

**Váci SZC Petőfi Sándor Műszaki Technikum,
Gimnázium és Kollégium**



SZOFTVERFEJLESZTŐ ÉS TESZTELŐ

Záródolgozat

Cím: LKSPORTS

Készítette: Hatházi Levente

Konzulens: Mizere Robin Roland

2025

Nyilatkozat

Alulírott tanuló kijelentem, hogy a szakdolgozat teljes mértékben a saját munkám eredménye. A felhasznált eszközöket és szakirodalmakat (tanulmány, internetes forrás, személyes közlés stb.) idézőjel és megjelölt források nélkül nem építettem be, egyértelműen megjelöltem. Az elkészült záródolgozatban található eredményeket a képzőintézmény és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja, a titkosításra vonatkozó esetleges megkötések mellett.

Aszód, 2024.

KONZULTÁCIÓS LAP

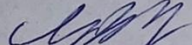
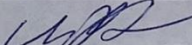
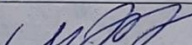
Szoftverfejlesztő
SZJ 50 613 1203A hallgató neve: Hatházi Levente Osztálya: 13/D

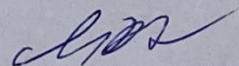
A záródolgozat témája:

.....

.....

.....

Sorszám:	A konzultáció időpontja:	A konzulens aláírása:
1.	2025. 03. 13.	
2.	2025. 03. 20.	
3.	2025. 03. 27.	

A záródolgozat beadható:
Aszód, 2025. 03. 31
.....
/ konzulens/

Tartalomjegyzék

1. Bevezető	6
2.0 Felhasználói dokumentáció	7
2.1 Kategorizált termékek	7
2.2 Szűrési és keresési lehetőségek	7
2.3 Termékinformációk és vásárlói vélemények	7
2.4 Kosár és vásárlási folyamat	7
2.5 Felhasználói fiók és történet	8
3.0 Hardver követelmények:	9
3.1 Minimális konfiguráció:	9
3.2 Egyéb szoftverek:	10
4.0 Telepítési útmutató	11
4.1 Node.js	11
4.2 XAMPP	15
4.3 Google Chrome	18
5.0 A program használatának a részletes leírása	19
5.1 Főoldal	19
5.2 Kosár	20
5.3 Bejelentkezés/Regisztráció	21
5.4 Termékoldalak	22
5.5 Felhasználási feltételek	23
5.6 Adatvédelmi Nyilatkozat	24
5.7 Profil	25
5.8 Admin	25
6.0 Fejlesztői dokumentáció	26
6.1 Témaválasztás indoklása	26
6.2 Az alkalmazott fejlesztői eszközök	26
6.3 Adatmodell leírása	28
6.4 Részletes feladat-specifikáció, algoritmusok	30
6.5 Forráskód	36

6.6 Tesztelési dokumentáció	39
6.7 Továbbfejlesztési lehetőségek	42
7.0 Összegzés	43
7.1 Források.....	43
8. Ábrajegyzék:.....	44

1. Bevezető

Szakdolgozatomat Csonka Kevinnel és Keresztes Krisztiánnal készíttem. Ennek neve pedig az LKSPORTS, ami egy Decathlon/Sportsdirects-re hajazó weboldal, legalábbis az volt az ihlet. Weboldalunkat HTML-ben fogjuk készíteni, a dizájnelemek pedig CSS-sel fognak készülni, az oldal “gerince”, ami az olyan folyamatokat fogja futtatni, amiket a felhasználó nem feltétlen lát pedig a JavaScript lesz. Azért választottuk ezt a témát mert mindhárman sportolunk/sportoltunk és fontos részét képezi az életünknek a sport, illetve a játéktervezés, amin még gondolkodtunk, viszont a weboldalfejlesztés mindhármanunknak jobban fekszik és jobban tetszik is. Munkánkhoz szükséges lesz egy adatbázis is, ahol tudjuk tárolni felhasználók információit (felhasználó név, email stb...), illetve az oldalon árult termékek (ruhák, cipők, sportkellékek stb..) is itt lesznek tárolva, ennek az alapja MYSQL lesz, amit a JavaScript és az ehhez letölthető package-k (express, mysql, cors stb..). Mi már felosztottuk egymás között a feladatokat, mivel 3 fő programozási nyelvünk van, próbáltunk mindenkinek az erősségeire ráfókuszálni, így Krisztián fogja a HTML részt, Kevin a CSS-t és jómagam pedig a JavaScriptet viszont ez nem azt jelenti, hogy el vagyunk szigetelve a másik programozási nyelvtől, ha bárkinek van bármilyen ötlete az simán belenyúlhat a programba és csinálhatja azt, amit kigondolt. Szakdolgozat közbeni kommunikációra Discord-ot fogunk használni, programunk megosztásához pedig GitHub-ot.

2.0 Felhasználói dokumentáció

A LKSPORTS célja, hogy gyors és kényelmes vásárlási élményt biztosítson a felhasználók számára, miközben minden igényt kielégít a sportfelszerelések és kiegészítők online vásárlása terén. A webáruház számos hasznos funkcióval rendelkezik, amelyek elősegítik a vásárlást, a termékek kezelését és a felhasználói élményt.

2.1 Kategorizált termékek

Az LKSPORTS különböző sportágakra lebontott kategóriákba rendezi a termékeket, mint például futás, kerékpározás, labdajátékok és egyéb sportok. E kategóriák gyors navigálást biztosítanak, így a felhasználók könnyen megtalálhatják az általuk keresett termékeket, anélkül, hogy időt veszítenének.

2.2 Szűrési és keresési lehetőségek

A vásárlás gyorsítására különböző szűrési funkciók állnak rendelkezésre, mint például árkategória, márka, méret, szín, és egyéb fontos paraméterek. Emellett a program rendelkezik egy fejlett keresőmotorral, amely lehetővé teszi, hogy a felhasználók kulcsszavak alapján kereshetnek a webáruházban, így könnyen rátalálnak a kívánt termékekre.

2.3 Termékinformációk és vásárlói vélemények

Minden termék részletes információval rendelkezik, beleértve a specifikációkat, használati útmutatókat és a termékekhez kapcsolódó képeket. Továbbá, a felhasználók olvashatják más vásárlók véleményeit és értékeléseit is, amelyek segítenek a vásárlás előtt a döntés meghozatalában.

2.4 Kosár és vásárlási folyamat

A felhasználók könnyen hozzáadhatják a termékeket a kosarukhoz, majd egyszerűen folytathatják a vásárlást. A rendelés leadásához egy gyors és biztonságos checkout folyamat áll rendelkezésre, amely tartalmazza a szállítási és fizetési lehetőségeket is. A vásárlók akár vendégként, akár regisztrált felhasználóként is vásárolhatnak.

2.5 Felhasználói fiók és történet

A regisztrált felhasználók számára elérhető egy egyedi fiók, amely lehetővé teszi a vásárlási előzmények megtekintését, a kedvenc termékek mentését, valamint az értesítések és akciók kezelését. A fiók egyszerűsíti a jövőbeli vásárlásokat és biztosítja, hogy a felhasználók mindig naprakészen követhessék rendeléseiket.

3.0 Hardver követelmények:

3.1 Minimális konfiguráció:

Processzor: 1.8 GHz-es vagy gyorsabb, kétmagos processzor

RAM: 4 GB

Tárhely: Legalább 10 GB szabad hely a rendszer számára

Grafikus kártya: 128 MB videomemória (amely képes megjeleníteni a weboldalon használt grafikai elemeket)

Képernyőfelbontás: 1366x768 vagy magasabb

Internet kapcsolat: Legalább 1 Mbps sebesség

Ajánlott konfiguráció:

Processzor: 2.5 GHz-es vagy gyorsabb, négymagos processzor

RAM: 8 GB vagy több

Tárhely: Legalább 20 GB szabad hely a rendszer számára

Grafikus kártya: 512 MB vagy nagyobb videomemória

Képernyőfelbontás: 1920x1080 vagy magasabb

Internet kapcsolat: Legalább 10 Mbps sebesség

Szoftver követelmények

A weboldalunk a következő operációs rendszereken működik:

Windows 10/11 (64-bites)

Mac OS X 10.15 vagy újabb

Linux (Ubuntu 20.04 vagy újabb)

A weboldalhoz szükséges további szoftver komponensek:

Böngészők:

Google Chrome (legújabb stabil verzió)

Mozilla Firefox (legújabb stabil verzió)

Microsoft Edge (legújabb stabil verzió)

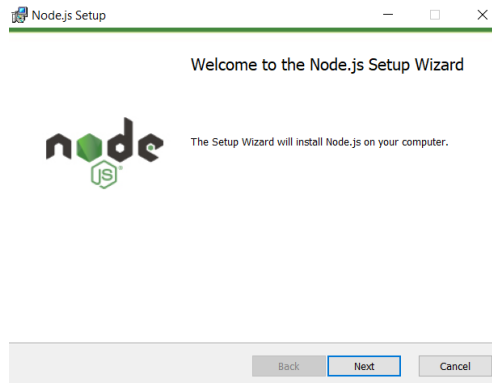
Safari (macOS esetén)

3.2 Egyéb szoftverek:

JavaScript és CSS: A weboldal dinamikus működéséhez szükséges. A böngészőknek támogatniuk kell a JavaScript futtatást és a modern CSS szabványokat. Adatbázis szerver: MySQL, ha az oldalon adatbázis-kezelés szükséges (pl. termékek, felhasználói fiókok tárolása) A weboldalhoz nem szükséges telepíteni külön programokat a felhasználói gépekre, mivel a böngészők elegendőek a weboldal teljes funkcionalitásának eléréséhez.

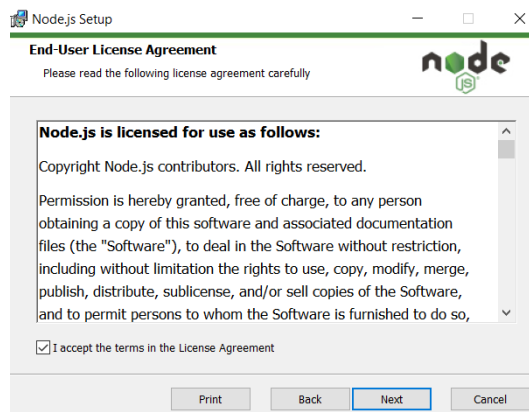
4.0 Telepítési útmutató

4.1 Node.js



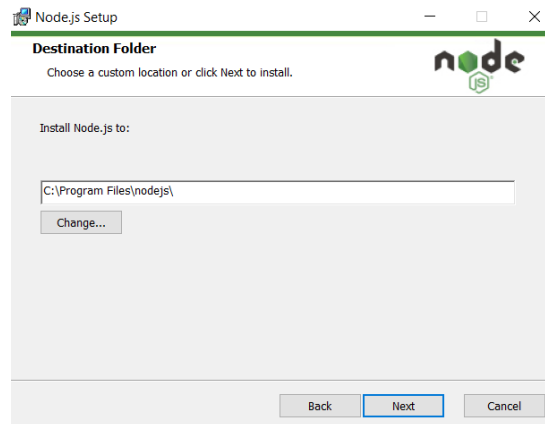
1. ábra: Node.js

Töltsd le a telepítőt a node.js weboldaláról(<https://nodejs.org/en>), majd futtasd le a letöltött .exe fájlt. Amikor a telepítőt elindítottad a „Tovább”(Next) gombra kattintva tudod elindítani a telepítést.



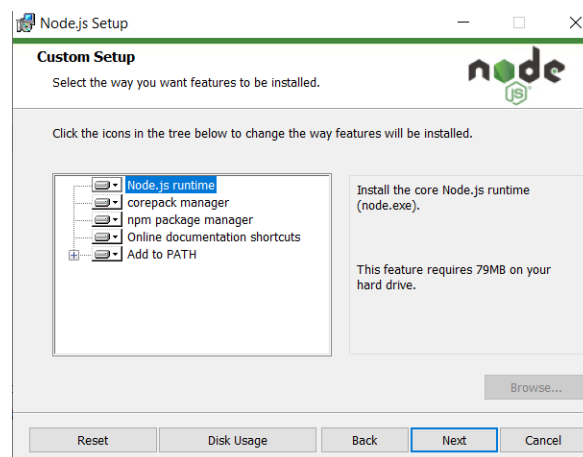
2. ábra: Node.js

Amikor a program átvisz a következő oldalra el kell fogadnod a licencszerződést ahhoz, hogy tudd folytatni a telepítési folyamatot. Majd pedig a „Tovább” (Next) gombra kattintva tudsz továbblépni.



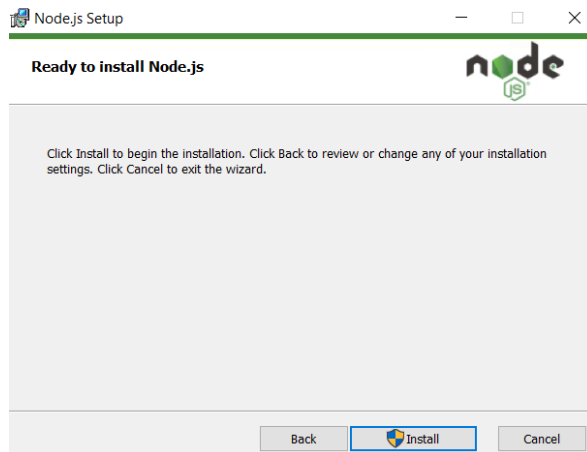
3. ábra: Node.js

Amikor az alkalmazás átvizsgálja a következő oldalt, akkor ki kell választanod a telepítési útvonalat, majd a „Tovább” (Next) gombra kattintva tovább lehet lépni.

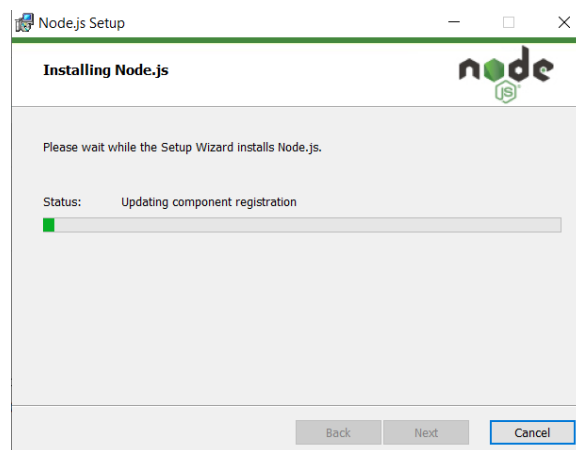


4. ábra: Node.js

Ebben az ablakban lehet kiválasztani a szükséges fájlokat, majd a „Tovább” (Next) gombra kattintva lehet tovább lépni.



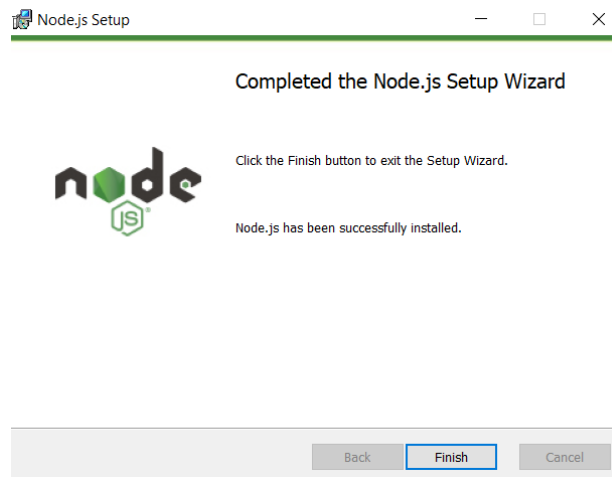
5. ábra: Node.js



6. ábra: Node.js

Miután a program átvitt a következő oldalra a telepítés végére ért. Itt az „Install” gombra kattintva be tudja fejezni a telepítési folyamatot.

A telepítő elkezd feltelepíteni a fájlokat majd, ha végzett akkor a „Tovább” (Next) gombbal lehet befejezni a telepítést.

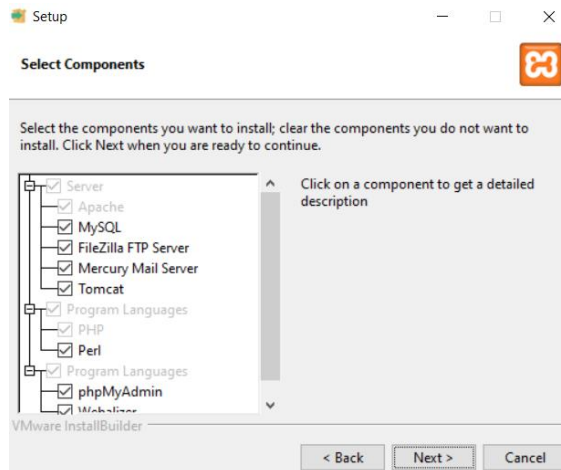


7. ábra: Node.js

Amikor átlép a következő oldalra a program a „Befejezés” gombra kattintva be lehet zárni a telepítőt és a Node.js használatra készen áll.

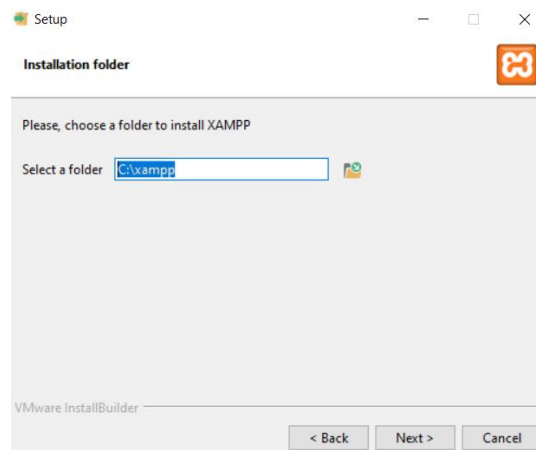
4.2 XAMPP

Első lépés, hogy a telepítőt az <https://www.apachefriends.org/hu/index.html> oldalról szükséges letölteni, majd a „Tovább”(Next) gomba kattintva tovább lépünk.



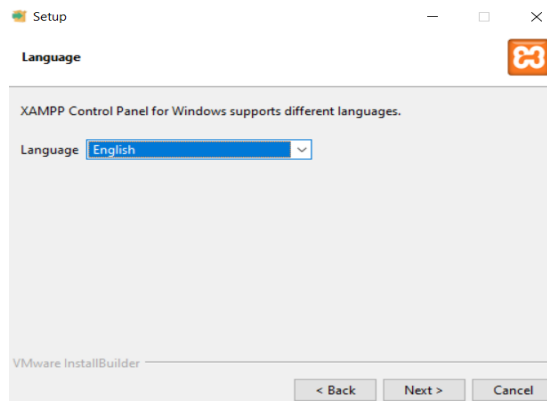
8. ábra: XAMPP

A telepítő ezen ablakjában ajánlott az alapbeállításokkal továbbhaladni amire, a „Tovább”(Next) gomb megnyomásával van lehetőség.



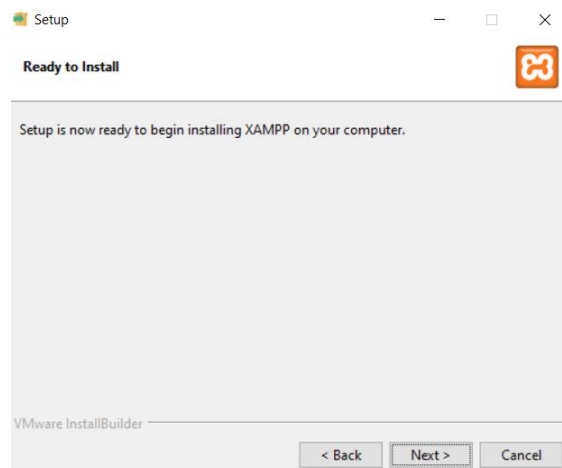
9. ábra: XAMPP

Ezen az ablakon van lehetőség kiválasztani a telepítési útvonalat, ahova a programot telepíteni kívánjuk, majd pedig a „Tovább” gombbal tudunk átlépni a következő oldalra.



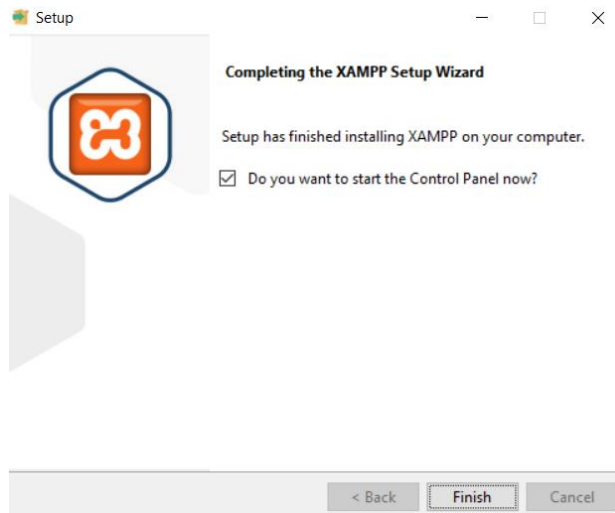
10. ábra: XAMPP

A telepítési szakasz ezen részében van lehetőség kiválasztani a kívánt nyelvet. Majd a „Tovább” gomb megnyomásával lehet tovább haladni.

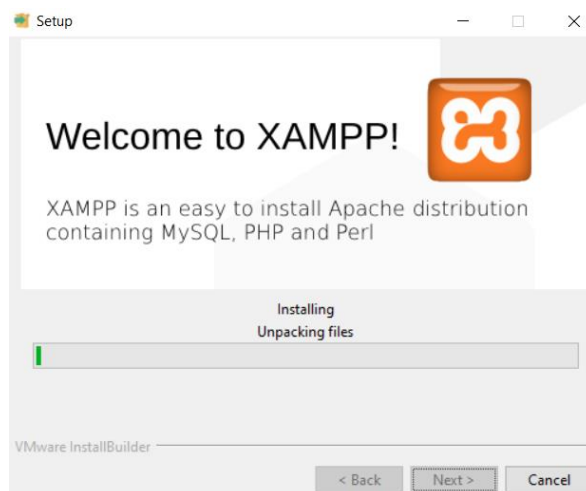


11. ábra: XAMPP

A XAMPP készen áll a telepítésre, a „Tovább” gombbal tudjuk elkezdni a telepítést.



12. ábra: XAMPP

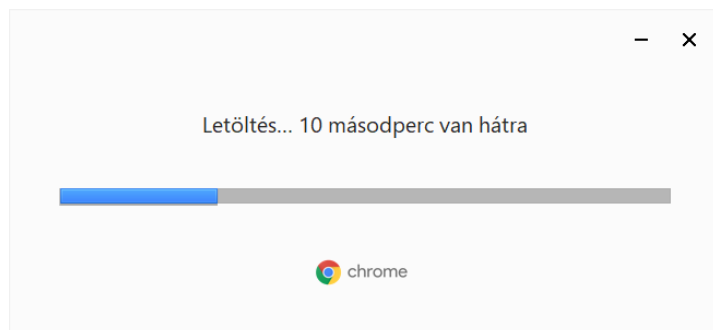


13. ábra: XAMPP

Majd a „Finish” gomb megnyomásával ki tudunk lépni a telepítőből ahol van lehetőség az azonnali indításra is.

4.3 Google Chrome

A böngészőt a https://www.google.com/intl/hu_hu/chrome/ oldalról lehet letölteni, majd a telepítőt szükséges futtatni.



14. ábra: Google Chrome

A telepítő automatikusan telepíti a fájlokat.

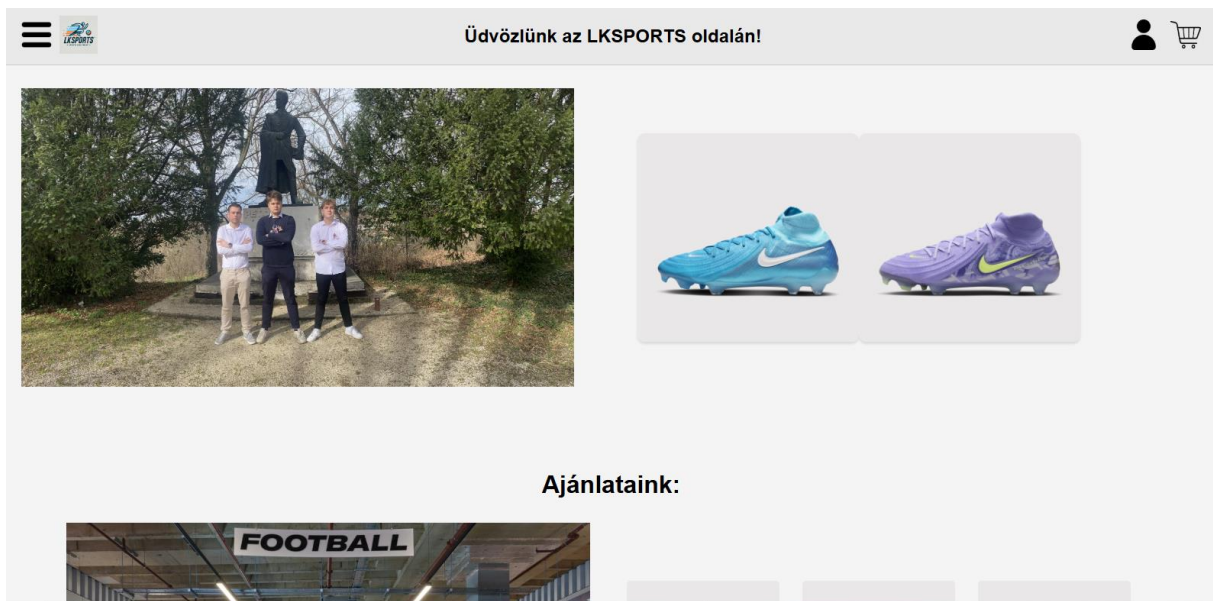


15. ábra: Google Chrome

A telepítés befejezése után lehetőség van bejelentkezni, illetve testreszabni a böngészőt.

5.0 A program használatának a részletes leírása

5.1 Főoldal





16.ábra: Index.html

Az oldalra fellépve egy kép található rólunk, emellett pedig egy slideshow ahol 3 cipőt lehet látni, lejjebb lapozva kiemelt ajánlatainkkal lehet találkozni, az oldal tetején lehet találni a navbart, ahol balról jobbra található egy menü gomb amire ráhúzva az egeret a 3 termékoldal közül lehet választani, emellett a főoldal gombját lehet találni amire rákattintva minden oldalról visszahoz a főoldalra középen egy felirat található, ami a felhasználó username-jét írja ki, jobb oldalt pedig a profil gomb, amire ráhúzva a bejelentkezés oldalra tud átlépni a gombra kattintva pedig a profil oldalra tud átlépni, legszélső gomb pedig a kosár gomb amivel a kosár oldalra van lehetőség átjutni.

5.2 Kosár

A főoldalon a „Kosár” gombra rányomva tud átlépni erre az oldalra, itt találhatóak a kosárba helyezett termékek, ezeket tudja törölni egyesével vagy akár az egészet is. Ezalatt a felhasználói adatok találhatóak. Legalján pedig a „Rendelés leadása” gomb, amivel tudjuk véglegesíteni a rendelést de csak akkor ha minden adat ki van töltve.

Kosár tartalma

	Nike Mercurial Superfly 10 Club FG stoplis focicipő, fekete - 23990 Ft - 42 - fekete	Törölés
	Asics Gel-Nimbus 27 férfi futócipő - 35000 Ft - 43 - zöld	Törölés

Összesen: 58990 Ft

Kosár ürítése

Név
Hatházi Levente

Felhasználónév
admin

Cím
2117, Isaszeg Bacsó Béla utca 4

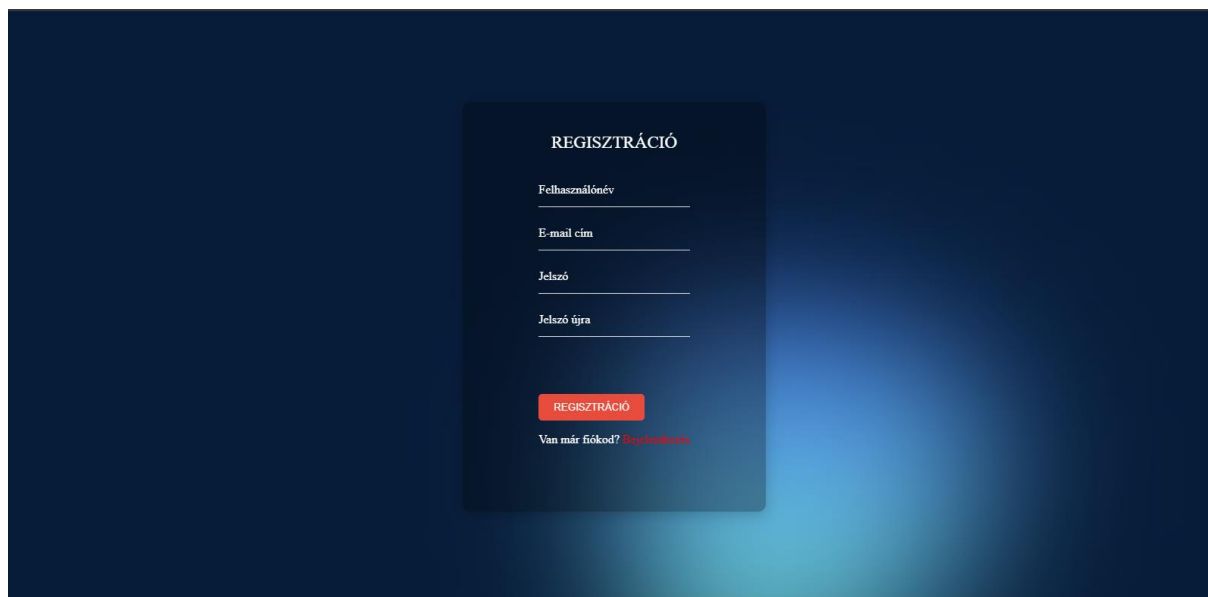
Telefonszám
06303065255

Email
hathazilevi465@gmail.com

Rendelés leadása

17. ábra: Kosár.html

5.3 Bejelentkezés/Regisztráció



REGISZTRÁCIÓ

Felhasználónév

E-mail cím

Jelszó

Jelszó újra

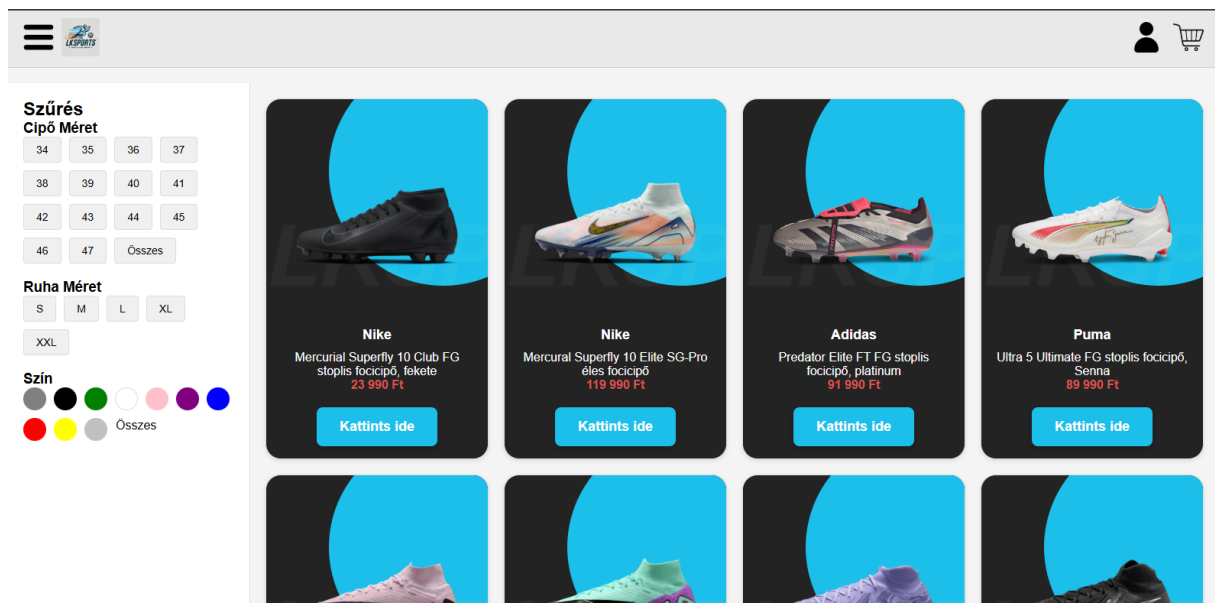
REGISZTRÁCIÓ

Van már fiókod? [Bejelentkezés](#)

18. ábra: Register.html

A regisztrációs és a bejelentkezési oldal nagyon hasonló, alapvetően a főoldalról a bejelentkezési oldalra tud átlépni és hogyha nincs regisztrált fiókja akkor tud a bejelentkezés oldalra átlépni a regisztrációs oldalra, ahol a fent látható mezőket szükséges kitölteni a regisztráció befejezése előtt.

5.4 Termékoldalak



19.ábra: Foci.html

Különböző termék oldalaink vannak, labdarúgással, futással, illetve kosárlabdával foglalkozók. A termékeket tartalmazó oldalaink különböző funkcióval rendelkeznek pl. szűrés méret és szín alapján, minden termék adott színnel, illetve mérettel rendelkezik így könnyen le lehet szűkíteni az opciókat. A termékekre rányomva pedig van lehetőségünk behelyezni a kosárba.

5.5 Felhasználási feltételek

Felhasználási Feltételek

1. Bevezetés

Köszönjük, hogy ellátogatott weboldalunkra. A jelen Felhasználási Feltételek (a továbbiakban: "Feltételek") szabályozzák a weboldal használatát. Kérjük, figyelmesen olvassa el őket, mivel a weboldal használatával elfogadja ezeket a feltételeket.

2. Szolgáltatás leírása

Weboldalunk információs és vásárlási lehetőségeket nyújt a látogatóknak, beleértve a ruházati termékek és kiegészítők online megvásárlását. Az oldalon található tartalmak kizárólag tájékoztatási célokat szolgálnak.

3. Felhasználói fiók

A weboldal bizonyos funkcióinak használatához felhasználói fiók regisztrációja szükséges. A felhasználó vállalja, hogy:

- Pontos és valós adatokat ad meg a regisztráció során.
- Fiókja biztonságát megőrzi, és nem osztja meg másokkal.
- Azonnal értesíti a weboldal üzemeltetőjét, ha fiókja illetéktelen hozzáférés áldozata lesz.

4. Szellemi tulajdon

A weboldalon található összes tartalom, beleértve a szövegeket, képeket, grafikákat, logókat, védjegyeket és egyéb anyagokat, a tulajdonos vagy licencpartnerei tulajdonát képezik. Ezek másolása, terjesztése, vagy módosítása csak a tulajdonos előzetes írásos engedélyével lehetséges.

5. Felelősség korlátozása

A weboldalon elérhető tartalmak pontosságára törekszünk, azonban hibák előfordulhatnak. A weboldalt "ahogy van" alapon nyújtjuk, és nem vállalunk felelősséget a használatából eredő károkért.

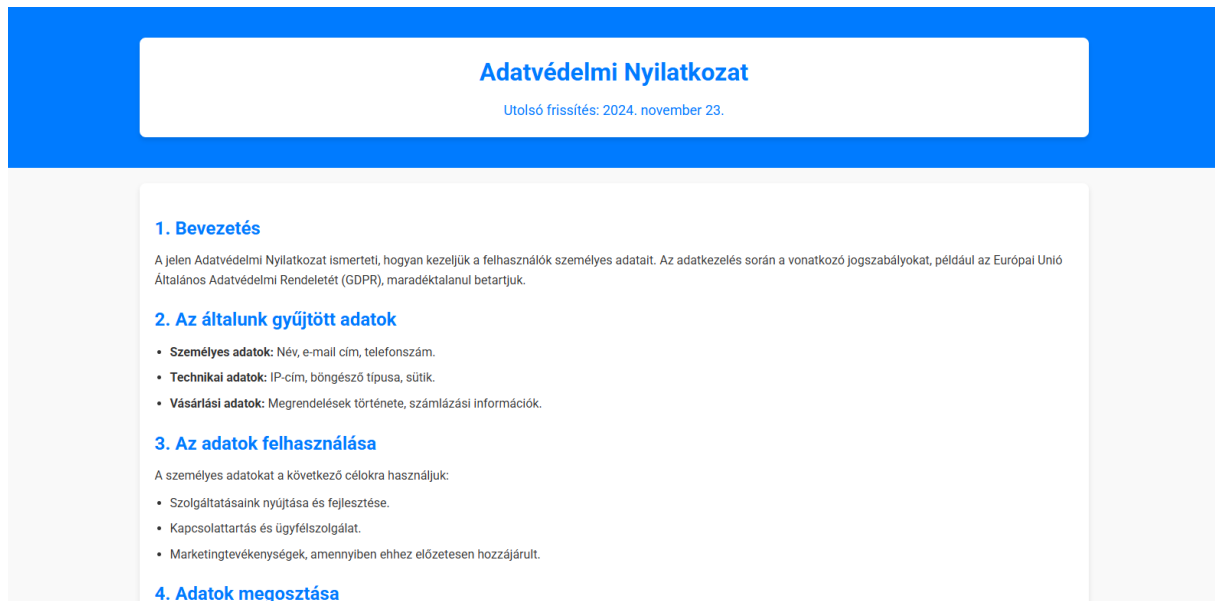
6. Vásárlási feltételek

20. ábra: felhasználási feltételek.html

Weboldalunk rendelkezik felhasználási feltételekkel, ennek több része is van.

1. Bevezetés
2. Szolgáltatás leírása
3. Felhasználói fiók
4. Szellemi tulajdon
5. Felelősség korlátozása
6. Vásárlási feltételek
7. Adatvédelem
8. Módosítások
9. Kapcsolat

5.6 Adatvédelmi Nyilatkozat



21.ábra:adatnyil.html

Weboldalunk rendelkezik adatvédelmi nyilatkozattal is hogy megvédjük a felhasználók személyes adatait.

1. Bevezetés

2. Az általunk gyűjtött adatok

3. Az adatok felhasználása

4. Adatok megosztása

5. Adatbiztonság

6. Sütik

7. Az ön jogai

8. Kapcsolat

5.7 Profil

Admin

Név: Levente

Felhasználónév: admin

E-mail: hathazilevi465@gmail.com

Lakcím: 2177, Aszód

Telefonszám: 06303065255

Módosítás

Eddigi rendeléseid:

Rendelés ára: 23990 Ft

Rendelés tartalma: Nike Mercurial Superfly 10 Club FG stoplis focipő, fekete

Kijelentkezés

22.ábra: profile.html

Profil oldalunkon a bejelentkezett felhasználók tudják az adataikat megtekinteni, megváltoztatni, illetve az eddig leadott rendeléseiket megtekinteni. Hogyha az adatbázisban a felhasználónak a role-ja 1-es akkor megjelenik a fent látható admin gomb amivel át tud lépni az admin felületre ahol minden felhasználónak tudja változtatni az adatait. Az oldal legalján található a „Kijelentkezés” gomb amire rányomva az oldal kijelentkeztetni a felhasználót a fiókjából.

5.8 Admin

Admin felület

Id Név Felhasználónév E-mail Lakcím Telefonszám

Role

Megváltoztatás

Kapcsolat

Email: lksports@gmail.com
Telefon: +36 1 234 5678
Cím: 2100 Godólló, Dózsa György út 1.

Kövess minket

Facebook
Instagram
Twitter

Információ

Adatvédelmi irányelvek
Felhasználási feltételek
GYIK

© 2024 LKSPORTS. Minden jog fenntartva.

23.ábra: admin.html

Az admin gombra rákattintva az „Admin felület” nevű oldal nyílik meg, itt tudja az admin változtatni minden felhasználó minden adatát(kivéve jelszó).

6.0 Fejlesztői dokumentáció

6.1 Témaválasztás indoklása

Az LKSPORTS egy sportáruház hasonló, mint például a Decathlon vagy a SportsDirect. Csatatunk azért ezt a témakört választotta mivel mindhárman nagy sportrajongók vagyunk és gyakran használjuk ezeket az oldalakat, viszont érzünk hiányosságokat használat közben így megálmodtuk az LKSPORTS-ot, ami egy olyan sportáruház aminek letisztult dizájnja, könnyen használható funkciói a nem túl tapasztalt felhasználóknak is értelmezhetőek és használhatóak.

6.2 Az alkalmazott fejlesztői eszközök

A program megírásához:

Programok:

-Visual Studio Code

Programozási nyelvek:

- HTML5

-CSS3

-JavaScript

-Node.js

-Bootstrap

-Illetve JavaScripthez szükséges modulok

Bővítmények:

express,

cors,

mysql2,

bcryptjs,

jsonwebtoken,

dotenv

Az adatbázis-szerver kapcsolathoz:

-Xampp

-Mysql

-JavaScript

Egymás közötti feladatmegosztási platform:

-GitHub

-Discord

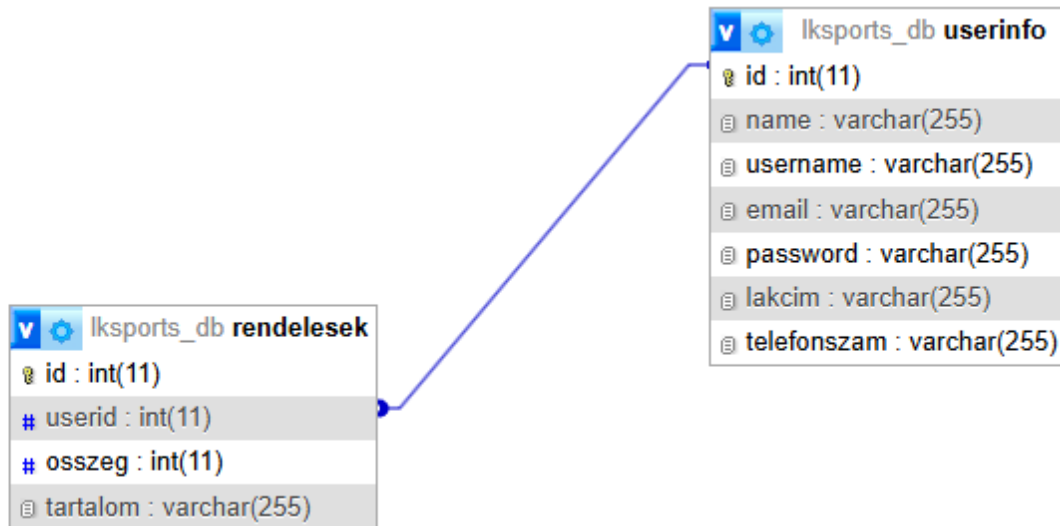
Dokumentációhoz:

-Microsoft Word

6.3 Adatmodell leírása

Adatbázisunk MySQL nyelven íródott és a programunkba JavaScript nyelven íródott a kapcsolat amivel összeköttöttük az adatbázist és az oldalunkkal.

Adatbázisunk neve: lksports_db



24.ábra: localhost: phpmyadmin

Adatbázisunk 2 táblát foglal magába.

1.tábla: userinfo

Ez a tábla tartalmazza a regisztrált felhasználók adatait:

<input type="checkbox"/>	Módosítás	Másolás	Törölés	2	Hatházi Levente	admin	hathazilevi465@gmail.com	\$2b\$10\$gs53fE0BHchRxdBaw8tY7.8fjucNmrZJofDCZD37f4K...	2117, Isaszeg Bacsó Béla utca 4	06303065255
--------------------------	-----------	---------	---------	---	-----------------	-------	--------------------------	--	---------------------------------	-------------

25.ábra: userinfo

-ID: Minden felhasználó egy külön ID-vel rendelkezik és ez alapján azonosítja be az oldal hogy éppen ki jelentkezik be.

-Name: A felhasználó teljes neve, regisztrációnál nem kötelező megadni viszont utána a profil oldalon szükséges beállítani, ahhoz hogy valaki rendelést tudjon leadni.

-Username: A felhasználó „beceneve” regisztrációnál szükséges megadni az adatot, felhasználó által kiválasztott név.




-Email: A felhasználó saját e-mail címe, ezt is szükséges megadni regisztrációnál.

-Password: A felhasználó jelszava, regisztrációnál szükséges megadni a felhasználó ezzel tud utána bejelentkezni az oldalra, az adatbázisba már titkosítva kerül fel.

-Lakcim: A felhasználó lakcíme, ezt nem szükséges megadni a regisztrációnál, viszont rendelés előtt szükséges hogy a felhasználó kitöltse a profil oldalon.

- Telefonszam: A felhasználó telefonszáma, ezt nem szükséges megadni a regisztrációnál, viszont rendelés előtt szükséges hogy a felhasználó kitöltse a profil oldalon.

2. tábla: Rendelesek

	id	userid	osszeg	tartalom
<input type="checkbox"/>  Módosítás  Másolás  Törölés	2	2	143980	Nike Mercurial Superfly 10 Club FG stoplis focicip...

26.ábra: rendelesek

ID: A rendelés száma, elsődleges kulcs.

Userid: A rendelést leadó USER id-je ez alapján tudja azonosítani a rendelőt az oldal

osszeg: A rendelés összege

Tartalom: Rendelés termékei

6.4 Részletes feladat-specifikáció, algoritmusok

Backend:

Adatbázis kapcsolat létrehozása:

```
const db = mysql.createPool({
  host: "localhost",
  user: "root",
  password: "",
  database: "lksports_db"
});
```

27. ábra: Server.js

Ezen a képen JS-ben kapcsolódik a programunk az lksports_db nevű adatbázishoz,

Felhasználó regisztrálása az oldalra:

```
app.post("/register", async (req, res) => {
  const { username, email, password } = req.body;

  try {
    const [users] = await db.query("SELECT * FROM userinfo WHERE username = ?", [username]);
    if (users.length > 0) {
      return res.status(400).json({ message: "Felhasználónév már használatban" });
    }

    const hashedPassword = await bcrypt.hash(password, 10);

    await db.query("INSERT INTO userinfo (username, email, password) VALUES (?, ?, ?)", [username, email, hashedPassword]);
    res.status(201).json({ message: "Felhasználó sikeresen regisztrálva" });
  } catch (err) {
    console.error("Hiba regisztráció során:", err);
    res.status(500).json({ message: "Hiba regisztráció során" });
  }
});
```

28. ábra: Server.js

Ez a függvény bekér minden ismert adatot ezzel a felhasználónévvel és megnézi hogy van-e már ilyen névvel felhasználó, majd pedig letitkosítja a jelszót és elküldi az adatbázisnak.

Oldalra való bejelentkezés:

```
app.post("/login", async (req, res) => {
  const { username, password } = req.body;

  try {

    const [users] = await db.query("SELECT * FROM userinfo WHERE username = ?", [username]);
    if (users.length === 0) {
      return res.status(400).json({ message: "Nem megfelelő felhasználónév vagy jelszó." });
    }

    const user = users[0];

    const isMatch = users.length > 0 && await bcrypt.compare(password, users[0].password);
    if (!isMatch) {
      return res.status(400).json({ message: "Nem megfelelő felhasználónév vagy jelszó." });
    }

    const secret = process.env.JWT_SECRET;
    if (!secret) {
      console.error("HIBA: JWT_SECRET nincs beállítva!");
      return res.status(500).json({ message: "Szerverhiba: JWT_SECRET hiányzik" });
    }
  }
}
```

29.ábra: Server.js

```
const token = jwt.sign({ id: user.id, username: user.username, password: user.password, lakcim: user.lakcim, telefonszam: user.telefonszam, email: user.email }, secret, { expiresIn: "1h" });
res.json({
  token,
  user: {
    id: user.id,
    username: user.username,
    email: user.email,
    password: user.password,
    lakcim: user.lakcim,
    telefonszam: user.telefonszam
  }
});

} catch (err) {
  console.error("Hiba a bejelentkezés során:", err);
  res.status(500).json({ message: "Hiba a bejelentkezés során" });
}
});
```

30.ábra: Server.js

Ellenőrzi hogy helyes-e a felhasználónév, majd pedig a jelszót is és ezután kiállít egy token a felhasználónak ami egy óráig érvényes, és tartalmazza a felhasználó adatait.

Profil megváltoztatása:

```
app.put("/profile", async (req, res) => {
  const authHeader = req.headers.authorization;
  if (!authHeader || !authHeader.startsWith("Bearer ")) {
    return res.status(401).json({ message: "Nincs hozzáférés" });
  }

  try {
    const token = authHeader.split(" ")[1];
    const decoded = jwt.verify(token, process.env.JWT_SECRET);

    const { username, email, telefonszam, lakcim } = req.body;

    const [users] = await db.query("SELECT * FROM userinfo WHERE id = ?", [decoded.id]);
    if (users.length === 0) {
      return res.status(400).json({ message: "Nincs ilyen felhasználó" });
    }

    const user = users[0];

    await db.query("UPDATE userinfo SET username = ?, email = ?, lakcim = ?, telefonszam = ? WHERE id = ?",
      [username || user.username, email || user.email, lakcim || user.lakcim, telefonszam || user.telefonszam, user.id]);

    res.json({ message: "Profil frissítve" });
  }
});
```

31.ábra: Server.js

Leellenőrzi hogy a felhasználó rendelkezik-e tokennel, majd pedig az oldalon látható felületen beírt adatokra megváltoztatja az adatokat.

Frontend:

Regisztráció:

```
document.querySelector('#registerForm').addEventListener('submit', async (e) => {
    e.preventDefault();

    const username = document.querySelector('#username').value;
    const email = document.querySelector('#email').value;
    const password = document.querySelector('#password').value;
    const password2 = document.querySelector('#confirm-password').value;

    if (password.length < 6) {
        alert("A jelszónak legalább 6 karakter hosszúnak kell lennie!");
        return;
    }

    if (password !== password2) {
        alert("Nem egyezik a két jelszó.");
        return;
    }
});
```

32.ábra:register.html

Ez a függvény ellenőrzi hogy a megadott jelszónak megfelelő-e a hossza ha pedig nem, akkor újra megkell adni. A függvény azt is ellenőrzi hogy a megadott jelszavak egyeznek-e.

```
const adatok = { username, email, password };

try {
    const response = await fetch('http://localhost:3000/register', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify(adatok)
    });

    const responseData = await response.json();

    if (!response.ok) {
        alert(responseData.message || 'Hiba történt a regisztráció során!');
        return;
    }

    alert(responseData.message);
    window.location.href = "login.html";
} catch (error) {
    console.error('Hiba történt:', error);
    alert('Hiba történt a regisztráció során!');
}

});
```

33.ábra: register.html

Bejelentkezés:

```
document.getElementById('loginForm').addEventListener('submit', function(event) {
    event.preventDefault();

    const username = document.getElementById('username').value;
    const password = document.getElementById('password').value;

    fetch('http://localhost:3000/login', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json'
        },
        body: JSON.stringify({
            username: username,
            password: password
        })
    })
    .then(response => response.json())
    .then(data => {
        if (data.token) {
            alert('Sikeres bejelentkezés!');
            localStorage.setItem('token', data.token);
            window.location.href = 'index.html';
        } else {
            alert(data.message || 'Hibás bejelentkezési adatok');
        }
    })
});
```

34.ábra: login.html

A regisztráció és a bejelentkezési függvény megegyezik nagyrésztében, a konstansok megkapják az oldalra beírt értékeket majd pedig ezeket továbbadják a backendnek amiket a backend felküld az adatbázisnak.

Profil változtatás:

```
document.addEventListener("DOMContentLoaded", () => {
  const token = localStorage.getItem("token");
  fetch("http://localhost:3000/profile", {
    method: "GET",
    headers: { "Authorization": `Bearer ${token}` }
  })
  .then(response => response.json())
  .then(data => {
    document.getElementById("nameDisplay").innerText = data.name || "Nincs információ";
    document.getElementById("usernameDisplay").innerText = data.username || "Nincs információ";
    document.getElementById("emailDisplay").innerText = data.email || "Nincs információ";
    document.getElementById("telefonszamDisplay").innerText = data.telefonszam || "Nincs információ";
    document.getElementById("lakcimDisplay").innerText = data.lakcim || "Nincs információ";
  })
  .catch(error => console.error("Hiba:", error));
});
```

35. ábra: profile.html

```
function saveChanges() {
  const name = document.getElementById("name").value;
  const username = document.getElementById("username").value;
  const email = document.getElementById("email").value;
  const telefonszam = document.getElementById("telefonszam").value;
  const lakcim = document.getElementById("lakcim").value;

  const updatedData = {};
  if (name.trim() !== "") updatedData.name = name;
  if (username.trim() !== "") updatedData.username = username;
  if (email.trim() !== "") updatedData.email = email;
  if (telefonszam.trim() !== "") updatedData.telefonszam = telefonszam;
  if (lakcim.trim() !== "") updatedData.lakcim = lakcim;

  console.log("Frissített adat:", updatedData);

  updateProfile(updatedData);
  closeModal();
}
```

36. ábra: profile.html

Lekéri az adatokat az adatbázisból, majd pedig ezeket az oldalon megváltoztatottra kicseréli.

6.5 Forráskód

Termék oldalon egy termékre kattintva tovább irányít ahol lehetőség van betenni a kosárba a terméket.

```
</script>

function atiranyitas(shoeName, shoePrice, shoeImage, shoeSize, shoeColor) {
  window.location.href = "details.html?name=${encodeURIComponent(shoeName)}&price=${encodeURIComponent(shoePrice)}&image=${encodeURIComponent(shoeImage)}&size=${encodeURIComponent(shoeSize)}&color=${encodeURIComponent(shoeColor)}";
}
```

37.ábra: foci.html

```
<div class="product-list">
  <div class="product-item" data-size="42" data-color="fekete">
    
    <h3>Nike</h3>
    <p>Mercurial Superfly 10 Club FG stoplis focicipő, fekete</p>
    <p class="price">23 990 Ft</p>
    <button class="button" onclick="atiranyitas('Nike Mercurial Superfly 10 Club FG stoplis focicipő, fekete', '23990', '../../kepek/superfly1.jpg', '42', 'fekete')>Kosárba</button>
  </div>
</div>
```

38.ábra: details.html

```
</script>

document.addEventListener("DOMContentLoaded", () => {
  const params = new URLSearchParams(window.location.search);
  const name = params.get("name");
  const price = params.get("price");
  const image = params.get("image");
  const size = params.get("size");
  const color = params.get("color");
  // ...
});
```

39.ábra: details.html

```
</nav>
<div class="container-fluid">
  <div class="row">
    <div class="col-md-3 col-md-6">
      <div id="shoe-details">
        <h2 id="shoe-name"></h2>
        <img id="shoe-image" src="" alt="Cipő képe" width="300" height="300">
        <p id="shoe-price"></p>
        <p id="shoe-size"></p>
        <p id="shoe-color"></p>
        <button id="kosarba">Kosárba</button>
        <a href="foci.html">Vissza a termékekhez</a>
      </div>
    </div>
  </div>
</div>
```

40.ábra: details.html

Kosárba gombra kattintva behelyezi a terméket a kosárba.

```
document.getElementById("kosarba").addEventListener("click", function() {
const name = document.getElementById("shoe-name").textContent;
const price = document.getElementById("shoe-price").textContent.replace("Ár: ", "").replace(" Ft", "");
let image = document.getElementById("shoe-image").getAttribute("src");
const size = document.getElementById("shoe-size").textContent.replace("Méret: ", "");
const color = document.getElementById("shoe-color").textContent.replace("Szín: ", "");

if (!image.startsWith("http")) {
    image = "kepek/" + image.split("/").pop();
}

let cart = JSON.parse(localStorage.getItem("cart")) || [];
cart.push({ name, price: parseInt(price), image, size, color });

localStorage.setItem("cart", JSON.stringify(cart));

alert("A termék sikeresen hozzáadva a kosárhoz!");
});
} else {
    document.getElementById("shoe-details").innerHTML = "<p>Nincs elérhető adat.</p>";
}
});
```

41.ábra: details.html

Majd pedig ezt megjeleníti akosar.html-ben, és le tudja adni a rendelést.

```
document.getElementById('checkout-form').addEventListener('submit', function(event) {
  event.preventDefault();

  const cart = JSON.parse(localStorage.getItem("cart")) || [];
  const total = cart.reduce((sum, item) => sum + item.price, 0);

  const items = cart.map(item => item.name);

  fetch("http://localhost:3000/profile", {
    method: "GET",
    headers: { "Authorization": `Bearer ${token}` }
  })
  .then(response => response.json())
  .then(data => {
    const userId = data.id;
    fetch("http://localhost:3000/saveOrder", {
      method: "PUT",
      headers: {
        "Content-Type": "application/json"
      },
      body: JSON.stringify({
        userid: userId,
        osszeg: total,
        tartalom: items.join(", ")
      })
    })
    .then(response => response.json())
    .then(data => {
      $('#successModal').modal('show');
      localStorage.removeItem("cart");
      loadOrderCart();
    })
  })
});
```

42.ábra:kosar.html

```

function loadOrderCart() {
  const cart = JSON.parse(localStorage.getItem('cart')) || [];
  const cartItemsList = document.getElementById('order-cart-items');
  const cartTotal = document.getElementById('order-cart-total');
  cartItemsList.innerHTML = '';
  let total = 0;

  cart.forEach((item, index) => {
    const listItem = document.createElement('li');
    listItem.className = 'list-group-item d-flex justify-content-between align-items-center';
    listItem.innerHTML = `
      
      ${item.name} - ${item.price} Ft - ${item.size} - ${item.color}
      <button class="btn btn-danger btn-sm" onclick="removeFromCart(${index})">Törlés</button>
    `;
    cartItemsList.appendChild(listItem);
    total += item.price;
  });

  cartTotal.textContent = `${total} Ft`;
}

```

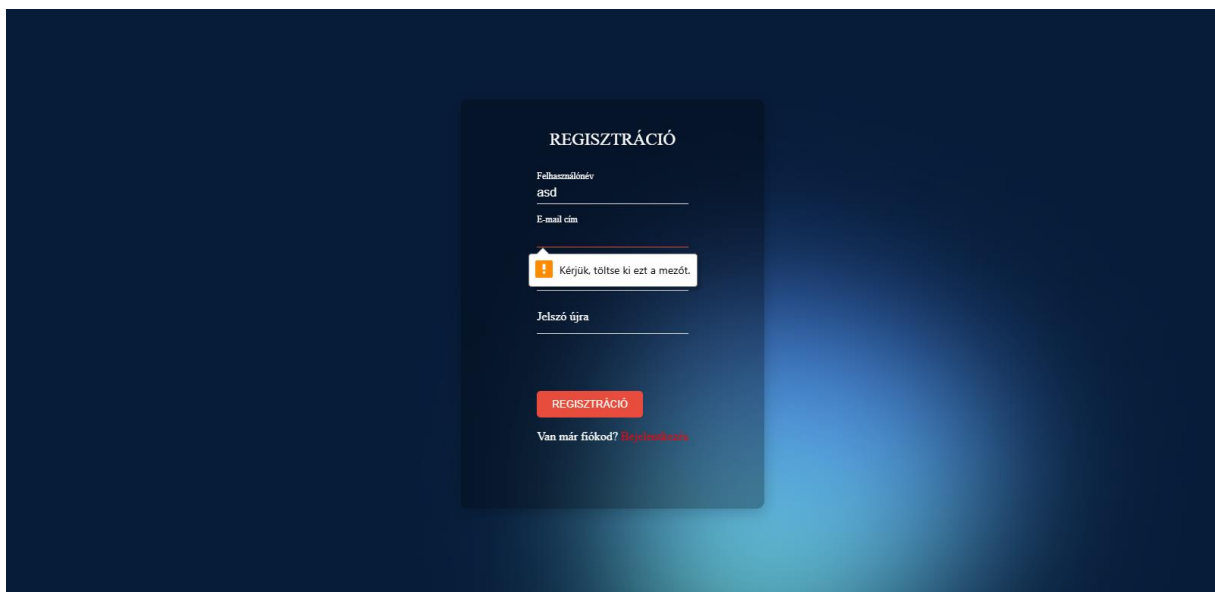
43.ábra: kosar.html

6.6 Tesztelési dokumentáció

1. teszt eset:

Hibás regisztráció: 1/1 Nincs kitöltve az oldal:

Ha a felhasználó regisztrációnál nem tölti ki a szükséges mezőket az oldal ezt jelzi neki.

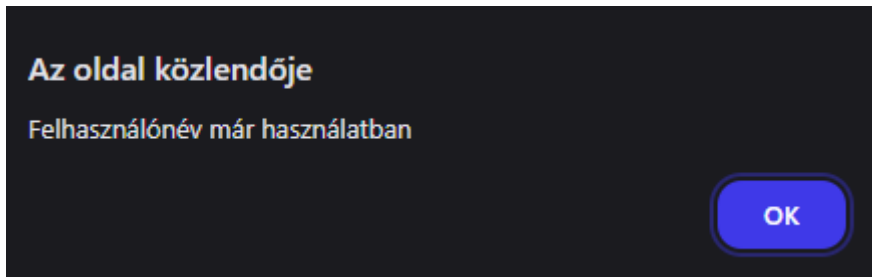


44.ábra: register.html

1/2

Már foglalt felhasználónév megadása:

Amennyiben a felhasználó már egy használt felhasználónevet akar regisztrálni az oldal ezt nem engedi.

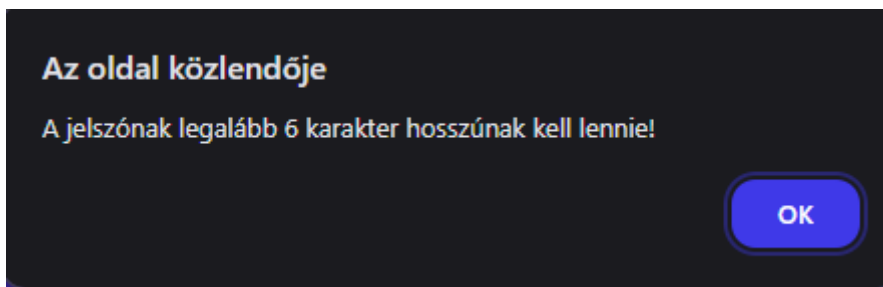


45.ábra: register.html

1/3

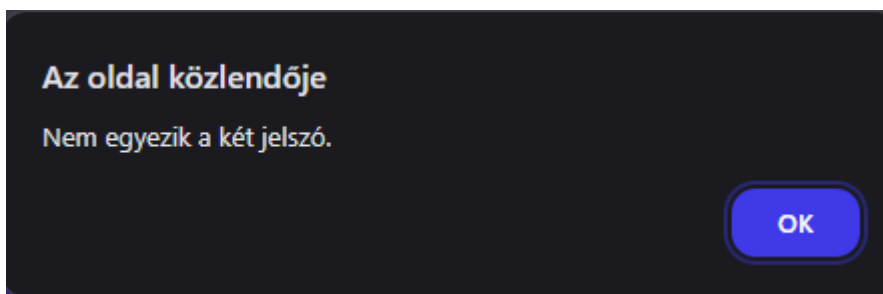
Nem egyező jelszavak/Nem elég hosszú jelszavak:

Ha a felhasználó 6 karakternél rövidebb jelszót ad meg a regisztráció sikertelen lesz, és jelzi az oldal.



46.ábra: register.html

Ha a felhasználó által megadott két jelszó nem egyezik azt is kijelzi az oldal



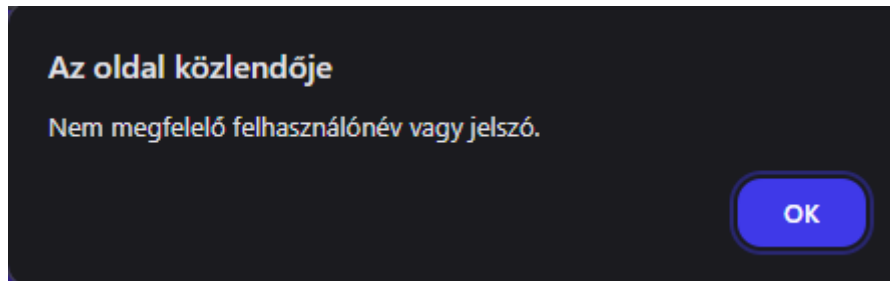
47.ábra: register.html

Bejelentkezés:

2/1

Hibás adatok megadása:

Ha a felhasználó nem megfelelő felhasználónevet vagy jelszót ad meg az oldal nem engedi hogy bejelentkezzen:



48.ábra: login.html

Profil:

3/1

Bejelentkezés nélküli rendelés:

Ha a felhasználó bejelentkezés nélkül szeretne rendelést leadni akkor hibaüzenetet kap a kosár oldalra érve:

Kosár tartalma

Összesen: 0 Ft

Kosár ürítése

Jelentkezz be ahhoz hogy le tudd adni a rendelésed!

Név

Felhasználónév

Cím

Telefonszám

Email

Rendelés leadása

49.ábra: kosar.html

6.7 Továbbfejlesztési lehetőségek

A termékeket nem tárolja az adatbázis, esetleges bővítés esetén ez megkönnyítheti a dolgokat.

Jelenleg az oldal 3 fajta sportot és ezekhez kapcsolódó termékeket tartalmaz, ezeket a későbbiekben lehet bővíteni.

7.0 Összegzés

Szakdolgozatunk elkészítése segített, megtanultam jobban csapatban dolgozni, feladatok elosztását, egymás segítségét és az ezzel járó plusz feladatokat, illetve a csapatban dolgozás egyfajta kényelmét, Krisztiánnal és Kevinnel sikeresen segítettük egymást fejlődni illetve teljesen pozitívan építő jellegű kritikákkal segítettük egymást. Bővítettem tudásom a HTML, JS és a CSS világában, illetve összefogó képet kaptam arról, hogy milyen egy „nagyobb” projekten dolgozni huzamosabb ideig. Megismertem különböző JavaScript bővítményeket, amik segítettek az oldal működésében. Nehézségeink voltak mivel egyikünk se dolgozott még ehhez hasonló projektben így együtt kellett betanulunk hogy mit és hogyan kell megosztani egymással, elkészíteni stb...

7.1 Források

Decathlon: <https://www.decathlon.hu/>

SportsDirect: <https://www.sportsdirect.hu/>

Molnár Máté Norbert órai anyagai: <https://tinyurl.com/psgmmate>

<https://freefrontend.com/>

<https://www.pngegg.com/>

<https://www.retailgazette.co.uk/blog/2024/11/sports-direct-milton-keynes/>

<https://www.w3schools.com/>

<https://medium.com/illumination/how-to-build-a-slideshow-using-html-css-and-javascript-977ecbdbf48c>

<https://focivilag.hu/hu>

<https://www.futobolt.hu/noi-futocipok>

8. Ábrajegyzék:

1. ábra: Node.js
2. ábra: Node.js
3. ábra: Node.js
4. ábra: Node.js
5. ábra: Node.js
6. ábra: Node.js
7. ábra: Node.js
8. ábra: Xampp
9. ábra: Xampp
10. ábra: Xampp
11. ábra: Xampp
12. ábra: Xampp
13. ábra: Xampp
14. ábra: Google Chrome
15. ábra: Google Chrome
16. ábra: Index.html
17. ábra: Kosar.html
18. ábra: Register.html
19. ábra: Foci.html
20. ábra: felhasználot.html
21. ábra: adatved.html
22. ábra: profile.html
23. ábra: admin.html
24. ábra: localhost:phpmyadmin
25. ábra: userinfo

26. ábra: rendelesek

27. ábra: server.js

28. ábra: server.js

29. ábra: server.js

30. ábra: server.js

31. ábra: server.js

32. ábra: register.html

33. ábra: register.html

34. ábra: login.html

35. ábra: profile.html

36. ábra: profile.html

37. ábra: foci.html

38. ábra: details.html

39. ábra: details.html

40. ábra: details.html

41. ábra: details.html

42. ábra: kosar.html

43. ábra: kosar.html

44. ábra: register.html

45. ábra: register.html

46. ábra: register.html

47. ábra: register.html

48. ábra: login.html

49. ábra: kosar.html