

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



SZOFTVERFEJLESZTŐ ÉS TESZTELŐ

Záródolgozat

Cím: LKSPORTS

Készítette: Keresztes Krisztián

Konzulens: Mizere Robin Roland




2025

KONZULTÁCIÓS LAP

Szoftverfejlesztő
SZJ 50 613 1203

A hallgató neve: KERESZTES KRISZTIÁN Osztálya: 13/D

A záródolgozat témája: LKSPORTS
LABDARÚGÁS, KOSÁRLABDA, FUTÁS
SPORTOKHOZ SZÜKSÉGES SPORTSZEREKET
ÉRTÉKESÍTŐ WEBÁRUHÁZ

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/

VÁCI SZC PETŐFI SÁNDOR MŰSZAKI TECHNIKUM,

GIMNÁZIUM ÉS KOLLÉGIUM

Nyilatkozat

Alulírotttanuló kijelentem, hogy a záródolgozat saját munkám
eredménye,

a felhasznált szakirodalmat és eszközöket (könyv, tanulmány, kézirat, internetes forrás,

személyes közlés stb.) idézőjel és pontos hivatkozások nélkül nem építettem be,

azonosíthatóan közölte. Az elkészült záródolgozatban található eredményeket a

képzőintézmény és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja, a

titkosításra vonatkozó esetleges megkötések mellett.

Aszód, 2025.....

Tartalomjegyzék

1. Bevezető	6
2.0 Felhasználói dokumentáció	7
2.1 Kategóriák sportágak szerint.....	7
2.2 Keresési és szűrési lehetőségek	7
2.3 Részletes termékleírások és vásárlói értékelések	7
2.4 Kosárkezelés és rendelési folyamat.....	7
2.5 Felhasználói fiók és vásárlási előzmények	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	28
6.4 Részletes feladat-specifikáció, algoritmusok	30
6.5 Forráskód	36
6.6 Tesztelési dokumentáció	39

6.7 Továbbfejlesztési lehetőségek	41
7.0 Összegzés	42
7.1 Források.....	42
8. Ábrajegyzék:.....	43

1. Bevezető

Szakedolgozatomat Csonka Kevinnel és Hatházi Leventével közösen készítem, és projektünk az **LKSPORTS** névre hallgat. Az ötlet egy olyan weboldal, mint a Decathlon vagy a SportsDirect, ami az online sportfelszerelés vásárlást könnyíti meg. Egy átlátható, de közben egy összetett weboldalt hozunk létre, ami felhasználó barát. Az egyik legfontosabb szempont, hogy a felhasználói élmény emlékezetes legyen a vásárlók részére, és persze az sem elhanyagolható, hogy legjobb termékeket a legjobb áron tudják megszerezni nálunk a vásárlók.

A weboldalt **HTML** alapokon készítjük, és a dizájnelemeket **CSS**-sel valósítjuk meg. Az oldal működéséhez szükséges háttérfolyamatokat, mint például az interaktív elemeket, **JavaScript** segítségével fogjuk megvalósítani. Emellett szükség lesz egy adatbázisra is, ahol a felhasználók adatai (pl. felhasználónév, email) és az értékesített termékek (ruházat, cipők, sporteszközök stb.) tárolása történik. Ezt az adatbázist **MySQL** segítségével építjük, amit JavaScript-alapú csomagokkal (pl. Express, MySQL, Cors stb.) kezelünk majd.

A feladatokat már felosztottuk egymás között: én a **HTML** részért vagyok felelős, Kevin a **CSS**-ért, Levi pedig a **JavaScript**-en dolgozik. Ugyanakkor fontos kiemelni, hogy bármelyikünk hozzájárulhat a többiek munkájához, ha ötletünk támad, vagy szeretnénk valamit továbbfejleszteni. Így a csapatmunka dinamikusan működik.

A szakedolgozat során a kommunikációhoz **Discordot**, a projekt megosztásához és verziókezeléshez pedig **GitHubot** használunk

2.0 Felhasználói dokumentáció

Az LKSPORTS célkitűzése, hogy a felhasználók számára gyors és kényelmes online vásárlási élményt nyújtson, miközben teljes körű megoldást kínál a sportfelszerelések és kiegészítők beszerzésére. A webáruház számos hasznos funkcióval rendelkezik, amelyek megkönnyítik a vásárlást, a termékek kezelését és javítják a felhasználói élményt.

2.1 Kategóriák sportágak szerint

Az LKSPORTS termékeit különféle sportágak szerint csoportosítja – például futás, kerékpározás, labdajátékok és egyéb mozgásformák –, így a vásárlók gyorsan és egyszerűen eligazodhatnak a kínálatban, és könnyedén megtalálhatják a keresett árucikkeket

2.2 Keresési és szűrési lehetőségek

A vásárlás gyorsítását különféle szűrők segítik, mint az ár, márka, méret, szín, illetve egyéb fontos jellemzők alapján. A beépített fejlett keresőfunkció segítségével a felhasználók kulcsszavak alapján is könnyedén megtalálhatják a számukra releváns termékeket.

2.3 Részletes termékleírások és vásárlói értékelések

Minden termékhez részletes információk állnak rendelkezésre, beleértve a specifikációkat, használati útmutatókat és termékképeket. Emellett a vásárlók más felhasználók véleményeit és értékeléseit is elolvashatják, ami segítséget nyújt a döntéshozatalban.

2.4 Kosárkezelés és rendelési folyamat

A termékek egyszerűen hozzáadhatók a kosárhoz, a rendelés leadása pedig gyors és biztonságos folyamaton keresztül történik, amely tartalmazza a fizetési és szállítási opciókat is. A vásárlás regisztrált felhasználóként és vendégként is lehetséges.

2.5 Felhasználói fiók és vásárlási előzmények

A regisztrációval rendelkező vásárlók saját fiókot kapnak, amelyben nyomon követhetik korábbi rendeléseiket, elmenthetik kedvenc termékeiket, és kezelhetik az értesítéseiket vagy akciós ajánlataikat. Ez a funkció hozzájárul a jövőbeli vásárlások egyszerűsítéséhez és az átlátható rendeléskövetéshez.

3.0 Hardver követelmények:

3.1 Minimális konfiguráció:

- **Processzor:** Legalább 1.8 GHz-es, kétmagos CPU
- **Memória (RAM):** Minimum 4 GB
- **Tárhely:** Legalább 10 GB szabad lemezterület
- **Grafikus kártya:** Minimum 128 MB videomemória, amely képes megjeleníteni a weboldalon alkalmazott grafikai elemeket
- **Képernyőfelbontás:** Minimum 1366x768
- **Internetkapcsolat:** Legalább 1 Mbps letöltési sebesség

Ajánlott rendszerkonfiguráció:

Processzor: 2.5 GHz-es vagy gyorsabb, négymagos CPU

Memória (RAM): 8 GB vagy több

Tárhely: Minimum 20 GB szabad hely

Grafikus kártya: Legalább 512 MB videomemória

Képernyőfelbontás: 1920x1080 vagy magasabb

Internetkapcsolat: Legalább 10 Mbps sebesség a zavartalan működés érdekében

Szoftverkövetelmények:

Támogatott operációs rendszerek:

Windows 10 vagy 11 (64 bites verziók)

macOS X 10.15 vagy újabb

Linux – Ubuntu 20.04 vagy frissebb kiadás

Ajánlott böngészők:

Google Chrome (legfrissebb stabil verzió)

Mozilla Firefox (legfrissebb stabil verzió)

Microsoft Edge (legfrissebb stabil verzió)

Safari (macOS rendszeren)

3.2 Egyéb szoftverek:

JavaScript és CSS:

A weboldal interaktív működéséhez elengedhetetlen a JavaScript támogatása, valamint a modern CSS szabványok használata. A böngészőknek képesnek kell lenniük ezek futtatására és megjelenítésére.

Adatbázis-szerver:

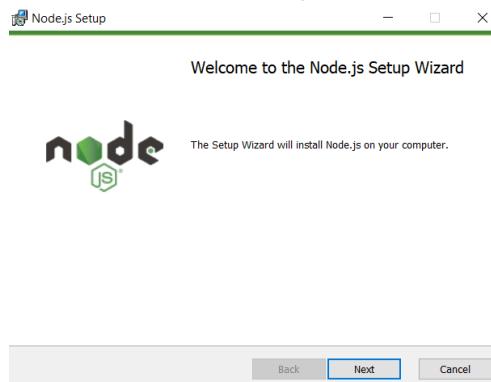
Amennyiben az oldal adatbázis-kezelést igényel – például termékek vagy felhasználói fiókok tárolására –, a MySQL használata javasolt.

Felhasználói oldal:

A weboldal használatához nincs szükség külön szoftver telepítésére a felhasználók számítógépén, mivel a modern böngészők önmagukban képesek biztosítani az oldal teljes funkcionalitását.

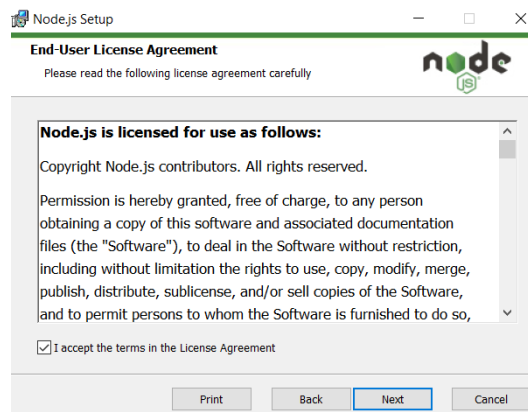
4.0 Telepítési útmutató

4.1 Node.js



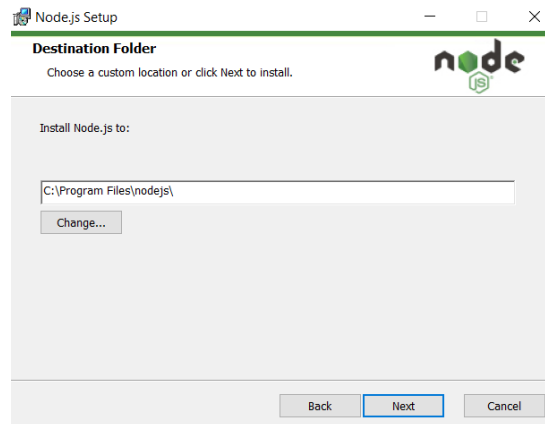
1. ábra: Node.js

Töltsd le a telepítőfájlt a Node.js hivatalos weboldaláról (<https://nodejs.org/en>), majd indítsd el a letöltött .exe fájlt. A telepítés megkezdéséhez kattints a „Tovább” (Next) gombra az első megjelenő ablakban.



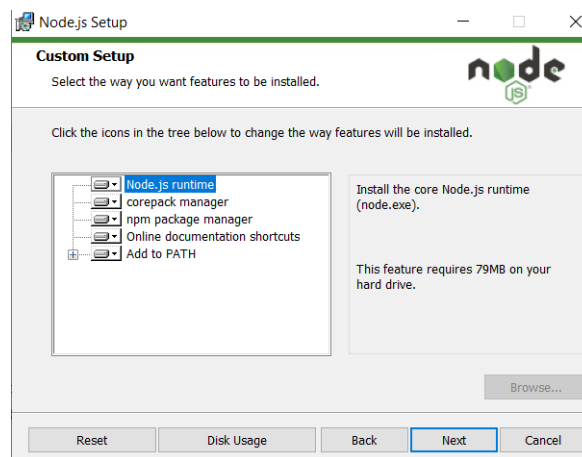
2. ábra: Node.js

A következő lépésben el kell fogadnod a licencszerződést, hogy folytathasd a telepítést. Az elfogadás után a „Tovább” (Next) gombra kattintva léphetsz tovább a telepítési folyamatban.



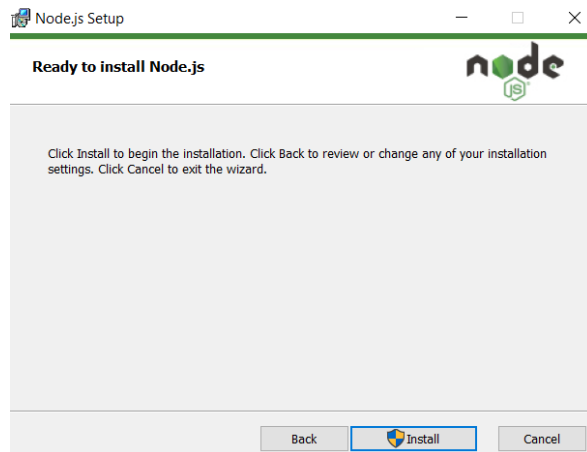
3. ábra: Node.js

A következő oldalon meg kell adnod a telepítés helyét, vagy elfogadhatod az alapértelmezett útvonalat. Ezután a „Tovább” (Next) gombra kattintva folytathatod a telepítést.



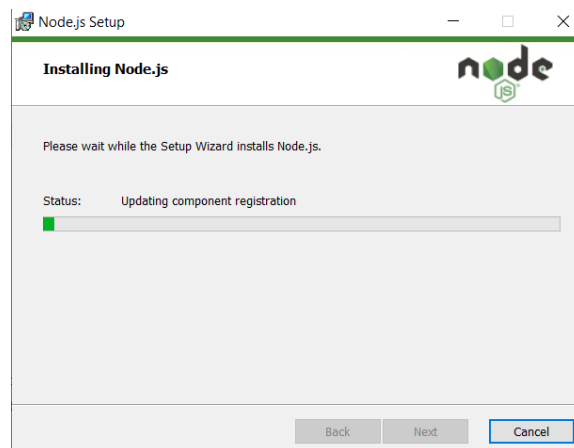
4. ábra: Node.js

Ebben az ablakban kiválaszthatod a szükséges fájlokat, majd a „Tovább” (Next) gombra kattintva folytathatod a folyamatot.



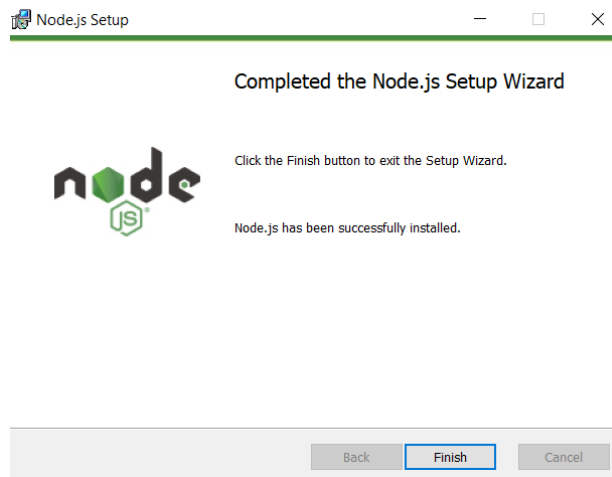
5. ábra: Node.js

Miután a telepítő átvitt a következő oldalra, a telepítés befejeződéséhez az „Install” gombra kell kattintani.



6. ábra: Node.js

A telepítő elindítja a fájlok telepítését, majd a folyamat befejezése után a „Tovább” (Next) gombra kattintva fejezheted be a telepítést.

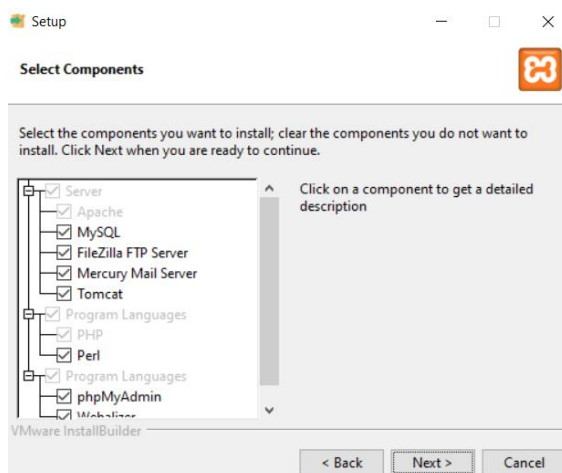


7. ábra: Node.js

Amikor a telepítő átvizsgálja a következő oldalra, az „Befejezés” gombra kattintva zárhatod be az ablakot, így a Node.js már használatra kész.

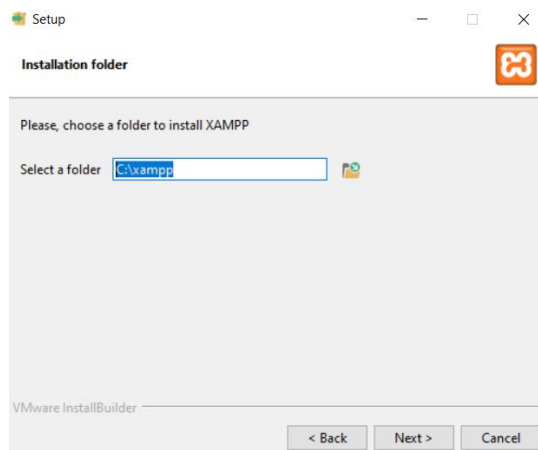
4.2 XAMPP

Az első lépés, hogy töltsd le a telepítőfájlt az <https://www.apachefriends.org/hu/index.html> oldalról, majd kattints a „Tovább” (Next) gombra a folytatáshoz.



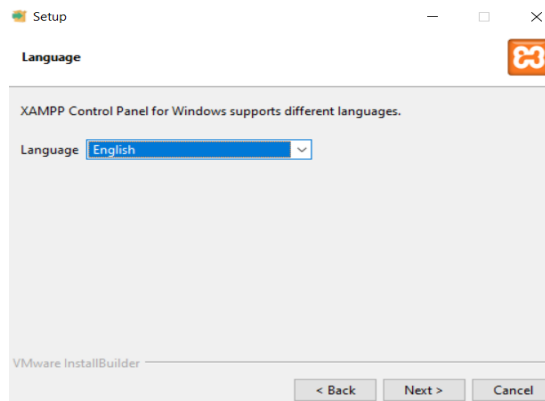
8. ábra: XAMPP

Ebben az ablakban javasolt az alapértelmezett beállításokkal haladni, amit a „Tovább” (Next) gomb megnyomása tesz lehetővé.



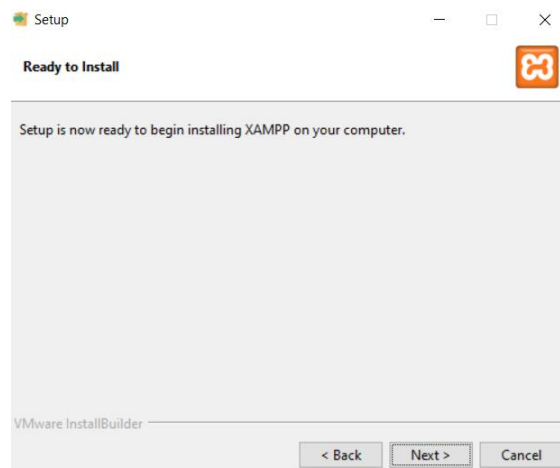
9. ábra: XAMPP

Ebben az ablakban kiválaszthatod a telepítési útvonalat, ahova a programot szeretnéd elhelyezni, majd a „Tovább” (Next) gomb segítségével léphetsz a következő oldalra.



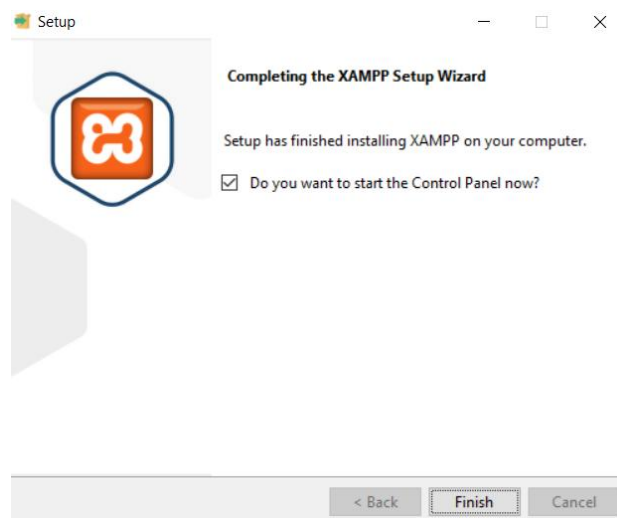
10. ábra: XAMPP

A telepítés ezen részében kiválaszthatod a kívánt nyelvet, majd a „Tovább” (Next) gomb megnyomásával folytathatod a folyamatot.



11. ábra: XAMPP

Most, hogy a XAMPP készen áll a telepítésre, a „Tovább” (Next) gombra kattintva indíthatod el a telepítést.



12. ábra: XAMPP

Végül, a „Finish” gombra kattintva bezárhatod a telepítőt, mely esetenként az azonnali indítást is lehetővé teszi.



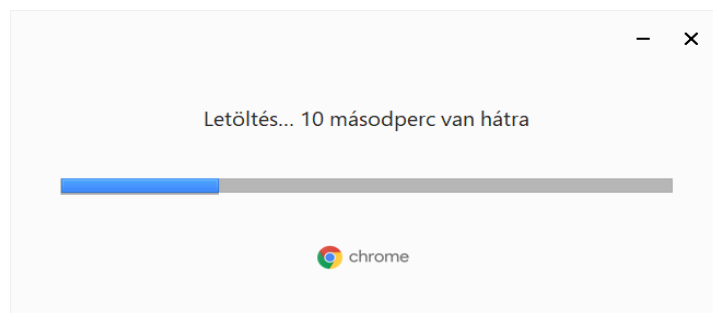
13. ábra: XAMPP

13. ábra: XAMPP

A XAMPP elkezdni futtatni magát.

4.3 Google Chrome

A böngészőt a https://www.google.com/intl/hu_hu/chrome/ oldalról lehet letölteni, majd a letöltött telepítőt kell lefuttatni.



14. ábra: Google Chrome

A telepítő automatikusan elvégzi a fájlok telepítését.

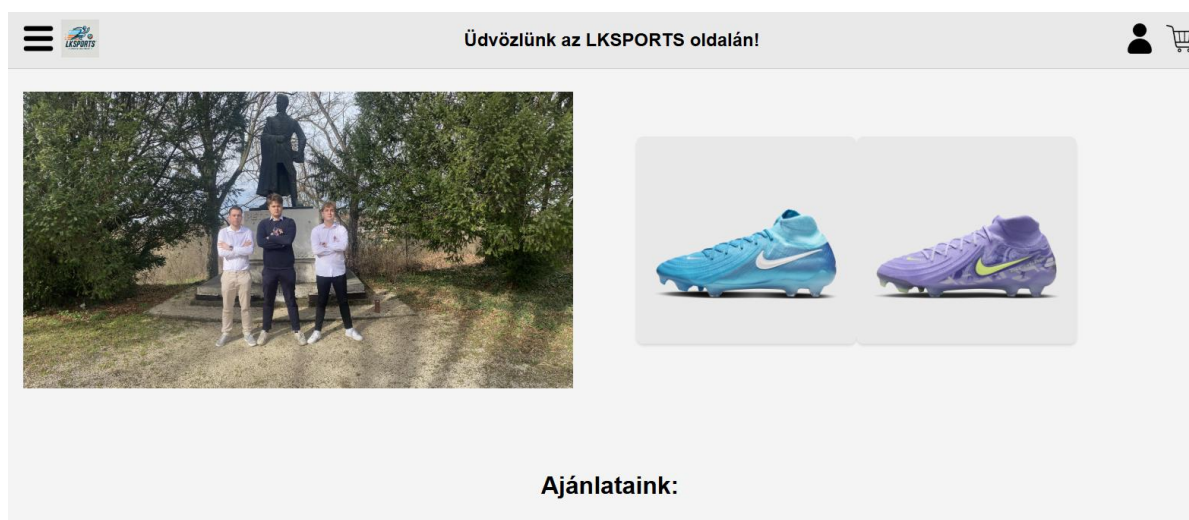


15. ábra: Google Chrome

A folyamat befejezése után lehetőség nyílik bejelentkezni, illetve a böngésző további testreszabására.

5.0 A program használatának a részletes leírása

5.1 Főoldal





16.ábra: Index.html

Belépéskor a látogatót egy képes slideshow köszönti, amely az aktuálisan akciós termékeket mutatja be. A főoldalon található keresőmező segítségével könnyedén kereshetők a termékek (*). A bal felső sarokban elhelyezett menüikon kattintásával elérhetők a sportágak szerinti kategóriák, míg az LKSPORTS logóra kattintva bármelyik oldalról visszatérhetünk a főoldalra.

A jobb felső sarokban látható kosár ikon lehetővé teszi a korábban kosárba helyezett termékek megtekintését és megvásárlását. Ha az egerrel a „profil” ikon fölé viszed az egeret, megjelenik a bejelentkezési oldal, ahol beléphetsz, vagy – ha még nem rendelkezel fiókkal – regisztrálhatsz. A „profil” ikonra kattintva pedig átkerülsz a saját profiloldaladra, ahol módosíthatod személyes adataidat (*).

5.2 Kosár

Kosár tartalma

	Nike Mercurial Superfly 10 Elite SG-Pro éles focicipő, fehér - 119990 Ft - 41 - fehér	Törés
	Air Jordan 1 Zoom CMFT Shoes Grey Blue - 60000 Ft - 41 - szürke	Törés

Összesen: 179990 Ft

Kosár ürítése

Név
Keresztes Krisztián

Felhasználónév
donci

Cím
Szada

Telefonszám
06207468070

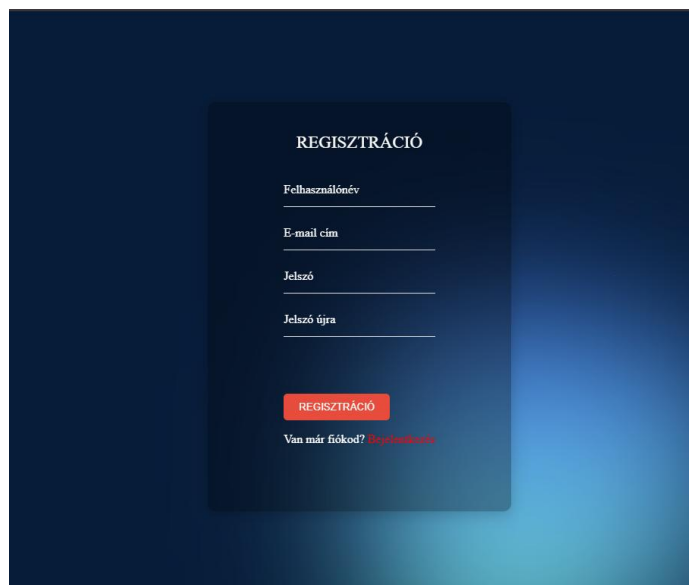
Email
krisz.050206@gmail.com

Rendelés leadása

17. ábra: Kosár.html

A főoldalon a „Kosár” gombra kattintva jutsz arra az oldalra, ahol megtekintheted a kosárban található termékeket, és egyenként vagy egyszerre törölheted azokat. Alatta láthatók a felhasználói adatok, míg az oldal alján a „Rendelés leadása” gomb található, amelynek segítségével véglegesítheted a rendelést, feltéve, hogy minden kötelező adatot kitöltöttél.

5.3 Bejelentkezés/Regisztráció



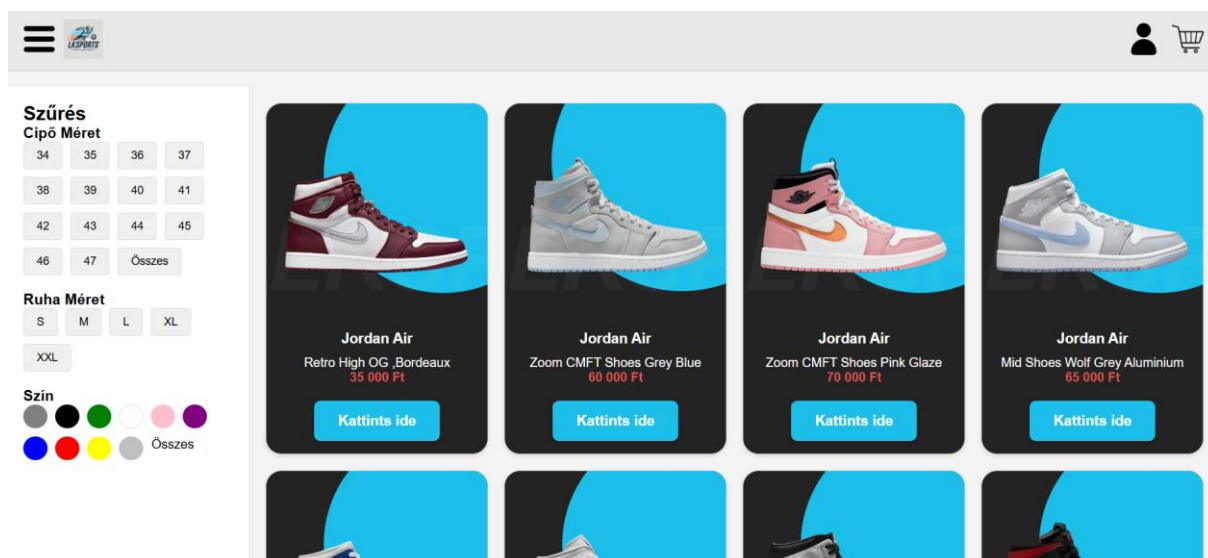
The image shows a registration form titled "REGISZTRÁCIÓ" (Registration) centered on a dark blue background. The form itself is a lighter blue rectangle. It contains the following fields and elements:

- Title: REGISZTRÁCIÓ
- Field 1: Felhasználónév (Username) with a text input line.
- Field 2: E-mail cím (Email address) with a text input line.
- Field 3: Jelszó (Password) with a text input line.
- Field 4: Jelszó újra (Repeat password) with a text input line.
- Button: A red button with the text "REGISZTRÁCIÓ".
- Text: "Van már fiókod? [Bejelentkezés](#)" (Do you already have an account? [Login](#)).

18. ábra: Register.html

A regisztrációs és a bejelentkezési oldal hasonló felépítésű: a főoldalról elérheted a bejelentkezési felületet, és ha még nem rendelkezel regisztrált fiókkal, innen navigálhatsz a regisztrációs oldalra. Itt a megadott mezők kitöltése szükséges a regisztráció befejezéséhez.

5.4 Termékoldalak



19.ábra: kosarlabda.html

Weboldalunkon külön termékoldalak találhatók a labdarúgás, futás és kosárlabda témakörökben. Ezek az oldalak különféle funkciókat kínálnak, például szűrési lehetőséget méret és szín szerint. Mivel minden termék rendelkezik meghatározott színnel és mérettel, a szűrés segítségével egyszerűen leszűkíthetők a találatok. A kiválasztott termékekre kattintva lehetőség van azokat a kosárba helyezni.

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.

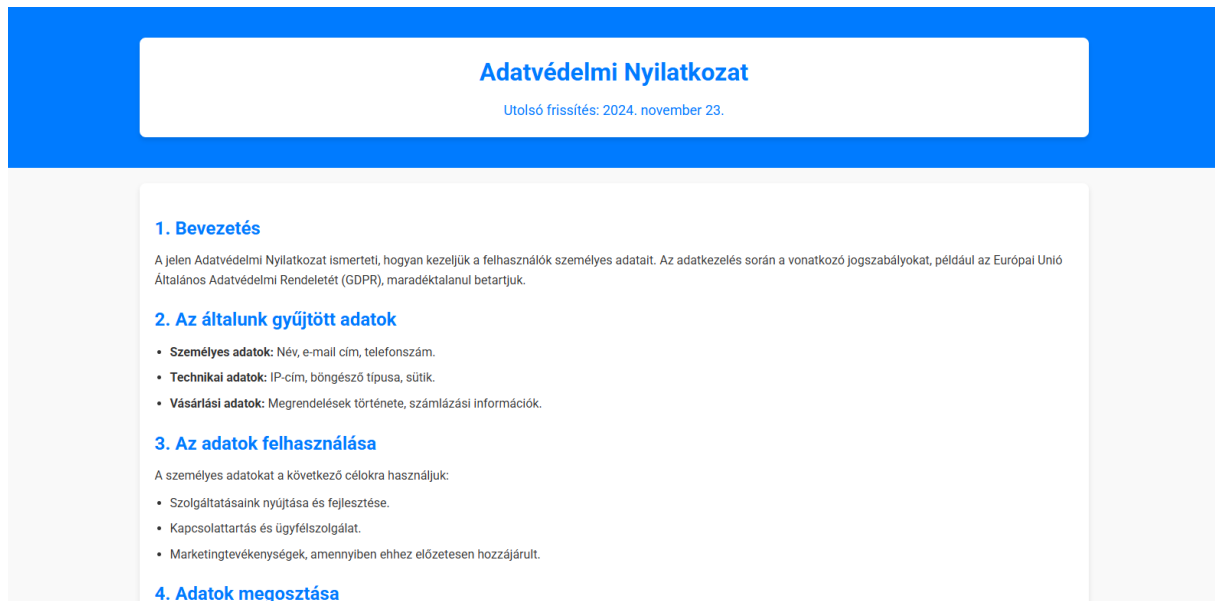
6. Vásárlási feltételek

20.ábra:felhaszfeltet.html

Weboldalunkhoz felhasználási feltételek tartoznak, amelyek több részből állnak:

1. Bevezetés
2. A szolgáltatás részletes leírása
3. Felhasználói fiókok kezelése
4. Szellemi tulajdonjogok
5. A felelősség korlátozása
6. Vásárlási feltételek
7. Adatvédelem
8. A feltételek módosítása
9. Kapcsolatfelvételi lehetőségek

5.6 Adatvédelmi Nyilatkozat



21.ábra:adatnyil.html

Weboldalunk külön adatvédelmi nyilatkozattal is rendelkezik, amelynek célja a felhasználók személyes adatainak védelme. A nyilatkozat a következő pontokat tartalmazza:

1. Bevezetés
2. Az általunk gyűjtött adatok köre
3. Az adatok felhasználásának módja
4. Az adatok továbbítása, megosztása
5. Az adatok biztonsága
6. Cookie-k (sütik) használata
7. A felhasználók jogai
8. Elérhetőségek, kapcsolatfelvétel

5.7 Profil

Név: Keresztes Krisztián
Felhasználónév: donci
E-mail: krisz.050206@gmail.com
Lakcím: Szada
Telefonszám: 06207468070

Módosítás

Eddigi rendeléseid:
Rendelés ára: 23990 Ft
Rendelés tartalma: Nike Mercurial Superfly 10 Club FG stoplis focipő, fekete

Kijelentkezés

22.ábra: profile.html

A profiloldalon a bejelentkezett felhasználók megtekinthetik és szerkeszthetik saját adataikat, valamint visszakereshetik korábbi rendeléseiket. Amennyiben a felhasználó szerepköre az adatbázisban „1” -es értéket kap, megjelenik számára egy admin gomb, amelyen keresztül hozzáférést kap az adminisztrációs felülethez. Ezen a felületen lehetősége nyílik valamennyi felhasználó adatainak módosítására. Az oldal alján található a „Kijelentkezés” gomb, amelyre kattintva a felhasználó biztonságosan kiléptethető a fiókjából.

5.8 Admin

Id	Név	Felhasználónév	E-mail	Lakcím	Telefonszám
Role					
Megváltoztatás					

23.ábra: admin.html

Az admin gombra kattintva megnyílik az „Admin felület” nevű oldal, ahol az adminisztrátor jogosultsággal rendelkező felhasználók szerkeszthetik az összes felhasználó adatait – a jelszavak kivételével.

6.0 Fejlesztői dokumentáció

6.1 Témaválasztás indoklása

Az LKSPORTS egy sportáruház koncepció, amely működésében a Decathlonhoz vagy a SportsDirecthez hasonlítható. Azért esett erre a témára a választásunk, mert mindhárman szenvedélyesen érdeklődünk a sport iránt, és rendszeresen használjuk ezeket a webáruházakat. Felhasználói tapasztalataink során azonban több hiányosságot is tapasztaltunk, amelyek inspiráltak bennünket arra, hogy megalkossuk az LKSPORTS-ot – egy olyan sportáruházat, amely letisztult megjelenésével és felhasználóbarát funkcióival a kezdő felhasználók számára is könnyen átlátható és egyszerűen használható.

6.2 Az alkalmazott fejlesztői eszközök

A program fejlesztéséhez használt eszközök:

Fejlesztői környezet:

Visual Studio Code

Programozási nyelvek:

HTML5

CSS3

JavaScript

Node.js

Bootstrap

JavaScript-hez használt modulok és bővítmények:

express

cors

mysql2

bcryptjs

jsonwebtoken

dotenv

Adatbázis- és szerverkapcsolathoz használt eszközök:

XAMPP

MySQL

JavaScript

Csapatmunka és feladatmegosztás eszközei:

GitHub

Discord

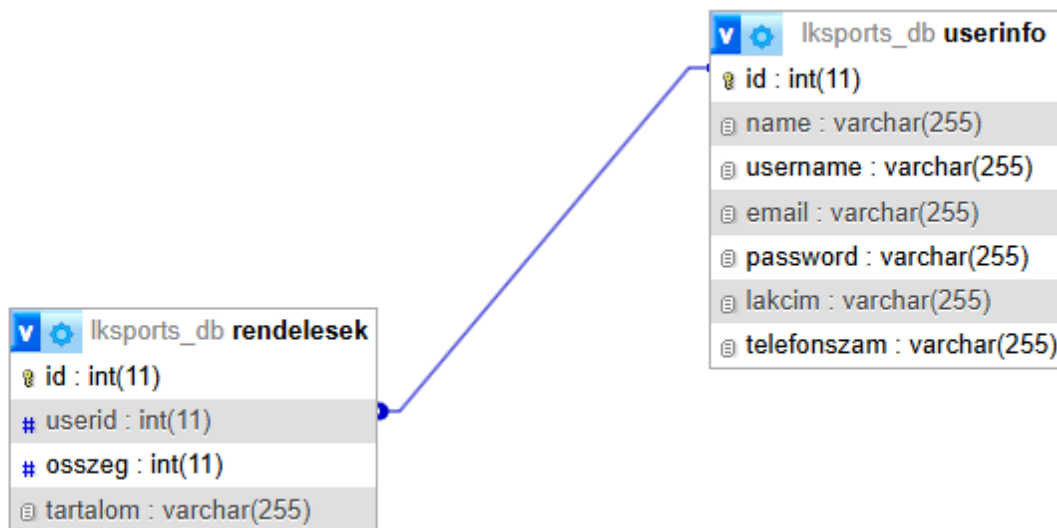
Dokumentáció készítéséhez:

Microsoft Word

6.3 Adatmodell

Adatbázisunk **MySQL** nyelven készült, a weboldalunkhoz való kapcsolódást pedig **JavaScript** segítségével valósítottuk meg. A kapcsolat a server és az adatbázis között így zökkenőmentesen működik.

Az adatbázis neve: **lksports_db**.



24.ábra: localhost: phpmyadmin

Az adatbázis két táblát tartalmaz:

1. Tábla: userinfo

Ez a tábla a regisztrált felhasználók adatait tárolja:

<input type="checkbox"/>	Módosítás	Másolás	Törölés	2	Hatházi Levente	admin	hathazilevi465@gmail.com	\$2b\$10\$gs53fEOBHchRxdBaw8tY7.8fjucNmrZJofDCZD37f4K...	2117. Isaszeg Bacsó Béla utca 4	06303065255
--------------------------	-----------	---------	---------	---	-----------------	-------	--------------------------	--	---------------------------------	-------------

25.ábra: userinfo

ID: Egyedi azonosító minden felhasználónak; ez alapján ismeri fel a rendszer, hogy ki jelentkezett be.

Name: A felhasználó teljes neve. Regisztrációkor nem kötelező, de a profiloldalon kötelező megadni rendelés leadása előtt.

Username: A felhasználó által választott felhasználónév, amelyet regisztráció során kell megadni.




Email: A felhasználó e-mail címe, megadása regisztrációkor kötelező.

Password: A felhasználó jelszava, szintén kötelező a regisztrációnál; az adatbázisba titkosított formában kerül mentésre.

Lakcím: A felhasználó címe. Regisztrációkor nem kötelező, de rendelés előtt ki kell tölteni a profiloldalon.

Telefonszám: A felhasználó elérhetősége, szintén nem kötelező regisztrációkor, de rendelés előtt szükséges megadni.

2. Tábla: rendelések

	ID	UserID	Osszeg	Tartalom
<input type="checkbox"/>  Módosítás  Másolás  Törlés	2	2	143980	Nike Mercurial Superfly 10 Club FG stoplis focicip...

26.ábra: rendelések

ID: A rendelés egyedi azonosítója (elsődleges kulcs).

UserID: Annak a felhasználónak az azonosítója, aki leadta a rendelést – ez alapján rendeli a rendszer a rendelést a megfelelő felhasználóhoz.

Osszeg: A rendelés végösszege.

Tartalom: A rendeléshez tartozó termékek listája.

6.4 Részletes feladatspecifikáció, algoritmusok

Backend

Adatbázis kapcsolat létrehozása:

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

27. ábra: Server.js

A fenti képen látható, hogyan kapcsolódik JavaScript-ben 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

Ebben a függvényben a rendszer lekéri a regisztrációhoz szükséges adatokat, majd ellenőrzi, hogy létezik-e már felhasználó a megadott felhasználónévvel. Ha nem talál egyezést, a jelszót titkosítja, és az adatokat elmenti az adatbázisba.

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

A rendszer először ellenőrzi, hogy a megadott felhasználónév helyes-e, majd validálja a hozzá tartozó jelszót is. Sikeres hitelesítés esetén egy, a felhasználó adatait tartalmazó tokent állít ki, amely egy órán keresztül érvényes.

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

A kód először ellenőrzi, hogy a felhasználó rendelkezik-e érvényes tokennel, majd a felületen megadott adatok alapján frissíti az információkat.

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

A függvény először megvizsgálja, hogy a megadott jelszó megfelel-e a szükséges hosszúságnak. Ha nem, a felhasználónak újra kell írnia. Ezen kívül azt is ellenőrzi, hogy a két

```
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 regisztrációs és bejelentkezési függvények nagyjából azonosak, mivel mindkettő esetén a konstansok az oldalon megadott értékeket kapják meg, majd ezeket továbbítják a backendnek, amely ezeket az adatokat eljuttatja az adatbázisba.

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

Az adatokat lekéri az adatbázisból, majd az oldalon található módosított értékekkel frissíti azokat.

6.5 Forráskód

```
</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

A termékoldalon egy termékre kattintva a felhasználó átirányításra kerül egy új oldalra, ahol lehetősége van a kiválasztott terméket a kosárba helyezni.

```
<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

Ez a kód jeleníti meg a Details oldalt ahol belehet rakni a kosárba terméket.

```

        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

A "Kosárba" gomb megnyomásakor a termék hozzáadásra kerül a kosárhoz.

```

<div class="product-list">
  <div class="product-item" data-size="42" data-color="fekete"
    onclick="atiranyitas3('Asics Gel-Nimbus 27 férfi futócipő', '35000', '../kepek/asics noi fekete.png', '42', 'fekete')">
    
    <h3>Asics</h3>
    <p>Gel-Nimbus 27 férfi futócipő</p>
    <p class="price">35 000 Ft</p>
    <button class="button">Kattints ide</button>
  </div>

```

42.ábra: index.html

Ezzel a kóddal jelenítődig meg a termék a weboldalon.

```

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();
    })
  })
});

```

43.ábra:kosar.html

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

```

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`;
}

```

44.ábra: kosar.html

Ez a függvény betölti a kosár tartalmát a böngésző helyi tárolójából, majd megjeleníti a kosár elemeit egy listában, és kiszámítja azok összegét. Minden elemhez tartozik egy "Törlés" gomb is, amellyel az adott termék eltávolítható a kosárból.

6.6 Tesztelési dokumentáció

1/1

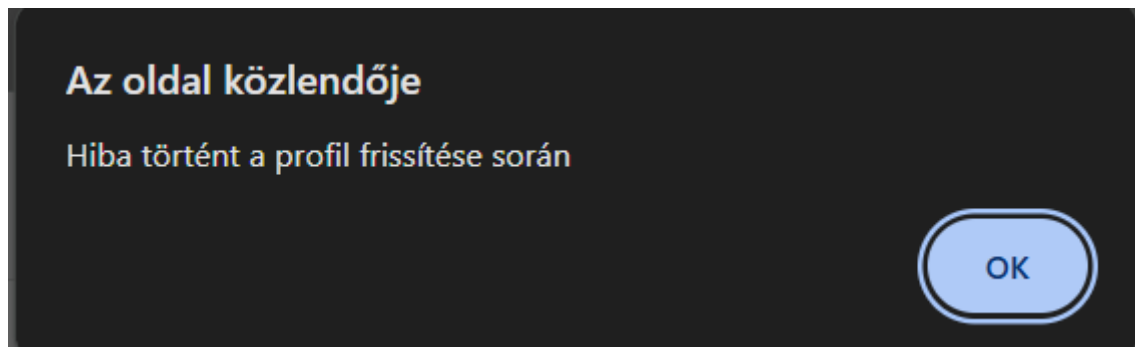
Nincs a kosárban termék:

45.ábra: kosar.html

Ha nincs a kosárban termék nem lehet rendelést leadni.

1/2

Profil módosítása:



46.ábra: profile.html

Ha nincs kitöltve az összes adat nem lehet a profilt frissíteni.

3/1

Bejelentkezés nélküli rendelés:

Kosár tartalma

Összesen: 0 Ft

Kosár ürítése

Jelentkezz be ahhoz hogy le tudd adni a rendelésed!

Név

Felhasználónév

Cím

Telefonszám

Email

Rendelés leadása

47.ábra: kosar.html

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

6.7 Továbbfejlesztési lehetőségek

A termékek jelenleg nem adatbázisban vannak tárolva, ami megkönnyítheti a jövőbeni fejlesztéseket. Az oldal jelenleg háromféle sportághoz tartozó termékeket kínál, de ezek köre a későbbiekben egyszerűen bővíthető.

7.0 Összegzés

A szakdolgozatunk segített jobban megérteni a weboldalak működését, HTML, JS és CSS kapcsolatot. Sokat fejlődtem frontend és backend fejlesztésben és a csapatmunkában is. Jobban együtt tudtam dolgozni Kevinnel és Leventével, mint ahogy tavaly gondoltam volna. Sok éjszakába nyúló programozás összekovácsolta a csapatot és szerintem ez volt az oka, hogy ilyen hatékonyan tudtunk együtt dolgozni. A legnagyobb kihívást számomra a frontend jelentette mivel a tanulás közben kellett élesben is megmutatni, hogy mit is tanultunk. Ennek a programnak az elkészítése közben rájöttem, hogy nem véletlen választottam a programozást 9.-ben, jövőben weboldal készítéssel szeretnék foglalkozni.

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>

Focicipők: <https://focivilag.hu/hu>

Futócipők: <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: kosarlabda.html
20. ábra: felhaszfeltet.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: index.html

43. ábra: kosar.html

44. ábra: kosar.html

45. ábra: kosar.html

46. ábra: profile.html

47. ábra: kosar.html