

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



SZOFTVERFEJLESZTŐ ÉS TESZTELŐ

Záródolgozat

Cím: LKSPORTS

Készítették: Csonka Kevin

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.

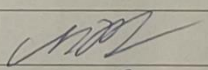
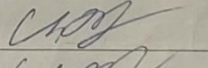
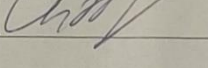
.....

KONZULTÁCIÓS LAP

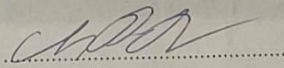
Szoftverfejlesztő
SZJ 50 613 1203

A hallgató neve: CSUNKA KEVIN Osztálya: 13.0

A záródolgozat témája: AZ LKSPOOTS EGY SPORT FELTÉRE-
LÉSEKET ADUSÍTÓ WEBÁRUHÁZ, MELY TÖBBEK
ÜSÖTT LABDASZÜGASHOZ, KOSÁRLABDAHOZ ÉS FUTÁSHOZ
RENKÉPES TERMÉKEKET FORGALMAZ.

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	24
5.6 Adatvédelmi Nyilatkozat.....	25
6.0 Fejlesztői dokumentáció	26
6.1 Témaválasztás indoklása	26
6.2 Az alkalmazott fejlesztői eszközök	27
6.3 Adatmodell leírása	29
6.4 Részletes feladat-specifikáció, algoritmusok	32
6.5 Forráskód	38
6.6 Tesztelési dokumentáció	42
6.7 Továbbfejlesztési lehetőségek	47
7.0 Összegzés	47

7.1 Források.....	49
8. Ábrajegyzék:.....	50

1. Bevezető

Jelenleg Levente és Krisztiánnal dolgozok, akikkel közösen dolgozunk a projektjeinken mindenkinek meg van adva, hogy ki min dolgozik. Én CSS azaz vizuális felület, Krisztián ö HTML, ami a szerkezetét és hierarchiáját készíti, Levente pedig a JavaScriptet, ami lehetővé teszi, hogy működjenek a különféle gombok és interaktív legyen az oldal. Hozzájárulnak a feladatokhoz és segítenek a különböző fejlesztési és tervezési folyamatokban.

Jelenleg egy sport weboldalon dolgozunk, ahol termékeket adunk el különböző méretben színekben stílusban. A célunk, hogy a vásárlók könnyen és gyorsan navigálhassanak a termékek között, miközben esztétikusan van bemutatva minden egyes termék, és az oldal felhasználóbarát módon van megtervezve.

A projekt során HTML, CSS, JavaScript használatával alakítjuk ki az oldal struktúráját és dizájnját. A weboldalhoz elengedhetetlen az adatbázis, amit pedig Xampp-al oldjuk meg.

A projekt fejlesztése során a GitHub-ot használjuk megosztási felületként. Itt tároljuk a kódokat, dokumentációkat és verziókat, így biztosítva a gördülékeny munkafolyamatot és a csapattagok közötti hatékony együttműködést. A GitHub lehetővé teszi, hogy könnyen nyomon kövessük a változásokat. A csapatmunka során minden tag különböző területeken járul hozzá a projekt sikeréhez, miközben a kommunikáció és a munka eredményeinek megosztása elengedhetetlen.

2.0 Felhasználói dokumentáció

Az LKSPORTS célja, hogy a vásárlóknak egy gyors, egyszerű és átlátható online vásárlási élményt nyújtson. A webáruházban többféle célszerű funkció található, amelyek megkönnyítik a vásárlást és a termékek kezelését. Ezekkel a funkciókkal szeretnénk javítani a felhasználói élményt

Az oldal kialakításakor fontos volt, hogy könnyen kezelhető legyen. Így mindenki könnyedén eligazodni tudjon rajta, függetlenül attól, hogy mennyire ismeri az online vásárlást.

2.1 Kategorizált termékek

Az LKSPORTS áruház szétválasztja a termékeit sportágak alapján. Például külön kategóriák vannak futás, kosár, labdarúgás és más különböző sportokhoz. Ez a megközelítés segíti a gyors és könnyű eligazodást.

A jól átlátható oldalnak köszönhetően az emberek gyorsan megtalálhatják az igényeikhez szükséges termékeket.

2.2 Szűrési és keresési lehetőségek

Az online vásárlást sokféle szűrő segíti, hogy könnyebb és gyorsabb legyen megtalálni, amit keresel. Például választhatsz árkatégoriák, márkák, méretek vagy színek alapján.

2.3 Termékinformációk és vásárlói vélemények

Minden terméknel részletes leírást találhatsz, beleértve a specifikációkat, használati útmutatókat, és hogy ne maradj le semmiről, nagy felbontású képeket is.

2.4 Kosár és vásárlási folyamat

A kiválasztott termékeket egy kattintással a kosárba teheted, hogy onnan folytasd a vásárlást. A rendelés véglegesítése során egyszerű, biztonságos és felhasználóbarát folyamat vár, amely magában foglalja a szállítási és fizetési lehetőségeket is.

Csak regisztrált fiókkal tudsz vásárolni. Emellett a rendeléseket is nyomon követheted, ami növeli az átláthatóságot és a vásárlói elégedettséget.

2.5 Felhasználói fiók és történet

Regisztrált felhasználóként személyre szabott fiókot használhatsz, ami sok kényelmi funkcióval segíti a vásárlást. A fiókodban könnyen ránézhetsz a korábbi vásárlásaidra, nyomon követheted a rendeléseid állapotát, és a kedvenc termékeidet elmentheted későbbre (így bármikor újra megtalálod őket).

A fiókkal korábbi szállítási és fizetési adatok automatikusan elérhetők. Így minden rendelésnél időt spórolsz meg.

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)

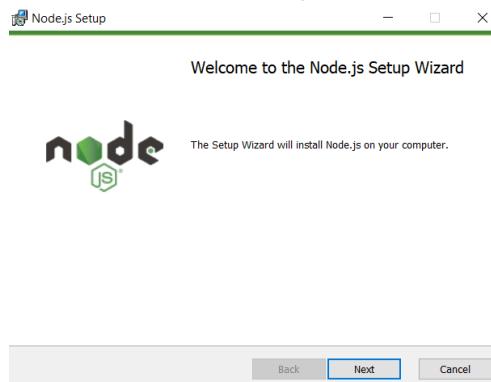
Safari (macOS esetén)

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.

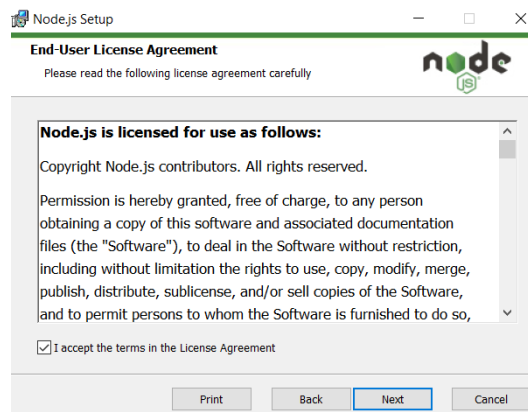
4.0 Telepítési útmutató

4.1 Node.js



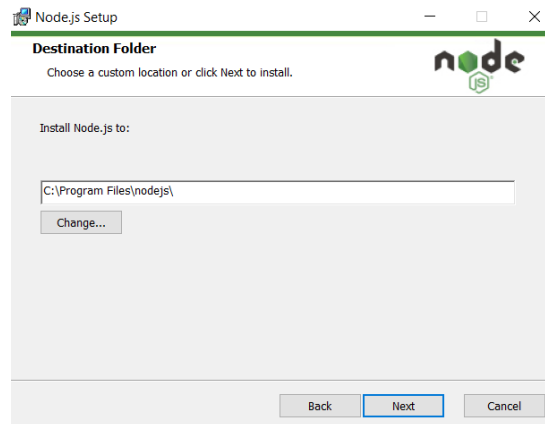
1. ábra: Node.js

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.



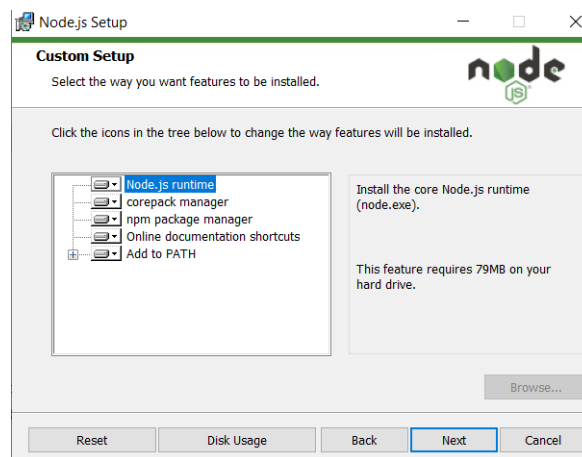
2. ábra: Node.js

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



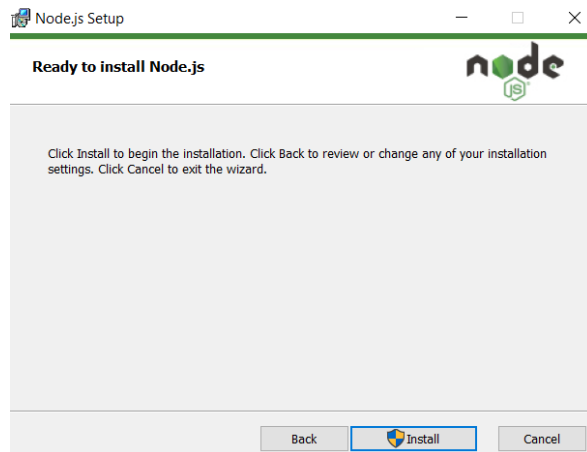
3. ábra: Node.js

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



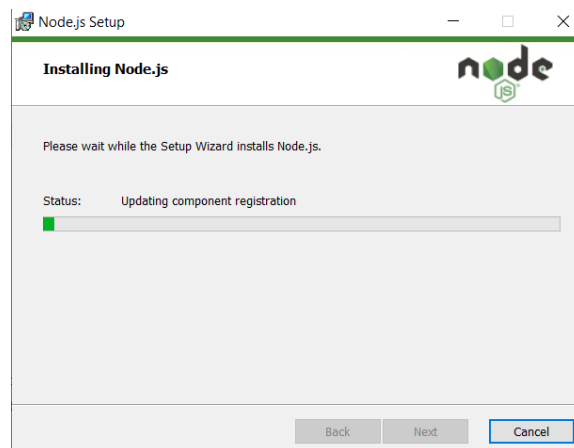
4. ábra: Node.js

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



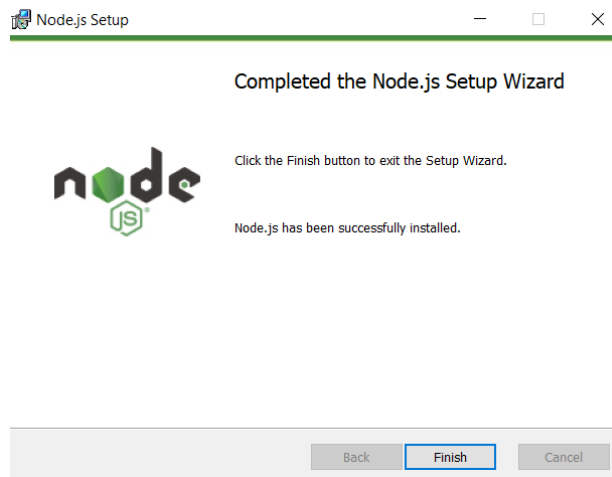
5. ábra: Node.js

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



6. ábra: Node.js

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

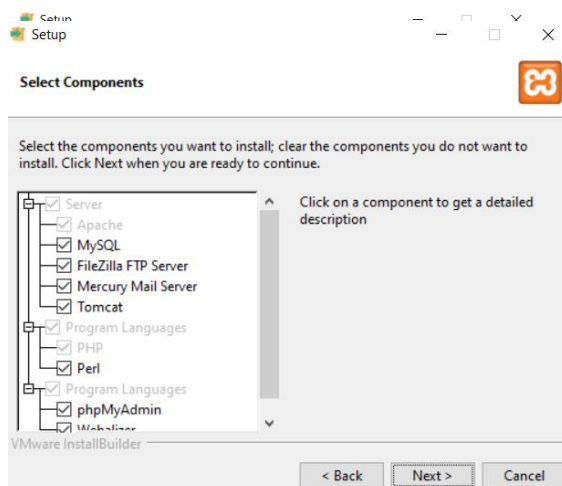


7. ábra: Node.js

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

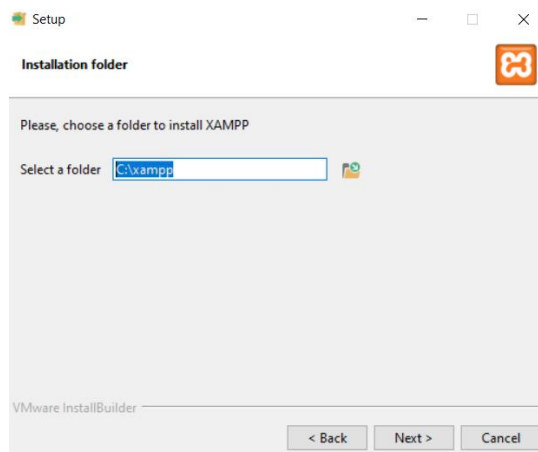
4.2 XAMPP

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



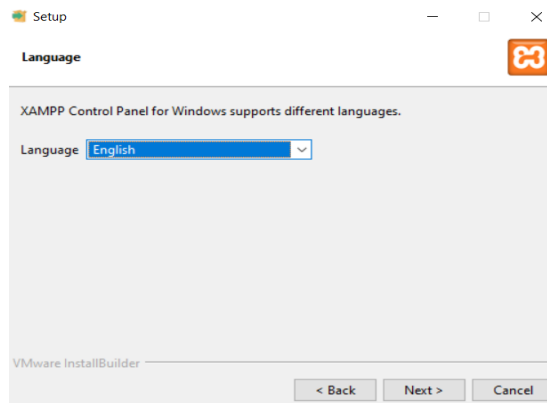
8. ábra: XAMPP

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



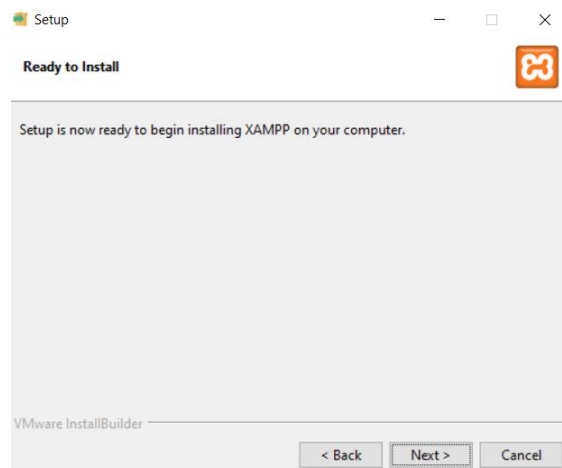
9. ábra: XAMPP

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



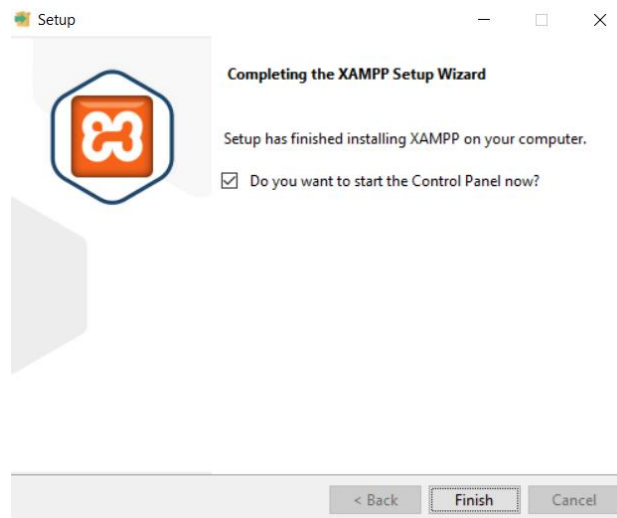
10. ábra: XAMPP

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



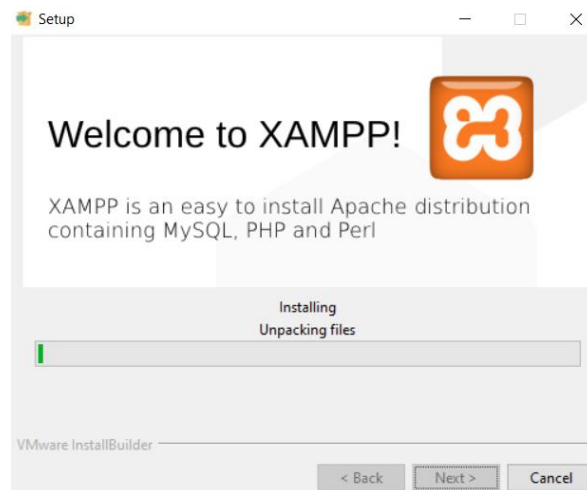
11. ábra: XAMPP

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



12. ábra: XAMPP

A XAMPP elkezd felmásolni az adatokat a gépre, majd a „Tovább” gombbal befejezzük a telepítést.

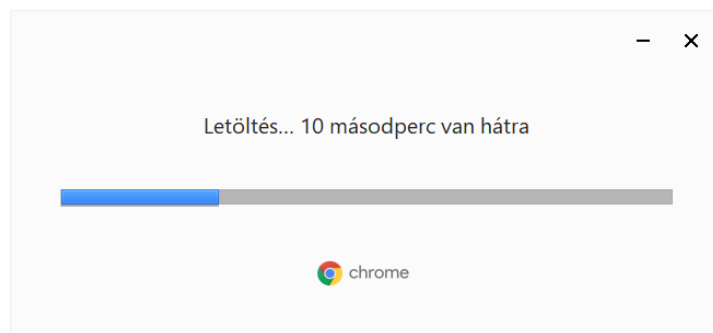


13. ábra: XAMPP

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.

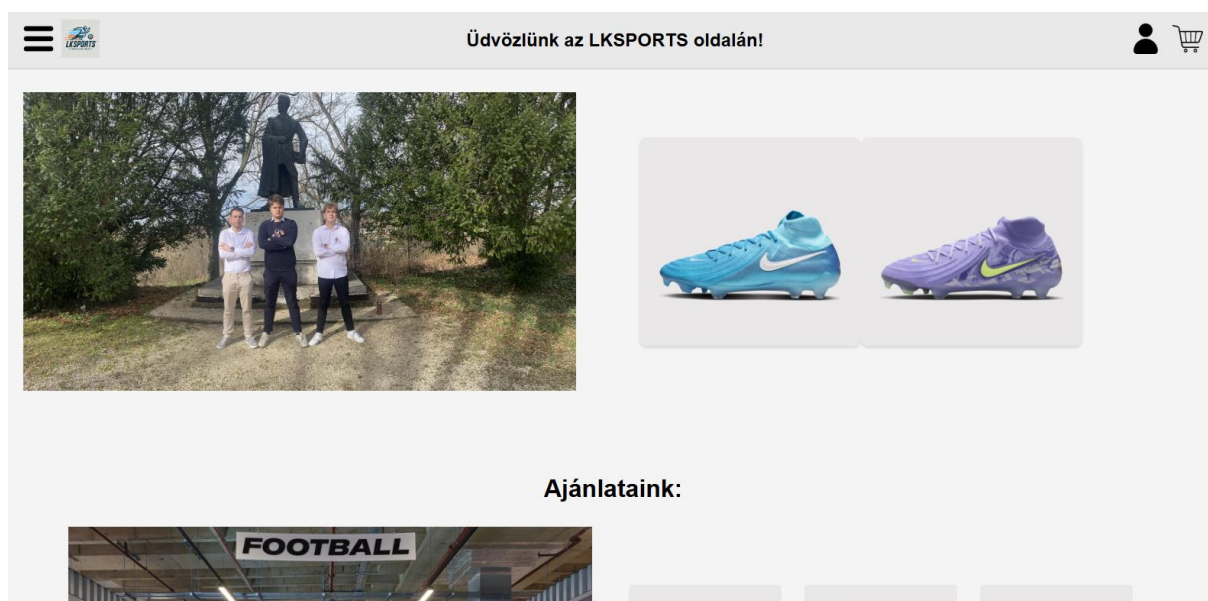


15. ábra: Google Chrome

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

5.0 A program használatának a részletes leírása

5.1 Főoldal



16.ábra: Index.html

Bejelentkezés után a felhasználók egy színes képes bemutatót látnak a főoldalon. Ez a látványos elem segít gyorsan észrevenni az aktuális ajánlatokat.

A bal felső sarokban egy "menü" ikon található. Erre kattintva elérhetők a különböző sportágakhoz kapcsolódó kategóriaoldalak. Ezzel a felhasználók gyorsan megtalálhatják az érdeklődési körüknek megfelelő termékeket.

A menü mellett található az LKSPORTS logó. Ezt bármikor megnyomva a felhasználó visszajut a főoldalra.

A jobb felső sarokban egy kosár ikon van. Erre kattintva megtekinthetők a kosárba helyezett termékek, és innen lehet elindítani a vásárlási folyamatot is.

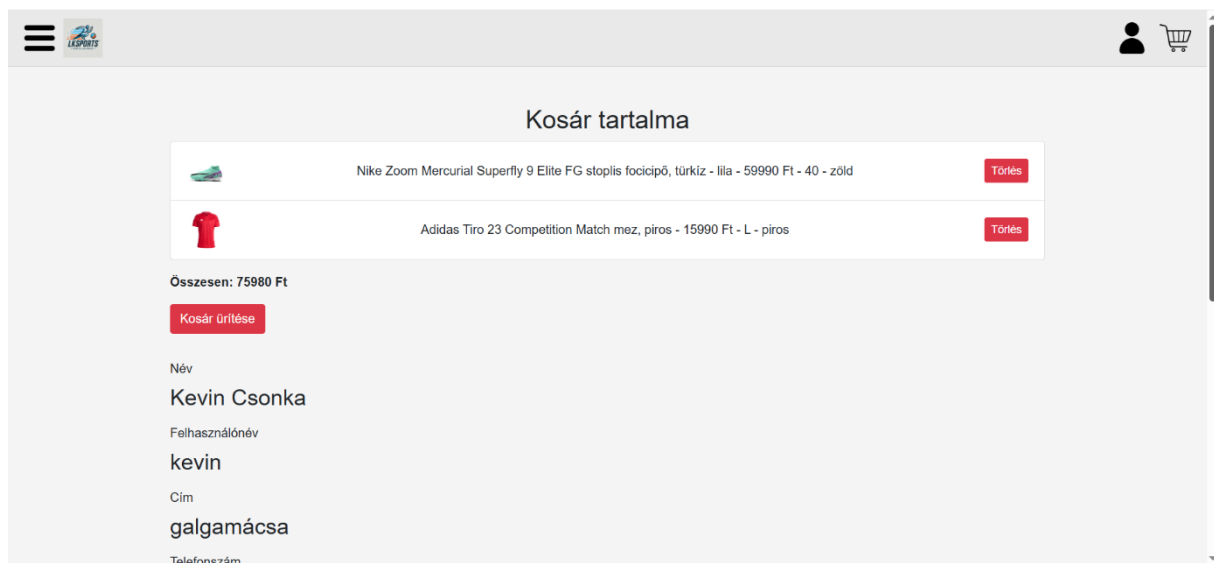
A kosár mellett van a profil ikonja. Ha egeret húzunk rá, megjelenik egy menü, ahol lehet bejelentkezni vagy regisztrálni, ha még nincs fiókunk. Ha rákattintunk, a profiloldalra jutunk, ahol módosíthatók a személyes adatok, megtekinthetők a rendeléseink.

5.2 Kosár

A főoldalon lévő „Kosár” ikonra kattintva a felhasználó a Kosár oldalra jut. Itt át tudja nézni a kiválasztott termékek adatait, változtatni a mennyiségüket, vagy akár el is távolíthatja őket.

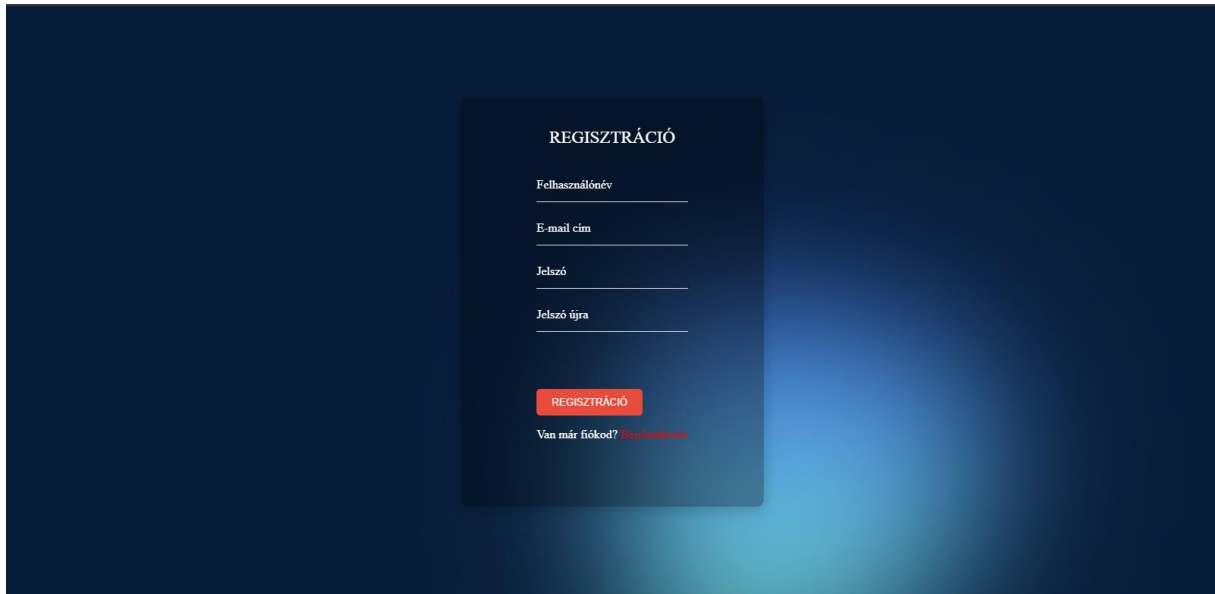
Ha a felhasználó meggyőződött arról, hogy minden szükséges termék a kosárban van, a „Rendelés” gombra kattintva elindíthatja a vásárlási folyamatot.

Ha a felhasználó még nem jelentkezett be, a rendszer felkínálja a bejelentkezési vagy regisztrációs lehetőséget.



17. ábra: Kosár.html

5.3 Bejelentkezés/Regisztráció

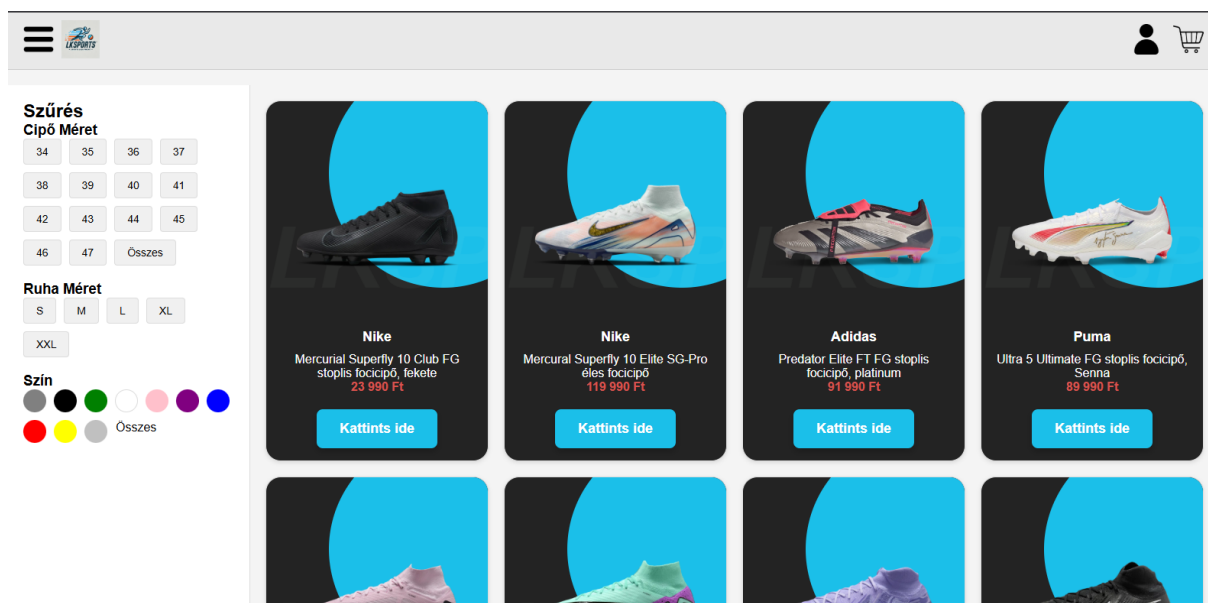
The image shows a registration form titled "REGISZTRÁCIÓ" centered on a dark blue background. The form is a light blue rectangle with the following fields: "Felhasználónév", "E-mail cím", "Jelszó", and "Jelszó újra", each with a corresponding input line. Below these fields is a red button labeled "REGISZTRÁCIÓ". At the bottom of the form, it says "Van már fiókod? [bejelentkezés](#)".

18. ábra: Register.html

A regisztrációs és a bejelentkezési oldalak kialakítása hasonló logikát követ. A felhasználóknak a megfelelő mezőkbe kell beírniuk a szükséges adatokat, például e-mail címet, jelszót, felhasználónév.

Az adatok megadása után a „Regisztráció/Bejelentkezés” gombra kattintva történik meg a regisztráció vagy a bejelentkezés. Sikeres regisztrációt követően a felhasználó azonnal használatba veheti személyes fiókját, míg bejelentkezés esetén a rendszer visszairányítja a főoldalra vagy az előzőleg meglátogatott oldalra.

5.4 Termékoldalak



19.ábra: Foci.html

Az LKSPORTS webáruház külön termékoldalt biztosít minden sport számára - mint például foci, futás és kosárlabda -, így a vásárlók gyorsan rábukkanhatnak az adott sportághoz tartozó kínálatra.

Az egyes termékoldalak különféle szűrőkkel rendelkeznek, ami segíti a pontos és gyors keresést. Méret és szín szerint is lehet szűrni, mert ezek az adatok minden termékhez pontosan meg vannak adva. Így a felhasználók könnyen leszűkíthetik a lehetőségeket a saját igényeiknek megfelelően.

Amikor a felhasználó rákattint egy adott termékre, a rendszer egy külön termékoldalra irányítja. Itt nagy képeket, részletes leírást és specifikációkat találhat a kiválasztott termékről. Ezen az oldalon további - ugyanolyan kategóriába tartozó - termékeket is megtekinthet, így könnyen találhat alternatívát vagy kiegészítőt a kiválasztott termékhez.



Nike
Mercurial
Superfly 10
Elite SG-
Pro éles
focicipő,
fehér

Ár: 119990 Ft

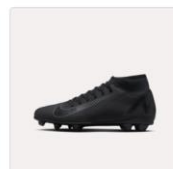
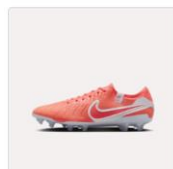
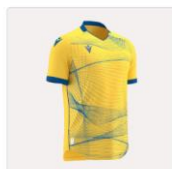
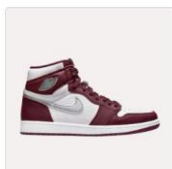
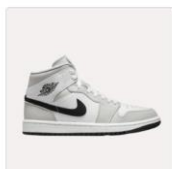
Méret: 41

Szín: fehér



További termékeink:

További termékeink:



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.

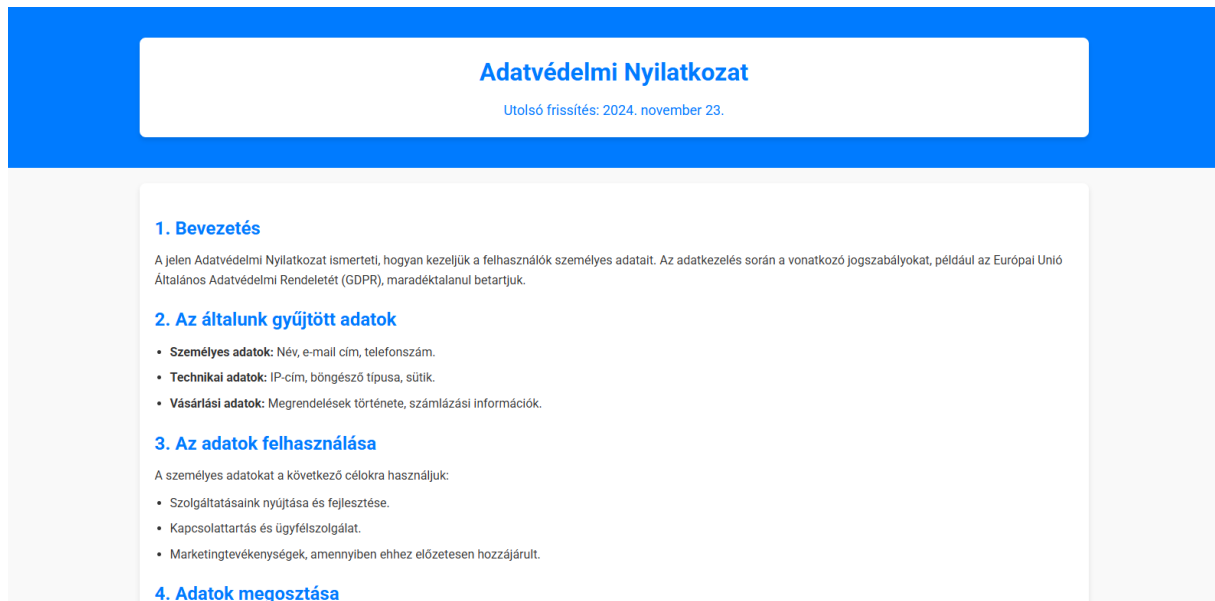
6. Vásárlási feltételek

20.ábra: felhaszfeltet.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

6.0 Fejlesztői dokumentáció

6.1 Témaválasztás indoklása

Az LKSPORTS egy olyan sportáruház, amely a már jól ismert Decathlonhoz vagy SportsDirecthez vagy egyéb nagyobb sport webáruházokhoz hasonlóan működik. Azért választottuk ezt a témát, mert mindhárman arra jutottunk, hogy mindenki szeret online vásárolni és a fiatalok körébe általános a sportolás és gyakran vásárolunk sportfelszereléseket online. De egy letisztult weboldalt szerettünk volna, ahol mindenki igényi alapján gyorsan megtalálja magának a sportfelszereléseket, sport termékeket.

Ezért született meg az LKSPORTS ötlete: egy modern, letisztult felületű és felhasználóbarát sportáruház, ahol minden funkció átlátható, könnyen elérhető, és még azok számára is egyszerű eligazodni, akik nem mozognak otthonosan az online vásárlás világában.

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

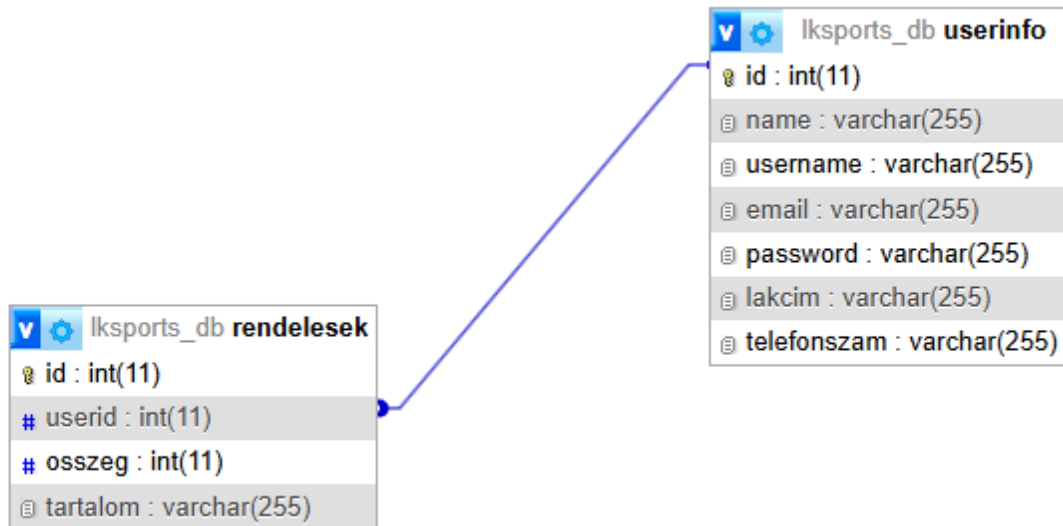
Dokumentációhoz:

- 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



22.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
--------------------------	-----------	---------	---------	---	-----------------	-------	--------------------------	--	---------------------------------	-------------

23.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.

-Lakcím: 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: Rendelések

		ID	userid	összeg	tartalom		
<input type="checkbox"/>	 Módosítás	 Másolás	 Törlés	2	2	143980	Nike Mercurial Superfly 10 Club FG stoplis focicip...

24.ábra: rendelések

ID: A rendelés száma, elsődleges kulcs.

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

összeg: A rendelés összege

Tartalom: Rendelés termékei

6.4 Részletes feladatspecifikáció, algoritmusok

Backend:

Adatbázis kapcsolat létrehozása:

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

25. 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" });
  }
});
```

25. ábra: Server.js

Ez a függvény bekér minden ismert adatot ezzel a felhasználónévvel és megnézi, hogy van-e már ilyen névvel felhasználó, majd pedig le titkosítja 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" });
    }
  }
}
```

27.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" });
}
});
```

28.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" });
  }
});
```

29.ábra: Server.js

Le ellenőrzi, 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;
  }
});
```

30.ábra: register.html

```

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!');
}
});

```

31.ábra: register.html

Le ellenőrzi hogy megfelelő-e a jelszó hossza, majd átadja a backendnek az adatokat

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');
        }
    })
});
```

32.ábra: login.html

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));
});
```

33.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();
}
```

34.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)}";
}
```

35.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>
```

36.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");
  // ...
});
```

37.ábra: details.html

```
</div>
<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>
```

38.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>";
}
});
```

39.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();
        })
    })
});
```

40.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`;
}

```

41.ábra: kosar.html

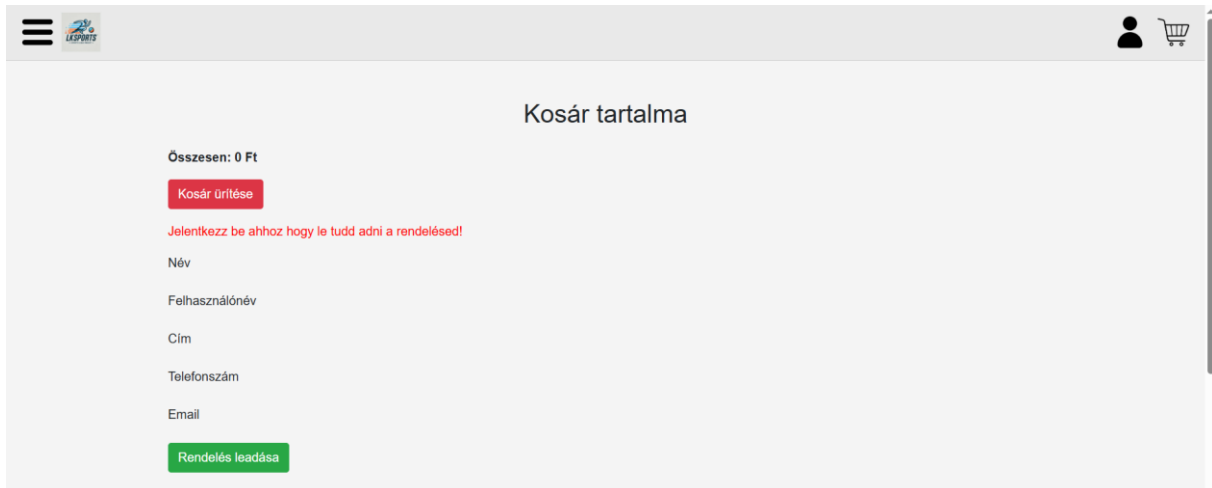
6.6 Tesztelési dokumentáció

1. Tesztetset:

Bejelentkezés nélküli rendelés

Hibás rendelés: 1/1 Nincs bejelentkezve a felhasználó:

Abba az esetben, ha a felhasználó nem jelentkezik be nem fogja tudni leadni a rendelést.



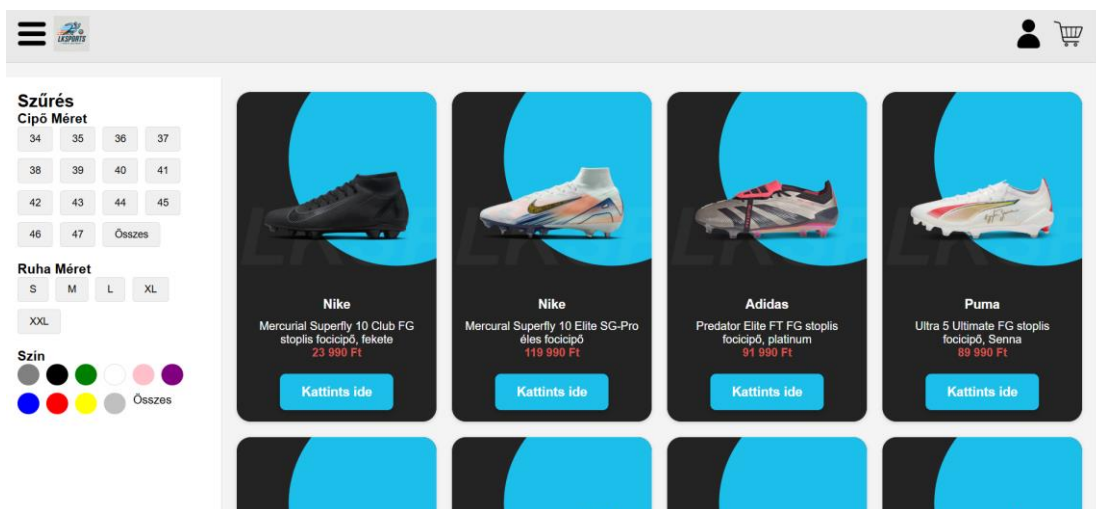
42.ábra: kosar.html

2. Tesztetset:

2/1

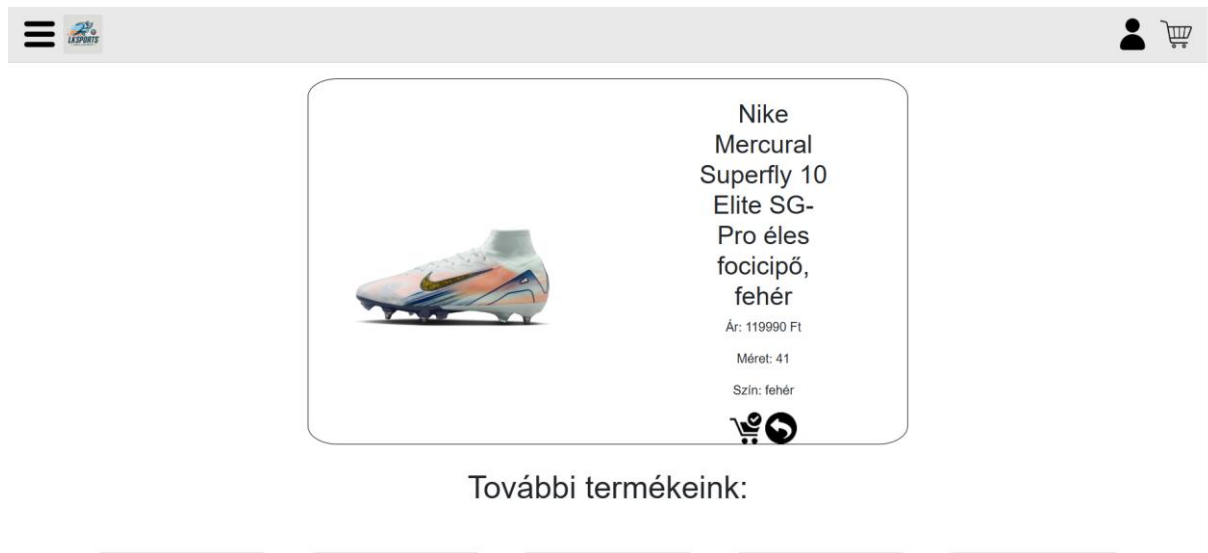
Kosár feltöltése:

A bejelentkezés követően el lehet kezdeni a kosár feltöltését különböző termékekkel labdarúgás, kosárlabda, futás oldalakon.



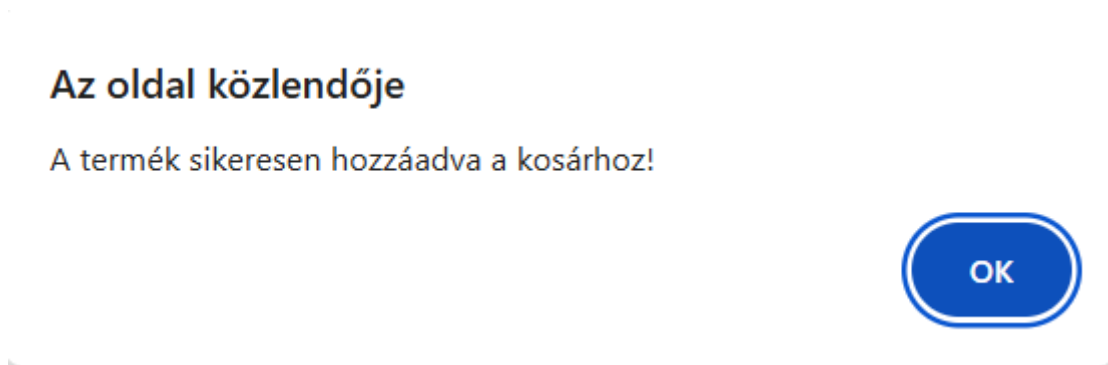
43.ábra: labdarugás.html

Ezek az oldalakon megtalálható egy kattints ide gomb, ami a lenyomást követően át visz a details.html oldalra, ahol kosárba is lehet tenni.



44. ábra: details.html

Mi után kosárba tettük ezt a szöveget fogjuk kapni:



45. ábra: details.html

2/2

Rendelés leadása:

Ha a felhasználó befejezte a vásárlás akkor a vissza tud menni a kosárba, ahol megtalálja az elmentett termékeket.

Kosár tartalma

 Nike Mercurial Superfly 10 Elite SG-Pro éles focicipő, fehér - 119990 Ft - 41 - fehér Törlés

Összesen: 119990 Ft

46.ábra: kosar.html

Rendelés leadása követően ezt az üzenetet kapjuk:

Sikeres rendelés

Köszönjük, hogy leadta a rendelését! Hamarosan felvesszük Önnel a kapcsolatot.

Bezárás

47.ábra: kosar.html

2/3

Rendelés létrejött:

Ha minden sikerült akkor a Xampp meg nyitását követően lkports_db adatbázisba rendelések táblánál megtaláljuk a rendelést.

	id	userid	osszeg	tartalom
<input type="checkbox"/>  Módosítás  Másolás  Törlés	1	1	119990	Nike Mercurial Superfly 10 Elite SG-Pro éles focici...

48.ábra: localhost: lkports_db

Ezt a rendelést a felhasználó is láthatja a profil menübe eddigi rendeléseid: rész alatt.

Eddigi rendeléseid:

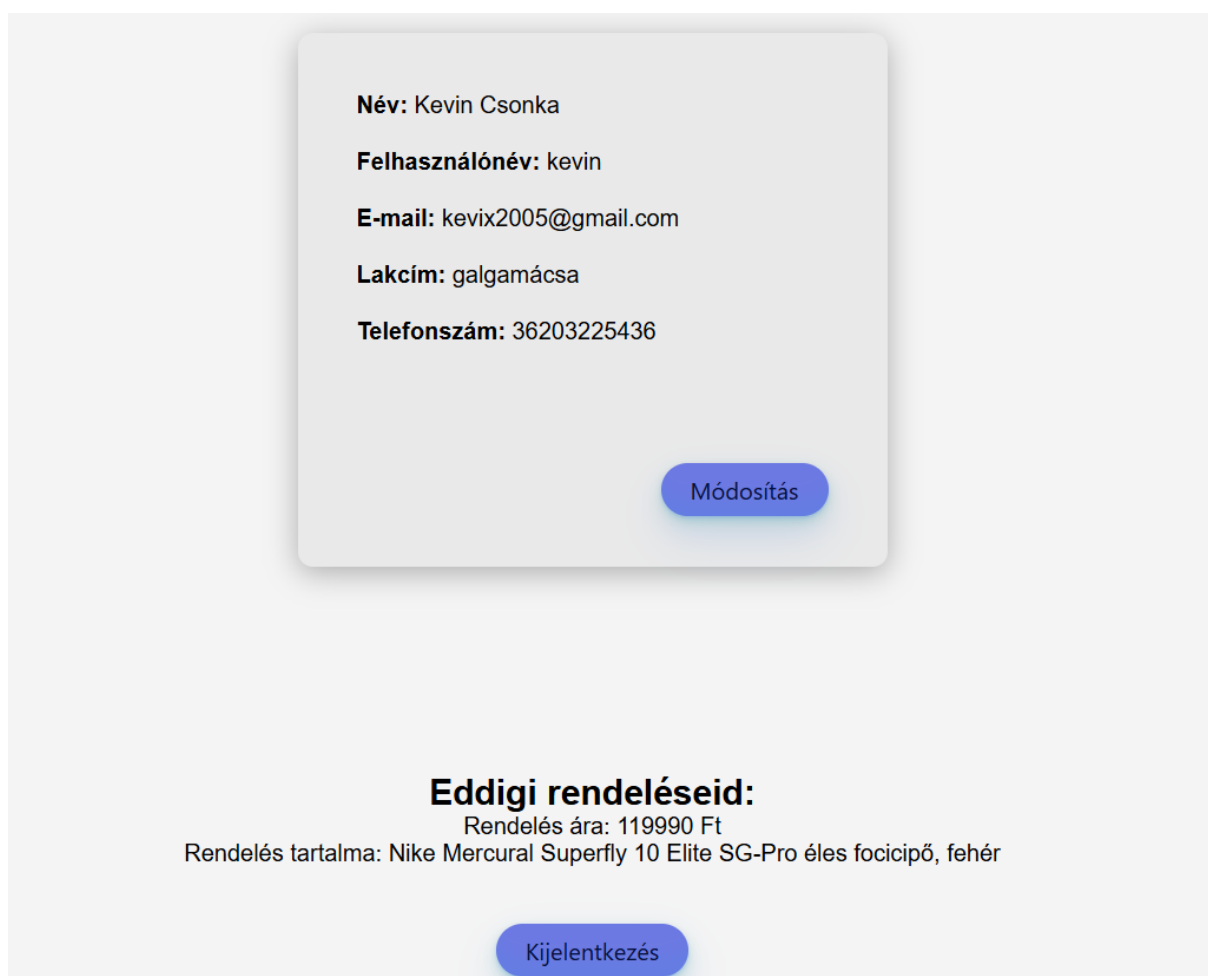
Rendelés ára: 119990 Ft
Rendelés tartalma: Nike Mercurial Superfly 10 Elite SG-Pro éles focicipő, fehér

49.ábra: profil.html

Kijelentkezés:

3/1

Ha a felhasználó befejezte az oldalon a vásárlást vagy esetleg szeretne fiokot váltani akkor a profil oldalon alul talál egy kijelentkezés gombot.



The image shows a user profile card and an order summary. The profile card is a light gray rounded rectangle with a shadow, containing the following text: **Név:** Kevin Csonka, **Felhasználónév:** kevin, **E-mail:** kevix2005@gmail.com, **Lakcím:** galgamácsa, **Telefonszám:** 36203225436. A blue rounded button labeled 'Módosítás' is at the bottom right. Below the card, the text 'Eddigi rendeléseid:' is followed by 'Rendelés ára: 119990 Ft' and 'Rendelés tartalma: Nike Mercurial Superfly 10 Elite SG-Pro éles focicipő, fehér'. A blue rounded button labeled 'Kijelentkezés' is at the bottom.

Név: Kevin Csonka

Felhasználónév: kevin

E-mail: kevix2005@gmail.com

Lakcím: galgamácsa

Telefonszám: 36203225436

Módosítás

Eddigi rendeléseid:

Rendelés ára: 119990 Ft

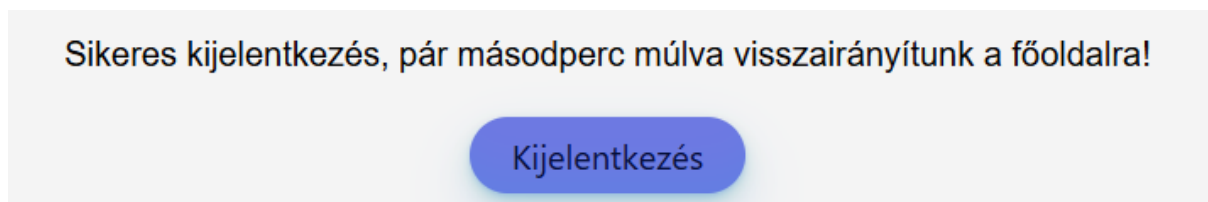
Rendelés tartalma: Nike Mercurial Superfly 10 Elite SG-Pro éles focicipő, fehér

Kijelentkezés

50.ábra: *profil.html*

3/2

Majd kijelentkezés gomb megnyomása után pár másodpercet követően kijelentkezik és visszavisz a főoldalra, ahol már vissza is jelentkezhetünk vagy esetleg el is hagyhatjuk a webshopot.



51.ábra: profil.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, 3 oldalt lehetne még bővíteni több termék családdal és több termékekkel feltölteni. A főoldalról ugyan ezen kellene fejleszteni több termék.

Az oldalról hiányzik a search bar, ami egy webáruháznál nagyon hasznos tud lenni, hogy a sokk termék között nem vesszünk el ennek a dizájnja meg volt, de túl nehéznek bizonyult a kód megírása, hogy lehessen keresni termékeket ezért másra fordítottuk az időt, hogy minden megfélon működhessen.

Termékeknél a szűrőt egy gombra szerettük volna váltani, amit megnyomása követően arrébb csúsztatja a termékeket és feljön a szűrők opciók, ahol, mint a jelenlegi oldalnál személyre lehet szabni, hogy milyen termékekre szeretnénk keresni szín, méret alapján.

7.0 Összegzés

Szakdolgozat egy komplex dolog, ami nem kis feladat. Rengeteg tudás és tapasztalatot szereztem a csapat munkáról annak előnyei és hátrányairól is. Fejleszteni tudtam a HTML, JS, CSS, és Xampp tudásomat. Leventét és Krisztiánt régóta ismerem ezért gyorsan megtaláltuk az összhangot, ami különösen jó jött az elején mivel nem tudtuk merre induljunk el a szakdolgozattal, de gyorsan összeraktuk a prototípust, amit folyamatosan tudtunk fejleszteni, bővíteni. Csapatmunkának köszönhetően mindenki olyan területen tudott dolgozni, ami neki közelebb állt. Nehézségeink sok volt hiszen egyikünk se dolgozott ekkora projekten, rengeteg kód után kellett nézünk és programozás során rengetek hiba lépet fel. Legnehezebb rész a termékek voltak annak dizájnolása, feltöltése. Itt lehetne még fejleszteni tovább több termékkel és több termék családdal.

Egy olyan weboldalt hoztunk létre, amin minden webáruház épül. A jövőbe szeretném/szeretnénk még fejleszteni a weboldalt mert amit itt összehoztunk ezt akár egyszer egy rendes weboldal lehet belőle, amit egy esetleges cég használhat a jövőbe.

7.1 Források

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

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

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

Kódhoz szükséges oldalak:

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

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

<https://freefrontend.com/>

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

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

képek: <https://focivilag.hu/hu>

képek: <https://www.pngegg.com/>

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: localhost: phpmyadmin
23. ábra: server.js

24. ábra: server.js
25. ábra: server.js
26. ábra: server.js
27. ábra: server.js
28. ábra: server.js
29. ábra: server.js
30. ábra: register.html
31. ábra: register.html
32. ábra: login.html
33. ábra: profile.html
34. ábra: profile.html
35. ábra: foci.html
36. ábra: details.html
37. ábra: details.html
38. ábra: details.html
39. ábra: details.html
40. ábra: kosar.html
41. ábra: kosar.html
42. ábra: kosar.html
43. ábra: labdarugás.html
44. ábra: details.html
45. ábra: details.html
46. ábra: kosar.html
47. ábra: kosar.html

48. ábra: localhost: lksports_db

49. ábra: profil.html

50. ábra: profil.html

51. ábra: profil.html