha van elképzelés, akkor próbálhatunk szöveget írni, leírni, amit szeretnénk a weblapon. Az editor kettő részre van osztva, bal része ahova írni lehet, a jobb oldalon meg megjelenik majd a formázott szöveg. A billentyűzettel írhatunk a mezőbe, dupla enterrel meg tudunk új sort kezdeni. Ha írtunk szöveget, akkor kipróbálhatjuk a formázásokat. Az editor ablak tetején találjuk a formázásokat. 2 féle képpen tudjuk használni. Vagy megnyomjuk a formázáshoz megfelelő gombot és a megjelent jelek közé írjuk a formázni kívánt szöveget, vagy kijelölünk egy szövegrészletet és utána nyomjuk meg a gombot. Ezáltal a szöveg formázva lesz, a jobb oldalon láthatjuk is. Ezután csak megírhatjuk a szöveged, közben nézhetjük a dokumentációt, hogy sikerüljön egy remek weblapot kreálni. Ha a szöveg elkészült, akkor végezzük el az utolsó simításokat, és menjünk az utolsó részre.

Az utolsó rész, hogy végig nézzük a szöveget, beállításokat, biztos így akarod-e, biztos jó-e stb. Ha nem tetszik, akkor változtassunk rajta, mert még lehet. Ha készen áll, akkor

nyomjuk 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 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 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.

Login

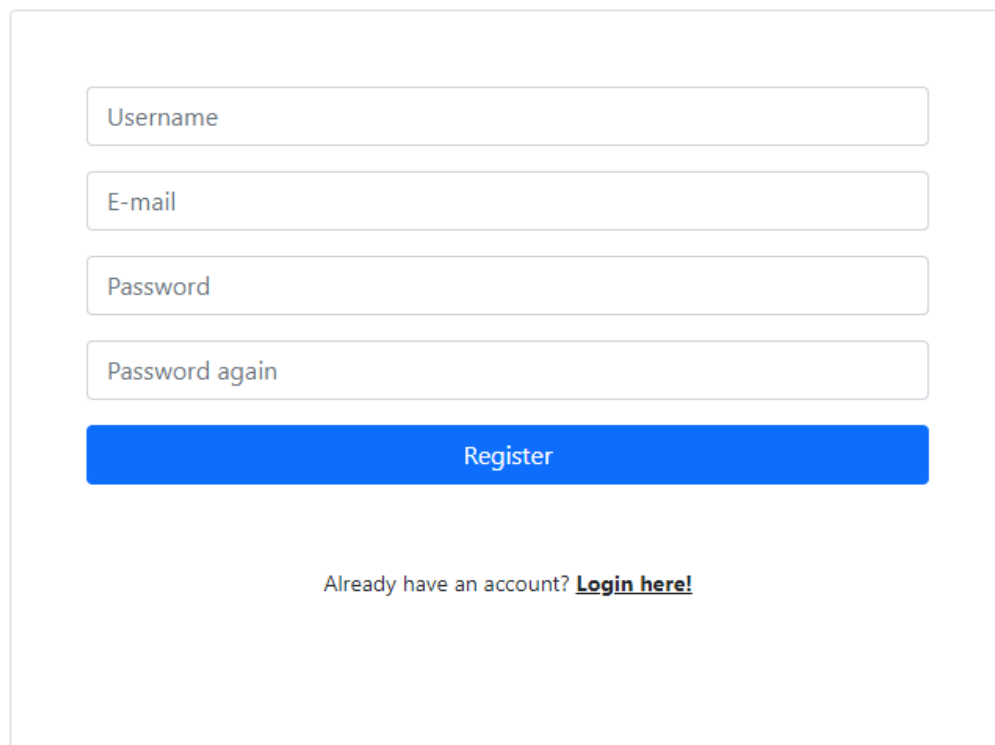
Not Registered? Interested in some extra features? [Create an account!](#)

Currently logged in:

7. kép Login oldal

Itt kettő mezőt és egy gombot találunk. A felső mezőbe kell írni a felhasználónevet, a másodikba pedig a jelszót. Ha beírtuk mindkettőt akkor menjünk a nagy kék gombra. Ilyenkor be kéne jelentkeztenie 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 funkcionalításban, se biztonságban. Ha viszont nincs, fiókja akkor regisztrálni kell egyet. A regisztrációs lapra el tud menni a „Create an account” gombbal.

Register



The image shows a registration form titled "Register". It contains four text input fields stacked vertically, labeled "Username", "E-mail", "Password", and "Password again". Below these fields is a prominent blue button with the text "Register". At the bottom of the form, there is a link that says "Already have an account? [Login here!](#)".

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 regisztrál minket. Ha a szöveg ott marad a mezőkben ne ilyedjünk meg. Csak frissítsen az oldalra és eltűnnek, de nem jelentenek problémát, se funkcionalításban, se

biztonságban. Ha megtettük, akkor rámehetünk a „Login here” szövegre, hogy visszamenjen a bejelentkezésre, és hogy meg tudjuk próbálni a bejelentkezést. Ha úgy látja, hogy nem sikerült akkor írjunk 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.

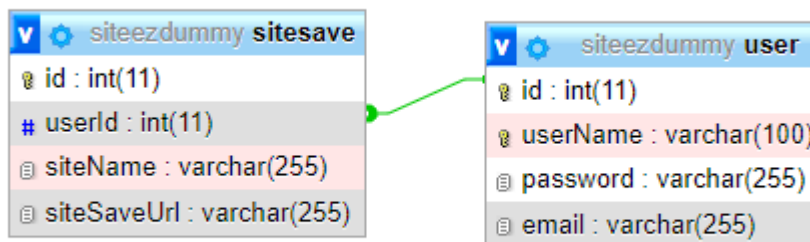
3 Fejlesztői dokumentáció

3.1 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 előszeretettel használtam. Programozási nyelvemnek a Javascript-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.

3.2 Adatmodell leírása

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

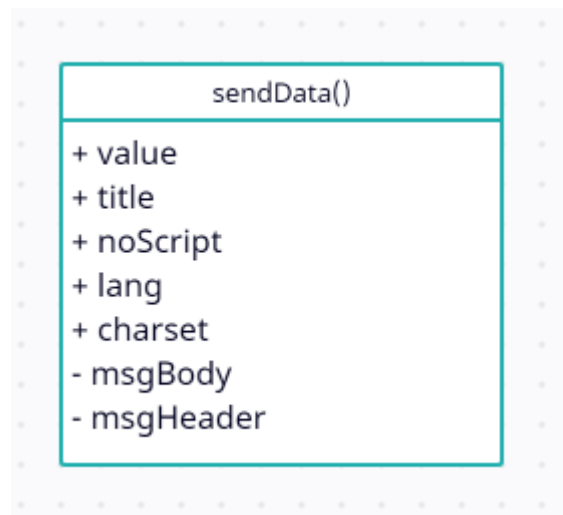


9. kép Adatbázis felépítése kapcsolattal

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 jelentkezni. 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 felöltött 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.



10. kép Készülő 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`, `noScript`, `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 begyűjti. A `title`, `noScript`, `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 készülő 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 áll, amik vagy a felhasználónév, jelszó vagy pluszban az e-mail címet tárolja, és ez megy a szerverre, a regisztráció vagy bejelentkezés funkcionál.

3.3 Részletes feladatspecifikáció, algoritmusok

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

Register() függvény:

```
const [usernameReg, setUsernameReg] = useState("");
const [passwordReg, setPasswordReg] = useState("");
const [password2Reg, setPassword2Reg] = useState("");
const [emailReg, setEmailReg] = useState("");

const register = () => {
  if (passwordReg === password2Reg) {
    Axios.post("http://localhost:3001/register", {
      username: usernameReg,
      password: passwordReg,
      email: emailReg,
    }).then((response) => {
      console.log(response);
    });
  } else {
    alert("passwordDoesntMatch");
    console.log("passwordDoesntMatch");
  }
};
```

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 error-t 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.

useEffect() függvény:

```
useEffect(() => {
  Axios.get("http://localhost:3001/login").then((response) => {
    if (response.data.loggedIn === true) {
      setLoginStatus(response.data.user[0].userName);
    }
  });
}, []);
```

12. kép useEffect függvény

A useEffect függvény, ami a ReactJS része, egy függvény, ami lefut, ha valami történik az oldalon. Ez a példány 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.

sendData() függvény:

```
const sendData = (e) => {
  e.preventDefault();

  function send() {
    Axios.post("http://localhost:3001/getData", {
      header: msgHeader,
      code: msgBody,
    }).then((response) => {
      if (response.data.message === "undefined") {
        send();
      } else {
        console.log(response.data.message);
      }
    });
  }

  if (title === "" || title === "undefined" || title === null) {
    alert("Empty field, check fields again.");
  } else {
    if (noScript === "" || noScript === "undefined" || noScript === null) {
      alert("Empty field, check fields again.");
    } else {
      if (lang === "" || lang === "undefined" || lang === null) {
        alert("Empty field, check fields again.");
      } else {
        if (charset === "" || charset === "undefined" || charset === null) {
          alert("Empty field, check fields again.");
        } else {
          if (typeof (title || noScript || lang || charset) !== "undefined") {
            setMsgHeader(lang + "," + charset + "," + title + "," + noScript);
            setMsg(value);
            if (msgHeader !== "" && typeof msgHeader !== "undefined" && msgHeader !== null) {
              send();
            } else {
              alert("Error in handling data: undefined error: try again!");
            }
          } else {
            alert("Error in handling data: undefined error: try again!");
          }
        }
      }
    }
  }
};
```

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 és 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 `'undefined'` akkor futassa a függvényt újra, de ha nem `'undefined'` 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 `'undefined'`. Ha mindegyik érték megfelelt, akkor megnézi, hogy ezeknek az adatoknak nem `'undefined'` e típusuk, mert akkor használhatatlannak lennének a szerver számára. Ha jó volt, akkor fűz. Utána megnézi, hogy az összefűzött `msgHeader` változó nem üres vagy null, vagy `'undefined'` e. Ha az, akkor dobjon hibát, ha nem akkor futassa a `send()` függvényt.

3.4 Tesztelési dokumentáció

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

Regisztráció tesztelés: normális körülmények, készítő által.

Register

Register

Already have an account? [Login here!](#)

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.

□✎ Edit📄 Copy🗑 Delete8test1\$2b\$10\$Mxct.U6mu/vjV30qwADte4TxoMzl2fwrZTWYIZ2jns...test@test.com

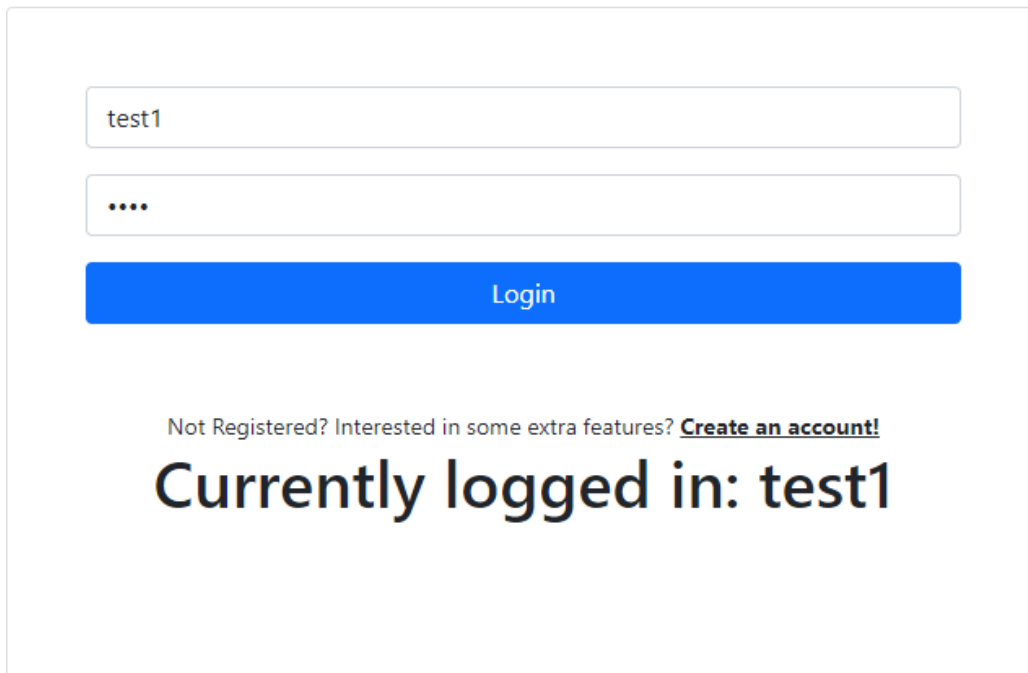
14. kép Reg-teszt

Regisztrációs teszt: hatalmas adatokkal, készítő által.

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.

Bejelentkezés teszt: normális körülmények, készítő által.



The screenshot shows a login interface. At the top, there are two input fields: the first contains the text 'test1' and the second contains four dots '....'. Below these fields is a blue button with the text 'Login'. Under the button, there is a line of text: 'Not Registered? Interested in some extra features? [Create an account!](#)'. Below this, the text 'Currently logged in: test1' is displayed in a large, bold font.

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 által.

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 által.

Head options

Website title	Website language
<input type="text" value="testTitle"/>	<input type="text" value="English"/>
Noscript text	Website character set
<input type="text" value="testNoscript"/>	<input type="text" value="UTF-8"/>

Body text editor

B *I*

TEST BODY

TEST BODY

'TEST BODY'

~~TEST BODY~~

TEST BODY

TEST BODY

TEST BODY

TEST BODY

TEST BODY **TEST BODY** *TEST BODY* ~~TEST BODY~~ -----

Currently logged in as:

[Convert and build](#)

TEST BODY
TEST BODY
 TEST BODY
 TEST BODY

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 által.

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.

4 Összefoglalás

4.1 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 szólni. 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.

4.2 Továbbfejlesztési lehetőségek

Dinamikus profil és link mutató:

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 készítesz. 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.

5 Felhasznált irodalom

<https://reactjs.org/docs/getting-started.html> - ReactJS dokumentáció (2021-09-16)

<https://reactrouterdotcom.fly.dev/docs/en/v6> - React Router dokumentáció (2021-09-20)

<https://axios-http.com/docs/intro> - Axios dokumentáció (2021-09-21)

<https://nodejs.org/en/docs/> - NodeJS dokumentáció (2021-10-01)

<https://getbootstrap.com/docs/5.1/getting-started/introduction/> - Bootstrap dokumentáció (2021-10-04)

<https://stackoverflow.com/> - Stack overflow kód megosztó weboldal (2021-09-10)

<https://uiwjs.github.io/react-md-editor/> - React MD editor by uiwjs dokumentáció (2022-01-05)

<https://www.npmjs.com/package/pastebin-ts> - Pastebin-Ts by j3lte dokumentáció (2022-01-10)

https://www.youtube.com/watch?v=W-sZo6Gtx_E – PedroTech - Register and Login Page Tutorial | ReactJs, NodeJS, MySQL (Beginner) videó (2022-01-15)

<https://www.youtube.com/watch?v=sTHWNPVNvm8> – Pedro Tech – Register and Login Tutorial | ReactJs, NodeJS, MySQL | Cookies, Sessions, Hashing videó (2022-03-18)

<https://nodejs.org/api/fs.html> - NodeJS FileStream dokumentáció (2022-03-20)

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

(2022-03-20)

<https://www.w3schools.com/js/default.asp> - Javascript segítség (2022-01-10)

<https://www.w3schools.com/mysql/default.asp> - MySQL segítség (2021-12-20)

<https://www.w3schools.com/react/default.asp> - React segítség (2021-09-10)

6 Ábrajegyzék

1. kép A XAMPP Control Panel megfelelő beállításai	6
2. kép GitHub oldal a letöltő linkkel.....	7
3. kép Sikeres modul telepítés	7
4. kép Főoldal.....	8
5. kép MD editor	9
6. kép FAQ Oldal	10
7. kép Login oldal	11
8. kép Register oldal	11
9. kép Adatbázis felépítése kapcsolattal	13
10. kép Készülendő weblap adatszerkezete	14
11. kép Register függvény	15
12. kép useEffect függvény.....	16
13. kép sendData függvény.....	16
14. kép Reg-teszt.....	17
15. kép Login-teszt.....	18
16. kép Weblap-teszt.....	19