**VÁCI SZAKKÉPZÉSI CENTRUM   
BORONKAY GYÖRGY   
MŰSZAKI TECHNIKUM ÉS GIMNÁZIUM**

**SZAKDOLGOZAT**

**Balogh Ádám – Schönberger Dominik**

**2023.**

**VÁCI SZAKKÉPZÉSI CENTRUM   
BORONKAY GYÖRGY   
MŰSZAKI TECHNIKUM ÉS GIMNÁZIUM**



**SZAKDOLGOZAT**

**Használt Sportszer**

|  |  |
| --- | --- |
| Konzulens: Wiezl Csaba | Készítette: Balogh Ádám  Schönberger Dominik |

# **Hallgatói nyilatkozat**

Alulírottak, ezúton kijelentjük, hogy a szakdolgozat saját, önálló munkánk, és korábban még sehol nem került publikálásra.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Balogh  Ádám |  |  | Schönberger Dominik |

# **Konzultációs lap**

Vizsgázók neve: Balogh Ádám, Schönberger Dominik

Szakdolgozat címe: Használt Sportszer Webshop

Program nyújtotta szolgáltatások:

* Bejelentkezés/regisztráció
* Saját hirdetés feltöltés/szerkesztés
* Sportszerek eladása
* Profil szerkesztés
* Hirdetés böngészés

|  |  |  |
| --- | --- | --- |
| Sorszám | A konzultáció időpontja | A konzulens aláírása |
| 1. |  |  |
| 2. |  |  |
| 3. |  |  |
| 4. |  |  |
| 5. |  |  |
| 6. |  |  |

|  |  |  |
| --- | --- | --- |
| A szakdolgozat beadható: |  | A szakdolgozatot átvettem: |
| Vác, 2023. |  | Vác, 2023. |
| Konzulens |  | A szakképzést folytató intézmény felelőse |

**Tartalomjegyzék**

[**Hallgatói nyilatkozat** 3](#_Toc131510707)

[**Konzultációs lap** 4](#_Toc131510708)

[**Tartalomjegyzék** 5](#_Toc131510709)

[**Témaválasztás** 7](#_Toc131510710)

[**1** **Fejlesztői dokumentáció** 8](#_Toc131510711)

[1.1 Rövid alkalmazás ismertetés 8](#_Toc131510712)

[1.2 Felhasznált alkalmazások 8](#_Toc131510713)

[1.2.1 Visual Studio Code 8](#_Toc131510714)

[1.2.2 Brackets 8](#_Toc131510715)

[1.3 Felhasznált fejlesztői nyelvek 9](#_Toc131510716)

[1.3.1 PHP 9](#_Toc131510717)

[1.3.2 HTML 9](#_Toc131510718)

[1.3.3 CSS 9](#_Toc131510719)

[1.3.4 Javascript 10](#_Toc131510720)

[1.4 Felhasznált technológiák 10](#_Toc131510721)

[1.4.1 PHPMailer 10](#_Toc131510722)

[1.4.2 JSON 10](#_Toc131510723)

[1.4.3 MySql 11](#_Toc131510724)

[1.4.4 Xampp 11](#_Toc131510725)

[1.4.5 Bootstrap 5 11](#_Toc131510726)

[1.5 Használati esetmodell, szerepkörök 11](#_Toc131510727)

[1.6 Adatbázis 12](#_Toc131510728)

[1.6.1 Adatszerkezet 12](#_Toc131510729)

[1.6.2 Adattáblák 12](#_Toc131510730)

[1.6.3 Utólag hozzáadott adattáblák 16](#_Toc131510731)

[1.6.4 Adatbázis kapcsolatok 20](#_Toc131510732)

[1.7 Modulok és kód ismertetése 21](#_Toc131510733)

[1.7.1 Adatbázis csatlakozás 21](#_Toc131510734)

[1.7.2 Admin felület 21](#_Toc131510735)

[1.7.3 Authentication 23](#_Toc131510736)

[1.7.4 CSS / Cascading Style Sheets 24](#_Toc131510737)

[1.7.5 Képek 25](#_Toc131510738)

[1.7.6 HTML kód nélküli funkciók 25](#_Toc131510739)

[1.7.7 HTML oldalak/funkciók 31](#_Toc131510740)

[1.8 Tesztelés/Tesztesetek 36](#_Toc131510741)

[1.8.1 Tesztesetek 37](#_Toc131510742)

[1.9 Továbbfejlesztési lehetőségek 38](#_Toc131510743)

[**2** **Felhasználói dokumentáció** 39](#_Toc131510744)

[2.1 Alkalmazás rövid ismertetése 39](#_Toc131510745)

[2.2 Rendszerkövetelmények 39](#_Toc131510746)

[2.3 Alkalmazás telepítése, szükséges beállítások 39](#_Toc131510747)

[2.3.1 XAMPP telepítése 39](#_Toc131510748)

[2.3.2 XAMPP elindítása 40](#_Toc131510749)

[2.3.3 A fájlok bemásolása a htdocs mappába 41](#_Toc131510750)

[2.3.4 Adatbázis futtatása 41](#_Toc131510751)

[2.3.5 Webáruház megnyitása 41](#_Toc131510752)

[2.4 Alkalmazás használata 41](#_Toc131510753)

[2.4.1 Navigáció, lábjegyzet 41](#_Toc131510754)

[2.4.2 Regisztráció, Bejelentkezés 42](#_Toc131510755)

[2.4.3 Főoldal 43](#_Toc131510756)

[2.4.4 Kosaram 45](#_Toc131510757)

[2.4.6 Rendelés leadása 46](#_Toc131510758)

[2.4.7 Kontakt 47](#_Toc131510759)

[2.4.8 Hirdetés feltöltése 48](#_Toc131510760)

[2.4.9 Profilom 49](#_Toc131510761)

[2.5 Felhasználható eszközök 50](#_Toc131510762)

[**3** **Irodalomjegyzék** 51](#_Toc131510763)

[3.1 Internetes források 51](#_Toc131510764)

[3.2 Könyvek 51](#_Toc131510765)

[**4** **Mellékletek** 51](#_Toc131510766)

# **Témaválasztás**

Az év elején alaposan átgondoltuk szakdolgozatunk témáját, és végül a webes irány mellett döntöttünk, hiszen mindketten nagyobb kedvvel és tapasztalattal rendelkezünk a webes alkalmazásokhoz szükséges nyelvek terén. Ezt követően el kellett döntenünk, milyen témában építjük fel az általunk tervezett webáruházat. A sport mindkettőnk életében meghatározó szerepet tölt be, és ráadásul nem találtunk olyan használt sporteszközökkel foglalkozó webáruházat, amely kifejezetten erre a célra fókuszál. Ezért úgy határoztunk, hogy ezt a témakört helyezzük a középpontba, és az általunk tervezett oldal lehetőséget biztosít a használt sporteszközök vásárlására és eladására egyaránt.

Célunk, hogy minél szélesebb körben ismertté és elérhetővé tegyük az általunk létrehozott webáruházat, és minél több visszajelzést kapjunk a felhasználóktól, hogy hatékonyan fejleszthessük az alkalmazást. Fontos számunkra, hogy a még felhasználható sporteszközök ne a szemétben kössenek ki, hanem olyan kezekbe kerüljenek, amelyeknek örömet okoznak és hosszú távon hasznosak lehetnek. Emellett szeretnénk támogatni azokat a sportolókat is, akik nem engedhetik meg maguknak a legújabb felszereléseket, de szívesen vásárolnának jó minőségű, áron aluli eszközöket.

Az általunk létrehozott webáruház kialakításában Schönberger Dominik vállalta a különböző funkciók megvalósítását, beleértve az adatbázissal való csatlakozást, a beléptetést, a regisztrációt, a profilok funkcióit, mint a kijelentkezés, profil adatok, illetve a kiszállítási cím módosítása, a hirdetések feltöltését és szerkesztését, az e-mail rendszert és az admin felületet. Balogh Ádám feladatai közé tartozott a frontend felépítése, reszponzív oldalak kialakítása, az adatbázis létrehozása és a Schönberger Dominik által írt funkciók beintegrálása projectmunkába. Fontosnak tartjuk megemlíteni, hogy a fejlesztés során csapatunk nem használt előre megírt weboldalakat, a template oldalak használata szembementek a mi elveinkkel, szerettük volna, hogyha a projekt 100%-ban a saját munkánk gyümölcse. Természetesen a feladatokban voltak átfedések, és amikor szükséges volt, segítettünk egymásnak. Csapatunk harmadik tagja Hajszter Botond volt, aki a tanév vége felé befejezte tanulmányait, így a rábízott feladatok is hirtelen a mi nyakunkba szakadtak. Ezen probléma kiküszöbölése kisebb-nagyobb sikerrel sikerült.

# **Fejlesztői dokumentáció**

## Rövid alkalmazás ismertetés

Alkalmazásunk egy sportszerekkel foglalkozó webshop. Felhasználóink kényelmesen vásárolhatnak legyenek bárhol az országban. Célunk az volt, hogy azokat az embereket is ösztönözzük a sportolásra akiknek nem áll módjukban megvásárolni a drágábbnál drágább sportszereket vagy csak nem szeretnének ennyi pénz áldozni rá.

## Felhasznált alkalmazások

### Visual Studio Code [[1]](#footnote-1)

A **Visual Studio Code** ingyenes, nyílt forráskódú kódszerkesztő, melyet a Microsoft fejleszt Windows, Linux és macOS operációs rendszerekhez. Támogatja a hibakeresőket, valamint beépített Git támogatással rendelkezik, továbbá képes az intelligens kódkiegészítésre az IntelliSense segítségével. A VSCode-ban a felhasználók megváltoztathatják a kinézetet (témát), megváltoztathatják a szerkesztő gyorsbillentyű-kiosztását, az alapértelmezett beállításokat és még sok egyebet. Támogatja a kiegészítőket, melyek segítségével további funkciók, testreszabási lehetőségek érhetőek el. A VSCode az Electron nevű keretrendszeren alapszik, amellyel asztali környezetben futtatható Node.js alkalmazások fejleszthetőek.

### Brackets[[2]](#footnote-2)

A **Brackets** ingyenes nyílt forráskódú szöveg- és forráskódszerkesztő melyet HTML-ben, CSS-ben és JavaScriptben írtak kifejezetten webfejlesztéshez. A programot az Adobe fejlesztette és karbantartása a GitHub-on történik. A Brackets elérhető Machintos, Linux és Windows platformokra. 2021. szeptember 1-jén az Adobe befejezte a támogatást.

## Felhasznált fejlesztői nyelvek

### PHP[[3]](#footnote-3)

A PHP elterjedt nyílt forráskódú szerver-oldali programozási nyelv, amely az SSI (Server Side Includes) és a Perl ötvözésére hivatott. Szintaktikája leginkább a C programozási nyelvéhez hasonlít. Megalkotója Rasmus Lerdorf. Mára azonban egy egész csapat foglalkozik a nyelvvel. Széles körben való elterjedtségét C programozási nyelvhez való hasonlóságának, sebességének és egyszerűségének köszönheti. A PHP nem csak szerveroldali alkalmazásoknál alkalmazható, bár asztali alkalmazások esetében nem terjedt el.

### HTML[[4]](#footnote-4)

A **HTML** egy leíró nyelv, melyet weboldalak készítéséhez fejlesztettek ki, és mára már internetes szabvánnyá vált a W3C (World Wide Web Consortium) támogatásával. Egy HTML dokumentum alapértelmezésként ISO-8859-1, azaz nyugat-európai kódolást használ. Gyakran előforduló hiba szokott lenni, hogy nincs beállítva a charset paraméter a fejléc content attribútumában, annak ellenére, hogy a dokumentum nem nyugat-európai kódolású szöveget tartalmaz.

### CSS[[5]](#footnote-5)

A CSS (Cascading Style Sheets) egy olyan leírónyelv, melynek segítségével különböző stíluslapokat hozhatunk létre és ágyazhatunk be HTML honlapjainkba. Ezek a stíluslapok befolyásolják az oldal megjelenését (design-ügyileg kiváló nyelv), meghatározhatjuk vele, hogy az egyes HTML tagek (bekezdések, táblázatok, címsorok stb.) hogyan jelenjenek meg, meghatározhatjuk a méretüket, színüket, stílusukat, típusukat stb.

### Javascript[[6]](#footnote-6)

A JavaScript, mint neve is mutatja, egy script nyelv, amit legelőször a Netscape Navigator 2.0 támogatott. A Javascript a WWW-ért született. Jelenleg nem is létezik más implementációja, mint ami a böngészőkben és a webszerverekben fut. A Sun és a Netscape által kifejlesztett parancsnyelvnek két változata létezik. Az ún. LiveWire lehetővé teszi, hogy szerver oldali alkalmazásokat készítsünk, ami hasonló a CGi-hez (Common Gateway interface). A másik a kliens oldali, ami lehetővé teszi egyszerűbb programozási feladatok elkészítését a Weben (pl. mozgó HTML elemek, interaktív funkciók). A kliens és a szerver oldali rész egy külön világ, szintakszisuk ugyan megegyezik, de a programozói interfész teljesen más. A kliens oldali JavaScript a böngészőkön keresztül jelenik meg számunkra. Mégpedig úgy, hogy a JavaScript kódot az általános HTML kód közé ékeljük. Amikor a böngésző megjeleníti a HTML kódot, akkor a <script> és </script> közé írt kódrészleteket nem megjeleníti, hanem értelmezi, és futtatja, ezáltal dinamikus viselkedést kölcsönöz az oldalnak. (Van lehetőség a JavaScript kódot egy külön állományba helyezni és a HTML-ből pusztán hivatkozni erre az állományra.)

## Felhasznált technológiák

### PHPMailer[[7]](#footnote-7)

A PHPMailer egy kódkönyvtár, amely biztonságosan és egyszerűen küld e-maileket PHP-kódon keresztül egy webszerverről. 2001-től a PHPMailer az egyik legnépszerűbb megoldása az e-mail küldéseknek a PHP-n belül.

### JSON[[8]](#footnote-8)

A különféle adatokkal való munkavégzés ismerete fontossá vált. A programozóknak, fejlesztőknek és informatikai szakembereknek át kell vinniük a feltöltött adatstruktúrákat bármely nyelvről olyan formátumba, amelyet más nyelvek és platformok is felismernek. A JavaScript Object Notation (JSON) az az adatcsere-formátum, amely ezt lehetővé teszi.

### MySql

Egy biztonságos adatbázisszolgáltatás.

### Xampp[[9]](#footnote-9)

Aki adatbázissal támogatott webszervert szeretne üzemeltetni, annak össze kell hangolnia a webszerver, az adatbázisszerver és a PHP szolgáltatásait, valamint a kommunikációs portokat. A XAMPP fejlesztői felismerték a lehetőséget, hogy ezeket a beállításokat mindenkinek el kell végeznie, viszont nem biztos, hogy mindenki rendelkezik adminisztrátori jogokkal és ilyen szintű ismeretekkel, ezért összeállítottak egy keretrendszert, amelyet kezdő fejlesztők is könnyedén használhatnak.

### Bootstrap 5 [[10]](#footnote-10)

A Bootstrap 5 a Bootstrap legújabb verziója, amely a legnépszerűbb HTML, CSS és JavaScript keretrendszer reszponzív webhelyek létrehozásához.

## Használati esetmodell, szerepkörök

## Adatbázis

### Adatszerkezet

**MySQL kliens verziója**: 4.9.0.1

**Adatbázis** **neve:** sportszer\_ab

**Tárolómotor:** InnoDB

**Alapértelmezett illesztés**: utf8\_hungarian\_ci

**SQL parancs:**

CREATE DATABASE IF NOT EXISTS `sportszer\_ab`

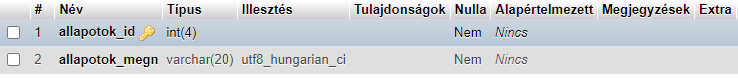
DEFAULT CHARACTER SET utf8 COLLATE utf8\_hungarian\_ci;

USE ` sportszer\_ab `;

### Adattáblák

#### ’allapotok’ tábla

A termék lehetséges állapotát tartalmazza (pl.: új, használt).



* allapotok\_id: elsődleges kulcs a termék állapotának azonosítója
* allapotok\_megn: a termék állapotának megnevezése

**Tábla létrehozása:**

CREATE TABLE `allapotok` (

`allapotok\_id` int(4) NOT NULL,

`allapotok\_megn` varchar(20) COLLATE utf8\_hungarian\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

#### ’markak’ tábla

A termék márkáját tartalmazza



* markak\_id: elsődleges kulcs, a márka azonosítója
* markak\_megn: a márka megnevezése

**Tábla létrehozása:**

CREATE TABLE `markak` (

`markak\_id` int(6) NOT NULL,

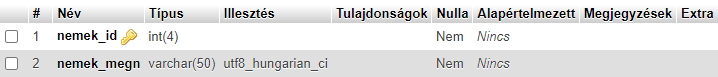
`markak\_megn` varchar(40) COLLATE utf8\_hungarian\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

#### ’nemek’ tábla

A termék célközönségét tartalmazza a nem alapján (férfi, női, unisex)



* nemek\_id: elsődleges kulcs, a nem azonosítója
* nemek\_megn: a nem megnevezése

**Tábla létrehozása:**

CREATE TABLE `nemek` (

`nemek\_id` int(4) NOT NULL,

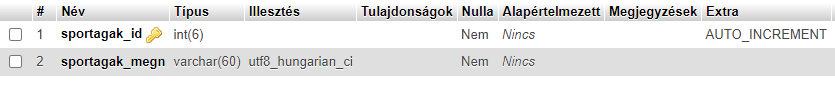
`nemek\_megn` varchar(50) COLLATE utf8\_hungarian\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

#### ’sportagak’ tábla

A termék sportági besorolását tartalmazza



* sportagak\_id: elsődleges kulcs, a sportág azonosítója
* sportagak\_megn: a sportág megnevezése

**Tábla létrehozása:**

CREATE TABLE `sportagak` (

`sportagak\_id` int(6) NOT NULL,

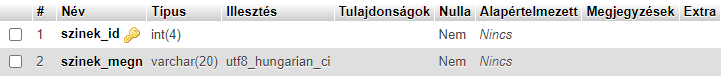
`sportagak\_megn` varchar(60) COLLATE utf8\_hungarian\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

#### ’szinek’ tábla

A termék domináns színét jelöli



* szinek\_id: a domináns szín azonosítója
* szinek\_megn: a domináns szín megnevezése

**Tábla létrehozása:**

CREATE TABLE `szinek` (

`szinek\_id` int(4) NOT NULL,

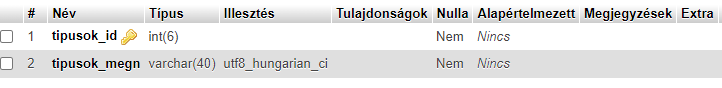
`szinek\_megn` varchar(20) COLLATE utf8\_hungarian\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

#### ’tipusok’ tábla

A termék típusát jelöli (pl.: cipő, melegítő felső)



* tipusok\_id: a termék típusának azonosítója
* tipusok\_megn: a termék típusának megnevezése

**Tábla létrehozása:**

CREATE TABLE `tipusok` (

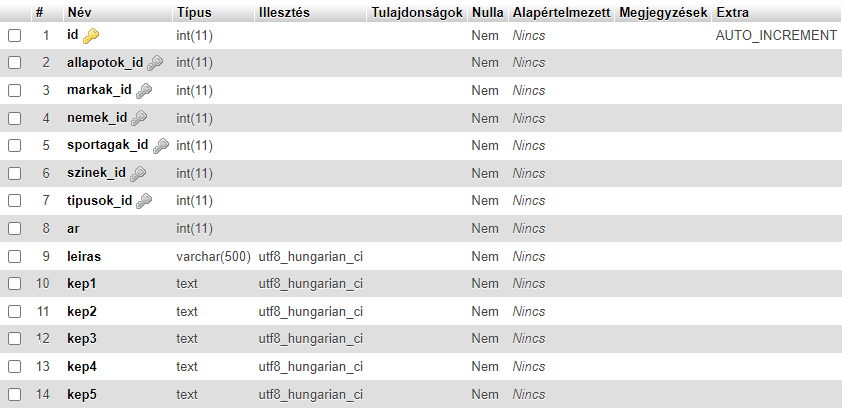
`tipusok\_id` int(6) NOT NULL,

`tipusok\_megn` varchar(40) COLLATE utf8\_hungarian\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

#### ’hirdetesek’ tábla

Ez a tábla egy átfogó tábla, tartalmazza a feltöltött hirdetések adatait.

* id: elsődleges kulcs, a hirdetés azonosítója
* allapotok\_id: idegen kulcs, az állapot azonosítója
* markak\_id: idegen kulcs, a márka azonosítója
* nemek\_id: idegen kulcs, a nem azonosítója
* sportagak\_id: idegen kulcs, a sportág azonosítója
* szinek\_id: idegen kulcs, a domináns szín azonosítója
* tipusok\_id: idegen kulcs, a termék típusának azonosítója
* ar: a termék ára
* leiras: a termék leírása
* kep1: a hirdetés 1. képe, ez a hirdetés ,,profilképe”
* kep2: a hirdetés 2. képe
* kep3: a hirdetés 3. képe
* kep4: a hirdetés 4. képe
* kep5: a hirdetés 5. képe

**Tábla létrehozása:**

CREATE TABLE `hirdetesek` (

`id` int(11) NOT NULL,

`allapotok\_id` int(11) NOT NULL,

`markak\_id` int(11) NOT NULL,

`nemek\_id` int(11) NOT NULL,

`sportagak\_id` int(11) NOT NULL,

`szinek\_id` int(11) NOT NULL,

`tipusok\_id` int(11) NOT NULL,

`ar` int(11) NOT NULL,

`leiras` varchar(500) COLLATE utf8\_hungarian\_ci NOT NULL,

`telszam` varchar(40) COLLATE utf8\_hungarian\_ci NOT NULL,

`kep1` text COLLATE utf8\_hungarian\_ci NOT NULL,

`kep2` text COLLATE utf8\_hungarian\_ci NOT NULL,

`kep3` text COLLATE utf8\_hungarian\_ci NOT NULL,

`kep4` text COLLATE utf8\_hungarian\_ci NOT NULL,

`kep5` text COLLATE utf8\_hungarian\_ci NOT NULL

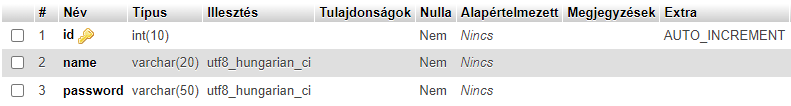
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

### Utólag hozzáadott adattáblák

A következő 4 tábla a weboldal fejlesztésének későbbi szakaszában lett hozzáadva az adatbázishoz. Ezen táblák kapcsolatainak felépítése, javítása a továbbfejlesztés része.

#### ’admin’ tábla

Az fejlesztői felülethez szükséges adattábla



* id: elsődleges kulcs, a fejlesztői(admin) profil azonosítója
* name: a fejlesztői profil neve
* password: a profil jelszava

**Tábla létrehozása:**

CREATE TABLE `admin` (

`id` int(10) NOT NULL,

`name` varchar(20) COLLATE utf8\_hungarian\_ci NOT NULL,

`password` varchar(50) COLLATE utf8\_hungarian\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

#### ’felhasznalo’ tábla

A regisztrált felhasználók adatait tartalmazó tábla



* id: elsődleges kulcs, a felhasználó azonosítója
* name: a felhasználó neve
* email: a felhasználó e-mail címe
* number: a felhasználó telefonszáma
* password: a felhasználó jelszava
* address: a felhasználó lakcíme

**Tábla létrehozása:**

CREATE TABLE `felhasznalo` (

`id` int(4) NOT NULL,

`name` varchar(30) COLLATE utf8\_hungarian\_ci NOT NULL,

`email` varchar(50) COLLATE utf8\_hungarian\_ci NOT NULL,

`number` varchar(50) COLLATE utf8\_hungarian\_ci NOT NULL,

`password` varchar(60) COLLATE utf8\_hungarian\_ci NOT NULL,

`address` varchar(300) COLLATE utf8\_hungarian\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

#### ’cart’ tábla

Minden felhasználó kosárba tett termékeinek az adatait tartalmazza



* id: elsődleges kulcs, a kosár azonosítója
* user\_id: a felhasználó azonosítója, aki a terméket a kosárba helyezte
* hir\_id: azon hirdetés, termék azonosítója, melyet a felhasználó a kosárba tett
* marka: a kosárba helyezett termék márkája
* ar: a kosárba helyezett termék ára
* termektipus: a kosárba helyezett termék típusa (pl.: cipő, melegítő felső)
* kep1: a kosárba helyezett termék elsőszámú képe

**Tábla létrehozása:**

CREATE TABLE `cart` (

`id` int(100) NOT NULL,

`user\_id` int(100) NOT NULL,

`hir\_id` int(100) NOT NULL,

`marka` varchar(100) COLLATE utf8\_hungarian\_ci NOT NULL,

`ar` int(10) NOT NULL,

`termektipus` varchar(100) COLLATE utf8\_hungarian\_ci NOT NULL,

`kep1` text COLLATE utf8\_hungarian\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

#### ’rendelesek’ tábla

A felhasználó megrendelt termékeinek adatait, illetve a felhasználó néhány adatát tartalmazza



* id: a rendelés azonosítója
* user\_id: a rendelést leadó felhasználó azonosítója
* nev: a rendelést leadó felhasználó neve
* telszam: a rendelést leadó felhasználó telefonszáma
* email: a rendelést leadó felhasználó e-mail címe
* modszer: a rendelést leadó felhasználó fizetésének módja
* cím: a rendelést leadó felhasználó kiszállítási címe
* ossz\_rendeles: az rendelt termékek összegzése
* ossz\_ar: a teljes rendelés összege
* datum: a rendelés leadásának időpontja
* fizetes: a fizetés állapota

**Tábla létrehozása:**

CREATE TABLE `rendelesek` (

  `id` int(100) NOT NULL,

  `user\_id` int(100) NOT NULL,

  `nev` varchar(20) NOT NULL,

  `telszam` varchar(20) NOT NULL,

  `email` varchar(50) NOT NULL,

  `modszer` varchar(50) NOT NULL,

  `cim` varchar(500) NOT NULL,

  `ossz\_rendeles` varchar(1000) NOT NULL,

  `ossz\_ar` int(100) NOT NULL,

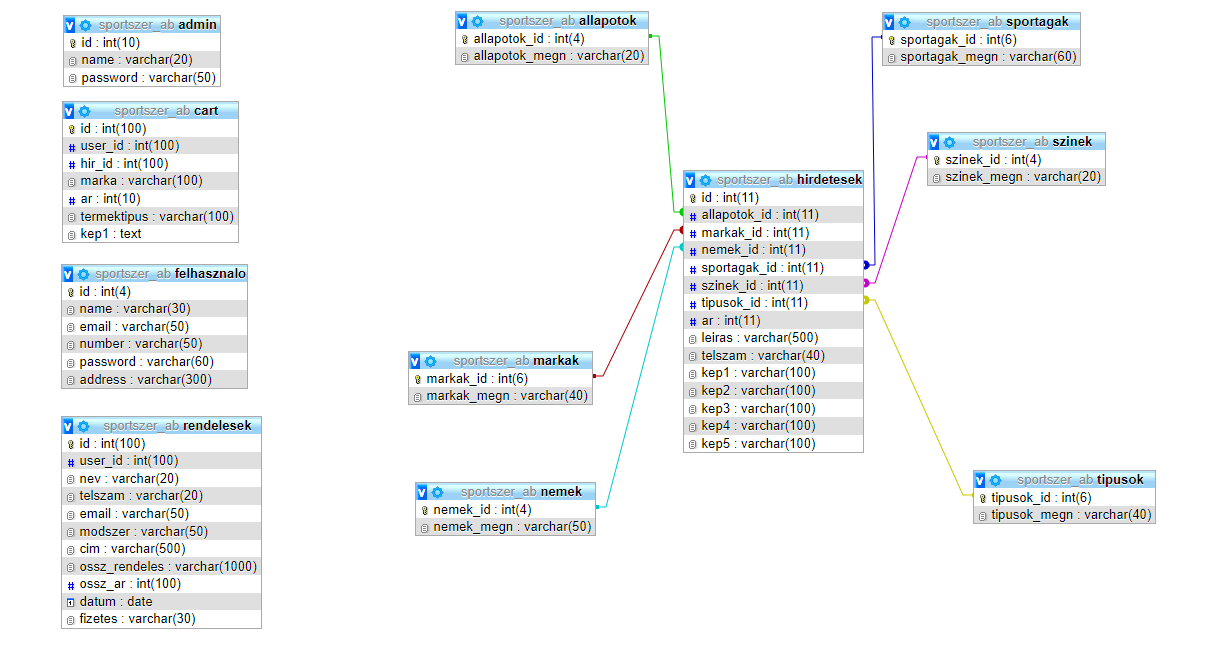
  `datum` date NOT NULL DEFAULT current\_timestamp(),

  `fizetes` varchar(30) NOT NULL DEFAULT 'fizetés nem történt meg'

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

COLLATE=utf8\_hungarian\_ci;) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

### Adatbázis kapcsolatok

Az utólag hozzáadott táblák javítása a továbbfejlesztés része.

## Modulok és kód ismertetése

### Adatbázis csatlakozás

Az adatbázis csatlakozást a db\_csat.php fájlunkban valósítottam meg, ahol megadjuk a szükséges informácókat (host, username, password) a csatlakozáshoz majd ezek álltal kapcsolódunk a hoston belül az adatbázishoz.

### Admin felület

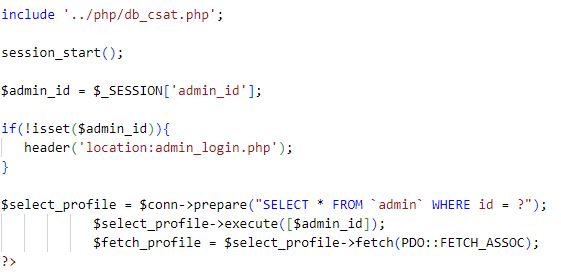
Úgy éreztük szükség van egy admin felületre is, hogy könnyebben tudjuk kezelni az oldalunkat és a felhasználóinkat.

Az admin felület egy saját css fájlt kapott így a stílus különbözik a fő oldalától.

Az admin felülethez tartozó fájljainkat az admin mappában találjuk egyből a törzskönyvtárból.

A *dashboard.php* adja meg a főképernyőjét az admin felületnek ahol különböző ablakokban érhetjük el a kívánt funkciókat.

Először is adatbázisból lekérdezzük a bejelentkezett admint és azt betöltjük a futó sessionbe.

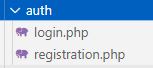


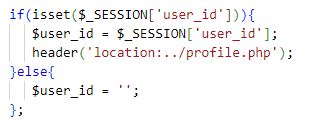
Ezután a főoldalon létrehoztuk az ablakokat és a megfelelő lekérdezéseket az adatbázisból a kívánt adatok kezeléséhez.

Először egy új php változóba bekérjük az adatokat az adatbázisunkból (pl $select\_users) majd egy új változóba betöltjük ezeket (pl $number\_of\_users) és egy rowCount() functionnel kiíratjuk az oldalunkra a meglévő adatok számát az oldalon.

### Authentication

Az *auth* mappában található a *login* illetve a *register* php fájl ami a egy új felhasználó létrehozásáért vagy egy már meglévőbe való beléptetésért felel.



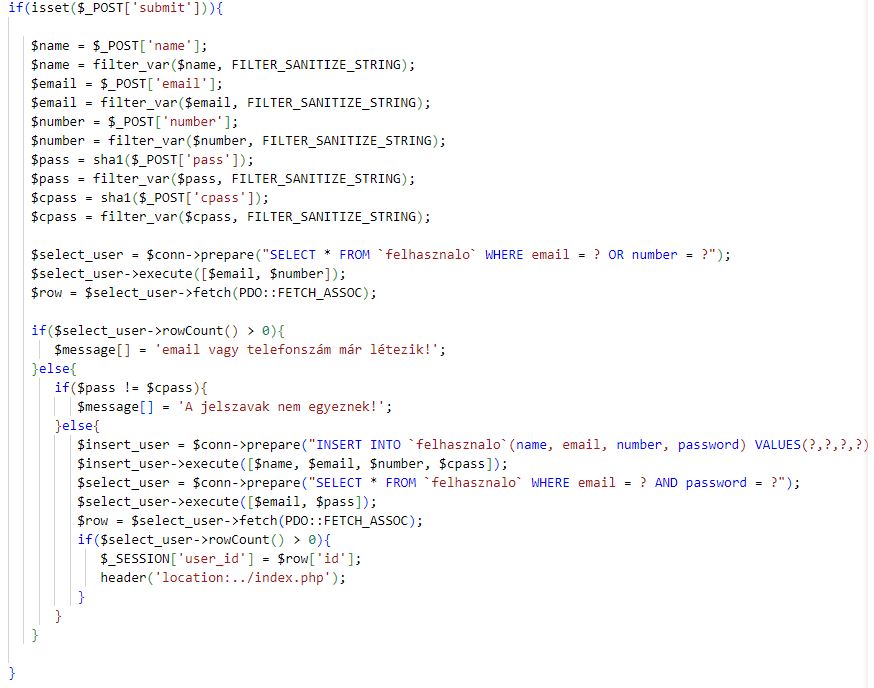


Először is ellenőrizzük a *login.php*-ban, hogy van-e a session-ünkben éppen aktív *user\_id,* ha van akkor az oldal átirányít a *profile.php* fájlba ahol a felhasználó eléri a megadott adatait és változtathatja a kiszállítási címét.



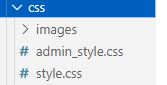
Miután a felhasználó kitöltötte a bejelentkezéshez megadott formot a *$\_POST* szuper globális változó lekéri az email címet és a jelszót majd a *filter\_var* function levédi az sql lekérdezést a külsö támadások ellen. Ezek után a *$select\_user*  változónkba betöltjük a lekért adatokat az adatbázisból, majd ha talált helyes párosítást akkor a felhasználó id-jét betölti a user\_id-be amit a session a továbbiakban használni fog és átirányít az *index.php* fájlra ami a webshopunk főoldalát tartalmazza, viszont ha nem talál ilyen párosítást küld egy üznetet a felhasználónak.

A *register.php* fájlban is megtörténik ugyan az az ellenőrzés ami a *login.php*-nál.



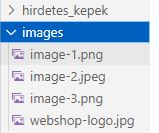
Szintúgy, ha a felhasználónk leadta a regisztrációs formot akkor a *$\_POST* változónk begyűjti a megadott adatokat. Két jelszó változót adtunk meg (*$pass* illetve *$cpass*) a dupla jelszó megadása végett. A *$select\_user* változó ellenőrzi, hogy új adatokat adnak-e meg, ha nem akkor hiba üzenetet kap a felhasználó. Abban az esetben, ha újak az adatok az ’INSERT INTO’ sql paranccsal betöltjük a felhasználó táblába az új adatokat és az akkor kapott id-t hozzá rendeljük egy új user\_id-hez ami a sessionbe futni fog és a felhasználó átkerül a webshop főoldalára.

### CSS / Cascading Style Sheets

Ebben a mappában találhatóak a felhasznált ikon képek és a főoldalon felhasznált carosuel képei illetve maga a css stílus fájlok amik az oldalak stílusáért felelősek.

### Képek

Van két mappánk *hirdetes\_kepek* illetve *images*. Az előbbi mappába kerülnek azok a képek amiket a felhasználók feltöltenek a hirdetéseikhez, hogy késöbb az oldal letudja kérni az egyes hirdetések megjelenítéséhez, az utóbbiba pedig a rendelést igazoló email fejléce és ikonjai találhatóak.



### HTML kód nélküli funkciók

A ’php’ nevezetű mappában találhatóak azok a php kódjaink amik nem tartalmaznak semmilyen más nyelvet.

#### Phpmailer

Két darab email küldő kódunk van amit phpmailer segítségével oldottunk meg.

Először is meghívjuk a phpmailer beépített funkcióit,

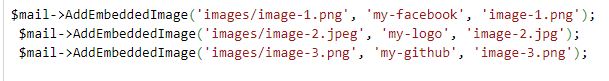


majd bekérjük a session-ünkből az éppen aktív felhasználót a user\_id-vel és lekérjük az id-hez rendelt email címet, nevet és telefonszámot illetve az üzenetet amit leírt a form-ba. Ezek után megadjuk az SMTP (Simple Mail Transfer Protocol) hostot ami a mi esetünkben a gmail beépített protokolja. Beállitjuk a bejelentkezési információkat (email cím illetve a generált alkalmazás jelszó) a biztonság érdekében, majd megadjuk a küldö illetve a befogadó email címet ami a segítségkérő üzenet esetében mindkét cím a webshopunk címe, majd a bekért adatokat betöltjük az email ’body’-jába és a felhasználót küldés után vissza irányítjuk a *segitseg.php* fájlra ami a kontakt oldal kódját tartalmazza.

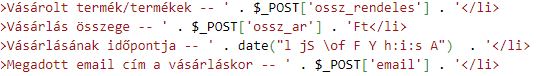


A rendelést igazoló emailt küldő kódunk eleje ugyan így épül fel azzal a különbséggel, hogy itt nem a webshopunk email címére kell küldenünk az emailt, hanem a felhasználónak aki éppen most rendelt akinek az email címét bekért *$email* változóval adunk meg.

Ezután mivel az emailünk tartalmaz képeket is a feljebb említett *images* mappából be kell kernünk őket hogy majd betudjuk illeszteni az emailünkbe.



Minden képnek adnunk kell egy ’cid’-t ami alapján majd az email HTML kódjában meghívhatjuk. A képen példaként a Facebook ikon raktam aminek a ’my-facebook’ ’cid’-t adtam.

Annak érdekében, hogy a kód automatikusan küldje ki ezt az emailt és mindig a megfelelő adatokat küldje ki meg kellett oldanunk, hogy magától kérje be az adatokat és küldje ki ezt a *$\_POST* változóval értük el az alább látható módon.

Az email maga pedig így kerül kiküldésre.



#### Hirdetés megnyitása

A főoldalunkon a hirdetések nem részletesen jelennek meg ezért szükségünk volt egy oldalra amit a főoldalról megnyitva ér el a felhasználó ahol elér minden fontos információt az adott termékről.



Megnyitott hirdetés

Hirdetés előnézet

Mindenek előtt létrehoztunk egy új adatbázis csatlakozást csak ebben az esetben a *mysqli* módszerrel a *PDO* helyett amit eddigiekben használtunk mert szükségünk volt a *mysqli\_query()* functionre, hogy a hir\_id által letudjuk kérdezi a megfelelő hirdetés adatait.



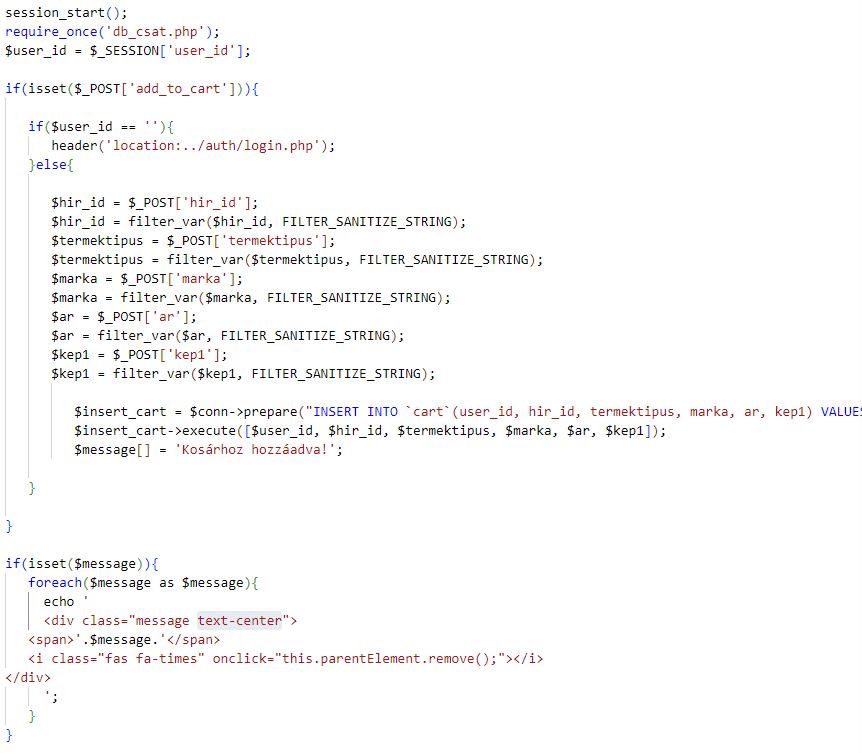
Az SQL lekérdezésben a különböző táblák egybe ágyazásait ez a kód teszi lehetővé.

"SELECT \* FROM hirdetesek INNER JOIN allapotok ON allapotok.allapotok\_id = hirdetesek.allapotok\_id INNER JOIN markak ON markak.markak\_id = hirdetesek.markak\_id INNER JOIN tipusok ON tipusok.tipusok\_id = hirdetesek.tipusok\_id INNER JOIN nemek ON nemek.nemek\_id = hirdetesek.nemek\_id INNER JOIN szinek ON szinek.szinek\_id = hirdetesek.szinek\_id INNER JOIN sportagak ON sportagak.sportagak\_id = hirdetesek.sportagak\_id WHERE id = $hir\_id";

Ezek után a HTML kódban a *$fetch\_product* metódussal iratjuk ki az oldalra a lekért adatokat ezen a módon.

#### Kosárba helyezés

Először is ismét ellenőrizzük, hogy van-e a sessionben ’user\_id’, ha nincs az oldal átírányít a *login.php*-ra, ha van akkor behelyezzük a ’cart’ nevű táblába azokat az adatokat amik a megfelelő ’hir\_id’-hez tartoznak ha a felhasználó lenyomta az ’add\_to\_cart’ nevű gombot az *index.php*-ban található formban, majd az *$insert\_cart* nevű változó lefutattja az ’INSERT INTO’ SQL parancsot és így a *bevasarlokocsi.php* letudja kérni belőle az adatokat, hogy bekerüljenek a kosárba.



#### Kijelentkezés

A *profil\_kijelentkezés.php* nevű fájlunk a *session\_unset()* nevű functionnel kitöröl minden éppen tárolt adatot a sessionből és leállítja, majd átirányítja a felhasználót a *login.php* nevű fájlra a *header()* functionnel.

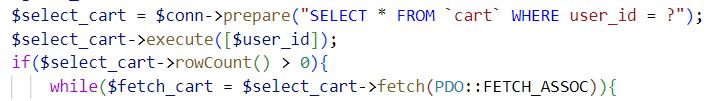


### HTML oldalak/funkciók

Ezeknél az oldalaknál felosztottuk a munkát. Balogh Ádám foglalkozott a HTML illetve a CSS kóddal, Schönberger Dominik pedig a php-val.

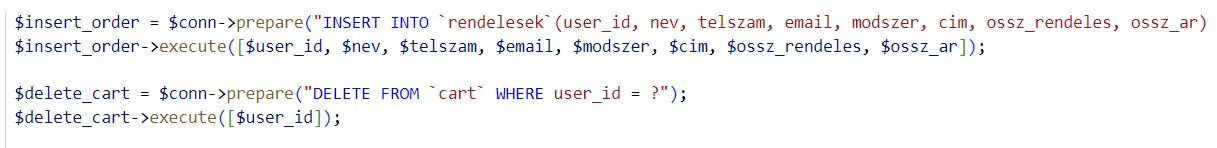
#### Bevásárlókocsi

A ’bevásárlókocsi’ mint funkció elengedhetetlen egy webshopnál így sokat foglalkoztam vele, hogy megfelelően működjön. Először is az első funkció amit létre akartam hozni az az, hogy ha a felhasználó több terméket is a kosárba helyezett, de nem feltétlen akar megválni az összestől ezt megtehesse, ezért két gomb található az oldalon ahol egyenként lehet terméket törölni vagy akár az egészet egyszerre. Ezt úgy értem el, hogy egy termék törlésénél a kosárba helyezésnél kapott id-re szűrtem rá míg a teljes kosárnál pedig az éppen aktív user\_id-re.

A kód belsejében csináltam még egy lekérdezést ami lehetővé teszi, hogy az oldalon megtudjuk jeleníteni a pontos adatait a kosár tartalmának.

#### Fizetés

Mindenek előtt bekérjük a *$\_POST*-al az adatokat, hogy feltudjuk tölteni a ’rendelesek’ táblánkba és lekérjük a megfelelő kosár tartalmat is a ’user\_id’-vel.

Kényelmi szempontból megakartam csinálni, hogy miután a rendelés leadásra került és bekerült a ’rendelesek’ táblába a kosár magától kiürüljön ehhez a már a *bevasarlokocsi.php*-ban latható módon ’user\_id’ szerint törlök mindent a kosárból az ’INSERT INTO’ SQL parancs után.

Később létrehoztam két változót amik a *$grand\_total* illetve a *$select\_cart.* Az előbbi a kosárban található termékek árának az összegét tárolja el, míg az utóbbi bekéri a ’cart’ táblából az adatokat amiket majd később a *$fetch\_cart* változóval kiiratunk az oldalra.



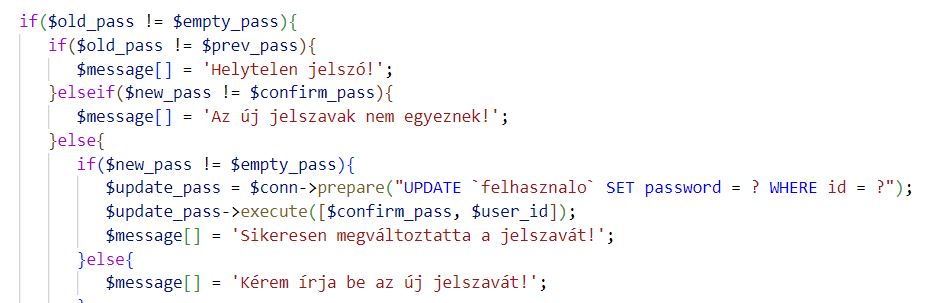
#### Profil műveletek

A felhasználói profil adatait és a kiszállítási címet lehet változtatni, ha esetleg más email címet kezdett volna el használni a felhasználó vagy ha éppen elköltözött.

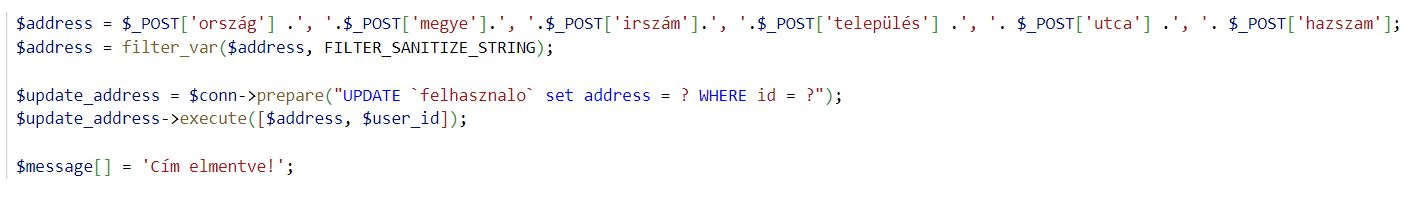
Először is ellenőrízzük, hogy az új adatok nem egyeznek-e már egy meglévővel a ’felhasznalo’ táblában, ha erre nem kerül sor akkor egy ’UPDATE’ paranccsal frissítjük a felhasználó adatait.

A jelszavak változtatása egy kicsit komplikáltabb folyamat. Először bekérjük a jelszavakt a formbol majd ellenőrizzük, hogy a felhasználó helyesen írta-e be a jelszavát mind a két alkalommal, hogy csak utána lehessen jelszót változtatni.

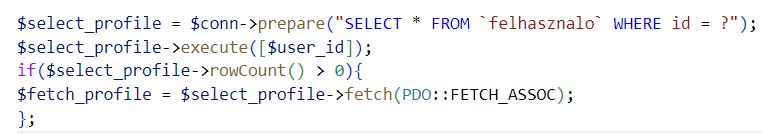




A kiszállítási cím esetén először *$\_POST*-al bekérjük az adatokat, majd egy ’UPDATE’ paranccsal megváltoztatjuk azt a ’felhasznalo’ táblánkban.



#### Profil megjelenítés

Mindenekelőtt csináltam egy lekérdezést az adatbázisból ami a *$fetch\_profile* változóba tölti az éppen bejelentkezett felhasználót.

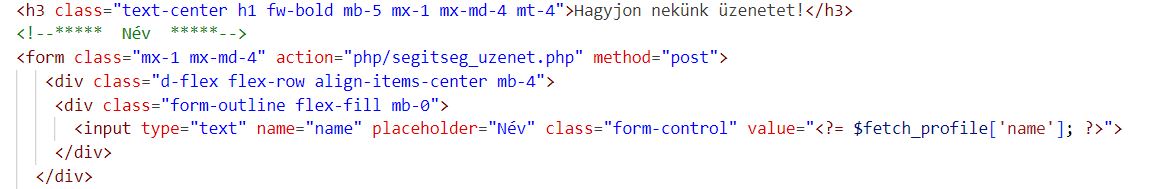
Majd ezt a kódot felhasználva kiiratjuk a felhasználó adatokat.



Ez a kód részlet így látható a frontenden.

#### Kontakt

A kontakt oldal a *segitseg.php* fájlban található amibe tettem egy formot ahova a felhasználó leírhatja a panaszát vagy a visszajelzését és a *segitseg\_uzenet.php*-t meghívva emailt küld a webshop email címére, hogy tudjunk rá reagálni.



## Tesztelés/Tesztesetek[[11]](#footnote-11)

A tesztelés rendkívül fontos egy program készítése közben, hiszen minden munkában fordulhatnak elő hibák. Ez a programozás esetében még inkább elmondható hiszen ebben az esetben akár egy hiányzó pontos vessző is okozhat nagy hibákat. A mi programunkon vizuális tesztet alkalmaztunk ami annyit jelent, hogy amint lekódoltunk egy adott funkciót vagy oldalt végig nyomkodtuk, hogy biztosak legyünk benne, hogy megfelelően működik mielőtt tovább haladnánk.

**Tesztelési alapelvek:**

1. A tesztelés hibák jelenlétét jelzi: A tesztelés képes felfedni a hibákat, de azt nem, hogy nincs hiba. Ugyanakkor a szoftver minőségét és megbízhatóságát növeli.
2. Nem lehetséges kimerítő teszt: Minden bemeneti kombinációt nem lehet letesztelni (csak egy 10 hosszú karakterláncnak 256^10 lehetséges értéke van) és nem is érdemes. Általában csak a magas kockázatú és magas prioritású részeket teszteljük.
3. Korai teszt: Érdemes a tesztelést az életciklus minél korábbi szakaszában elkezdeni, mert minél hamar találunk meg egy hibát (mondjuk a specifikációban), annál olcsóbb javítani. Ez azt is jelenti, hogy nemcsak programot, hanem dokumentumokat is lehet tesztelni.
4. Hibák csoportosulása: A tesztelésre csak véges időnk van, ezért a tesztelést azokra a modulokra kell koncentrálni, ahol a hibák a legvalószínűbbek, illetve azokra a bemenetekre kell tesztelnünk, amelyre valószínűleg hibás a szoftver (pl. szélsőértékek).
5. A féregirtó paradoxon: Ha az újratesztelés során (lásd később a regressziós tesztet) mindig ugyanazokat a teszteseteket futtatjuk, akkor egy idő után ezek már nem találnak több hibát (mintha a férgek alkalmazkodnának a teszthez). Ezért a tesztjeinket néha bővíteni kell.
6. A tesztelés függ a körülményektől: Másképp tesztelünk egy atomerőműnek szánt programot és egy beadandót. Másképp tesztelünk, ha a tesztre 10 napunk vagy csak egy éjszakánk van.
7. A hibátlan rendszer téveszméje: Hiába javítjuk ki a hibákat a szoftverben, azzal nem lesz elégedett a megrendelő, ha nem felel meg az igényeinek. Azaz használhatatlan szoftvert nem érdemes tesztelni.

***Tesztelési technikák***

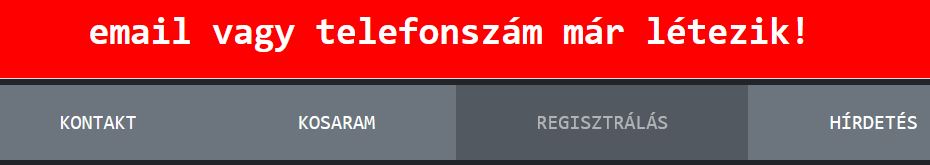
1. Feketedobozos (black-box) vagy specifikáció alapú, amikor a specifikáció alapján készülnek a tesztesetek.
2. Fehérdobozos (white-box) vagy strukturális teszt, amikor a forráskód alapján készülnek a tesztesetek.

Ezen csoportosítások az információn alapulnak mi szerint a tesztet lefutatják.

### Tesztesetek

Egy webshop esetében leginkább a profil műveleteknél kaphat a felhasználó hiba üzenetet, így ezt teszteltem leginkább.

Itt látható, ha a felhasználó helytelen adatokat ad meg a bejelentkezés illetve regisztráció során



Emellett leellenőriztem, hogy a profil módosításnál is működik-e a kód.



## Továbbfejlesztési lehetőségek

* Amivel mindenképp foglalkozni szeretnénk, hogy REST API-t használjunk a procedurális felépítés helyett, de mivel ketten maradtunk sajnos erre nem maradt kapacitásunk.
* A webshop hostolása, hogy ne csak localhoston keresztül lehessen elérni.
* A session helyett sütik alkalmazása.
* Befejezni a szűrést a főoldalon.
* Logo készítése.
* Saját hirdetéseim megtekintése, módosítása, törlése.
* Adatbázis tökéletesítése (pl.: kapcsolatok).

# **Felhasználói dokumentáció**

## Alkalmazás rövid ismertetése

Webes alkalmazásunk egy webáruház, melynek célközönsége a sportolók. Oldalunkon sportoló, illetve sportolni vágyó emberek tudnak kedvükre böngészni a különböző sporteszközök között. De nem csak böngészésre, vásárlásra alkalmas a webáruház, hanem saját termékeinket is eladásra tudjuk kínálni. Az elérhető hirdetések megtekintésén kívül, az oldal többi funkciója regisztrációhoz, bejelentkezéshez kötött. Vásárlóink kosárba tudják helyezni a megvásárolni kívánt termékeket, majd a fizetés gombra kattintás után, a fizetési adatok megadásával le tudják adni a rendeléseiket. A sikeres rendelés után mi gondoskodunk az eladó és a vevő kapcsolattartásáról, valamint a termékek kiszállításának teljes koordinációjáról.

## Rendszerkövetelmények

Szerencsére egy webshop futtatásához nincs szükségünk NASA szintű szuper perifériához. Mivel a webshopunk reszponzív, működéséhez tökéletesen elég egy olyan számítógép, laptop, tablet vagy telefon amin van internet hozzáférés és található rajta böngésző. Mi a Mozilla Firefoxot vagy a Chrome-ot ajánljuk a webshop használatához.

## Alkalmazás telepítése, szükséges beállítások

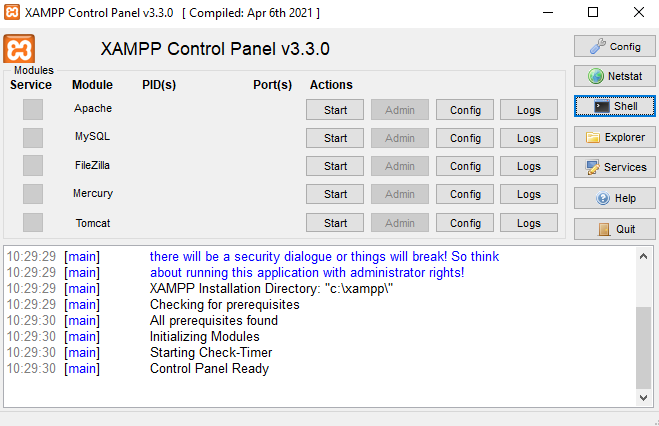
Webáruházunk egyelőre csak lokális szerveren érhető el, ezért a következő lépések elengedhetetlenek a webáruház eléréséhez

### XAMPP telepítése

Válasszuk ki a linkre kattintva az operációs rendszernek megfelelőt, majd kattintson a „Letöltés (64bit)” gombra és várjuk meg a letöltés befejezését. A ,,C” meghajtóra mentsünk. (<https://www.apachefriends.org/hu/download.html>)  
Ezután kattintsunk az exe fájlra, és engedélyezzük, hogy módosításokat hajtson végre a számítógépünkön. Innentől már csak a tovább gombokra kell kattintani.

### XAMPP elindítása

Indítsuk el az Apache és MySQL szervereket a Start gombokra kattintva.



Az elindítás sikeres, hogyha az Apache és MySQL feliratok háttere zöldre változik, illetve a gombok felirata Stop-ra módosul.



### A fájlok bemásolása a htdocs mappába

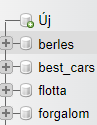
Helyezzük az adathordozóról vagy a GitHub-ról a HasznaltSportszer nevű mappát a **XAMPP** **htdocs** mappájába.

### Adatbázis futtatása

Nyissuk meg a XAMPP-ot, majd kattintsunk a MySQL admin gombjára.



Ekkor az adatbázis kezelőben találhatjuk magunkat (localhost/phpmyadmin), majd az oldal bal oldalán kattintsunk az *új* adatbázis lehetőségre.



Ezután 2 lehetőségünk van:

1. Felül kiválasztjuk az importálás menüpontot, majd a fájl kiválasztásánál kiválasztjuk a *sportszer\_ab.sql* fájlt, végül az importálás gombra kattintunk.
2. Kimásoljuk *sportszer\_ab.sql* fájl teljes tartalmát, majd felül kiválasztjuk az SQL menüpontot és beillesztjük a kimásolt szöveget, és az indítás gombra kattintunk.

### Webáruház megnyitása

Ha mindennel megvagyunk, akkor a böngésző keresőjébe írjuk be a következőt: *localhost:8080/HasznaltSportszer.* Ekkor az alkalmazás főoldalán találhatjuk magunkat.

## Alkalmazás használata

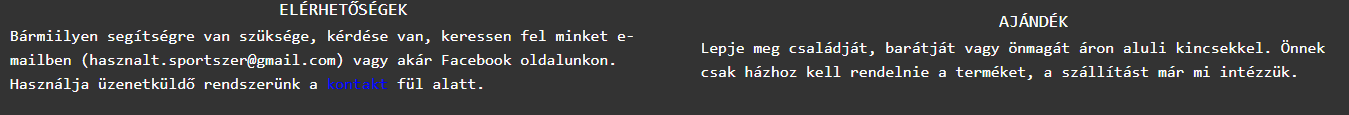
### Navigáció, lábjegyzet

#### Navigációs sáv



Az összes oldal tetején található egy navigációs rész az oldal könnyed használhatóságának érdekében. A sávban az éppen aktív oldal gombjának a háttérszíne sötétebb a többitől, illetve nem is tudunk rákattintani. Továbbá a bal oldali *VISSZA* gomb megnyomásával a legutóbb látogatott oldalra tudunk ugrani.

#### Lábjegyzet

Minden oldal alján található egy szürke hátterű lábjegyzet rész, amely a webáruház fejlesztőinek az elérési módját, és az ajándék vásárlásra ösztönzést foglalja magába.

### Regisztráció, Bejelentkezés

Webáruhazunkban, hogyha a felhasználó még nem regisztrált vagy csak nincs még belépve, akkor a bejelentkezési és regisztrációs oldalakon kívül csak a főoldalt tudja megnyitni, ha más oldalra szeretne lépni, akkor mindig a beléptetési oldalra fogja dobni a felhasználót az oldal.

#### Regisztráció

Oldalunk teljeskörű használatához regisztráció szükséges. Ezt pár perc alatt kivitelezhetjük is, nincs más teendőnk, mint néhány adatot megadni.



A rendszerbe egy e-mail címmel, illetve egy telefonszámmal csak egyszer lehet regisztrálni. A továbbfejlesztés része egy megerősítő e-mail vagy SMS küldése a regisztráció jóváhagyásához.

#### Bejelentkezés

Ha már regisztráltunk, akkor a belépéshez nincs más dolgunk, mint a belépési oldalon megadni a regisztrációkor használt e-mail címet illetve az ehhez tartozó jelszót.

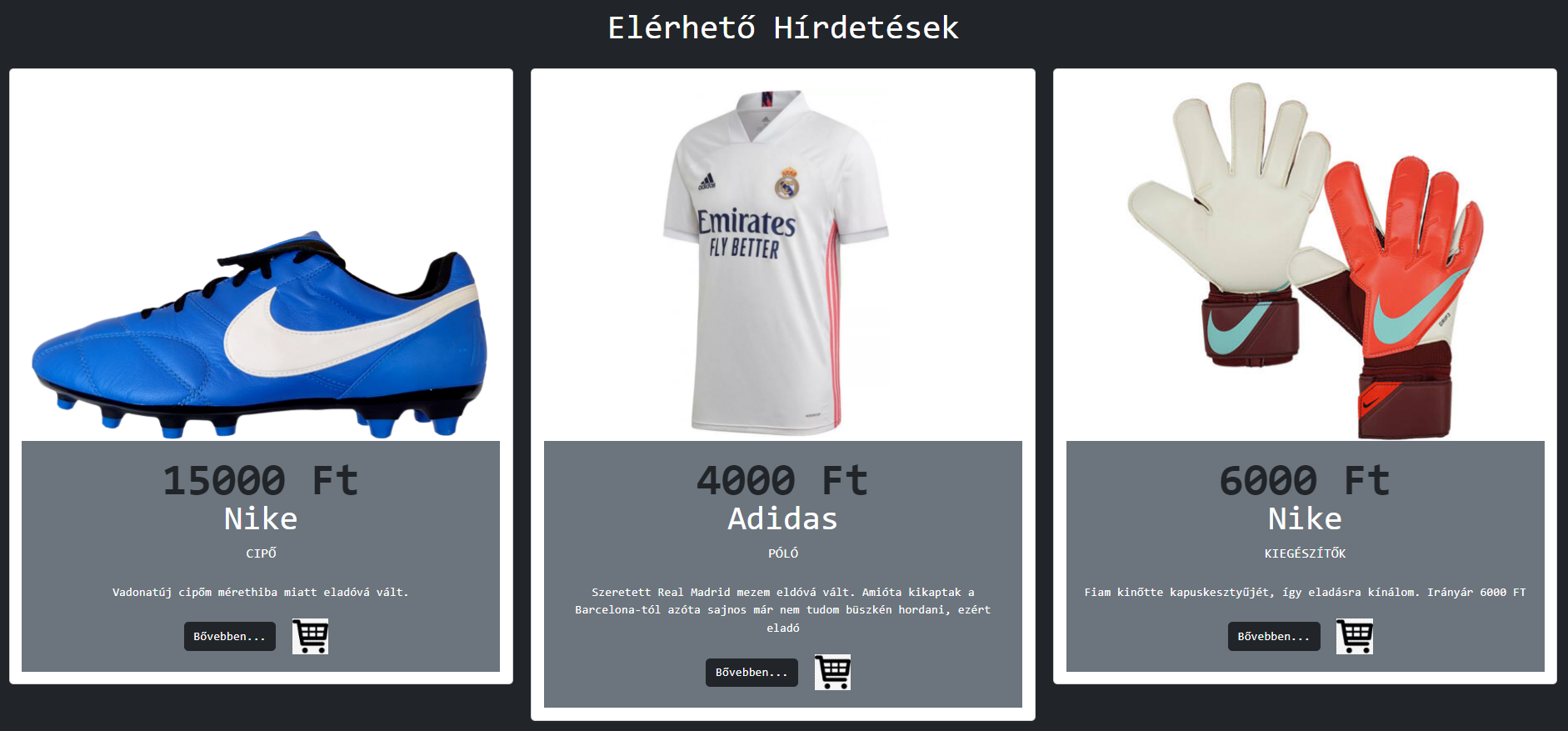
### Főoldal

Főoldalunkon a navigációs rész alatt található webáruházunk neve. A továbbfejlesztés része lesz, hogy ezt a feliratot egy saját embléma, logó váltsa majd. Ezt követően 3 lapozható kép található, melyek önmaguktól is lapozzák magukat. A rajtuk található adatok természetesen csak kitaláltak, mivel oldalunk még nem üzemel. Ezen képek célja a reklám, illetve, hogy pozitív hatást keltsen a felhasználókban, hogy ezen az oldalon igenis érdemes eladni a termékeket.



Ez alatt a szűrési szekció található. Ez a funkció egyelőre még nem működőképes. Megvalósítása a csoportunk 3. tagja (Hajszter Botond) feladata lett volna és a pótlásra sem maradt már idő ****

Végül az elérhető hirdetések találhatóak.



Itt láthatjuk a termék árát, márkáját, típusát és leírását. Ez alatt található 2 gomb, a *Bővebben* gombra kattintva láthatjuk a további képeket a hirdetésről (lapozható képek), illetve a termék részletes adatait.



A *kosár*gombra kattintva a kosárba tudjuk helyezni a kiválasztott terméket.

### Kosaram

A kosaram oldalon láthatjuk a kosárba helyezett termékünk főbb adatait. 

A *Termék törlése* gombra kattintva csak az adott termék kerül törlésre a kosárból.   
A *Kosár tartalmának törlése* gomb lenyomásával az egész kosarat kiürítjük.   
A *Folytatom a böngészés*t gombbal a főoldalra tudunk visszamenni, hogy folytassuk a nézelődést.  
*Tovább a fizetéshez* gomra kattintva egy új oldalra lépünk.:

### Rendelés leadása



Ezen az oldalon először is láthatjuk termékenként a márkát, a fajtát és az árat, illetve az összesített (fizetendő) árat. Alatta található gombbal módosíthatjuk a kiválasztott termékeinket. Ezután a személyes adatainkat láthatjuk, amit még itt is van módunk módosítani. Végül a kiszállítási cím és a fizetési mód megadása után leadhatjuk rendelésünk. A rendelés sikeres leadásáról visszaigazoló e-mailt kapunk a webáruháztól és a következő szöveg jelenik meg az oldal fejlécében.:

A vásárlás folytatása gombra kattintva a főoldalra lépünk vissza.

### Kontakt

A kontakt oldal célja, hogy az ügyfeleinket segíteni tudjuk bármilyen kérdés vagy panasz esetén. Az felhasználó e-mail címéről kapunk egy levelet, amely tartalmazza a személyes adatait, illetve az üzenetét. Ezután felvesszük a kapcsolatot az ügyféllel a megadott e-mail címen vagy telefonszámon.

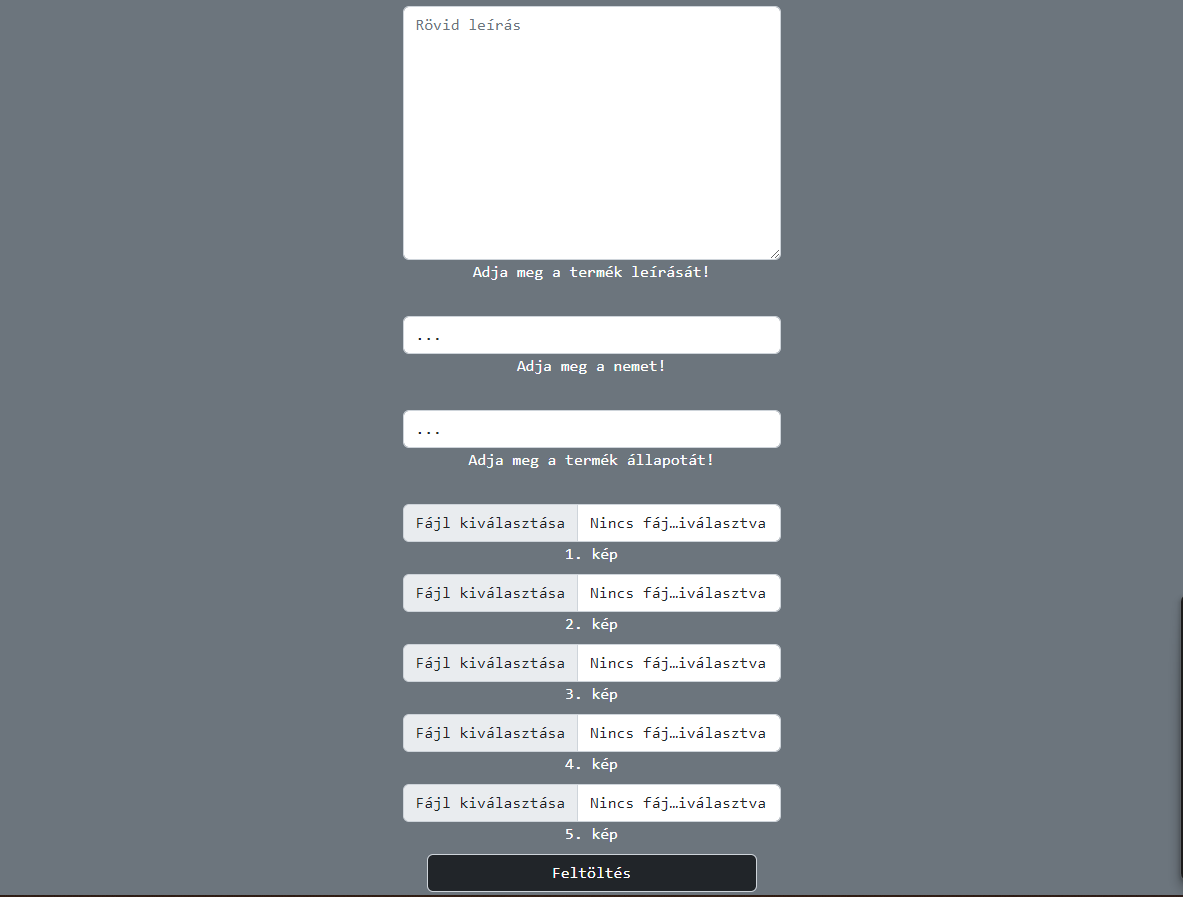


A rendszer a személyes adatokat automatikusan kitölti, de ha az üzenet írója meg szeretne változtatni bármiféle adatot az ügyintézéshez, akkor természetesen megteheti. Ehhez csak egyszerűen kitörli a már beírt adatokat és beírja az általa használni kívánt adatokat.

### Hirdetés feltöltése

A hirdetés feltöltéséhez az alább látható adatok megadása mind kötelező, képből elég csak egyet feltölteni, de ajánlott a több kép használata. Ahol legördülő menü található, ott az adatokat az adatbázisunkból hívja meg az oldal.   
(márka, sportág, szín, típus, nem, állapot)

A sikeres feltöltés után hirdetésünk megjelenik a főoldalon.

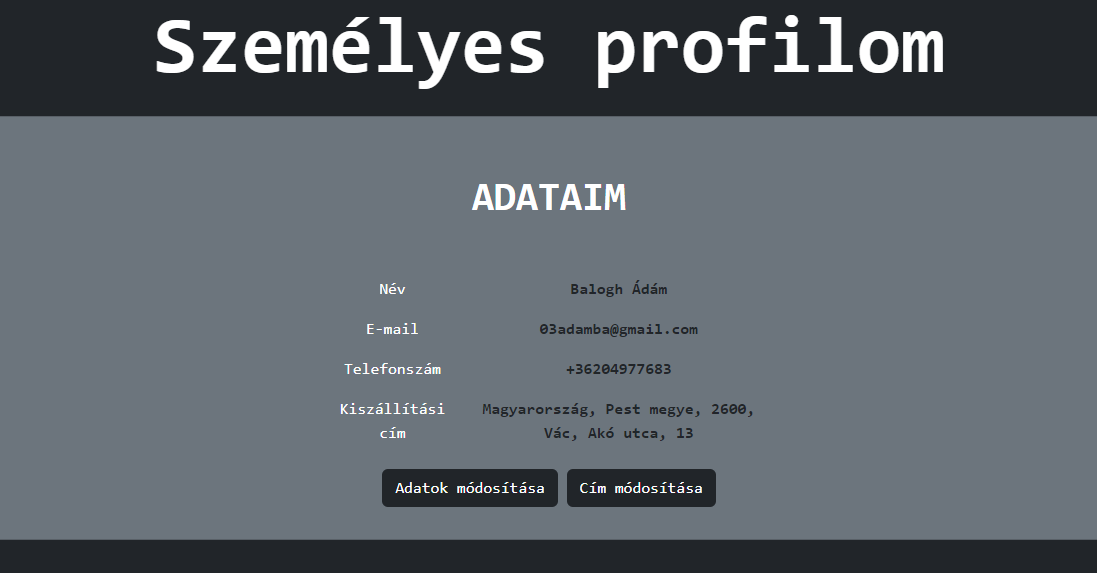


### Profilom

A profilom legördülő gombra kattintva az a bejelentkezésen kívül a következő lehetőségek közül választhatunk még.:

#### Profilom megtekintése

A profilom megtekintése fül alatt személyes adataink illetve a kiszállítási címünk található.



A megfelelő gombra kattintva lehetőségünk van megváltoztatni személyes adatainkat, jelszavunkat és a címünket.

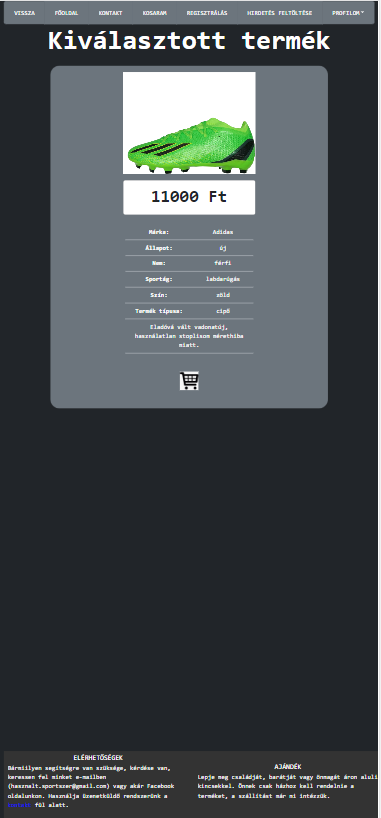
#### Kijelentkezés

A kijelentkezés gombra kattintva kilépünk profilunkból.

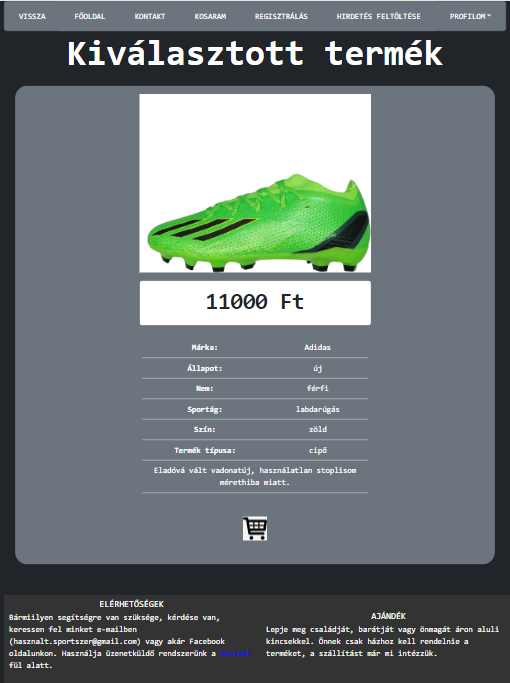
## Felhasználható eszközök

A webáruház teljes mértékben reszponzív, tehát akár telefonon, akár tableten is jól használható, melyet a Bootstrap keretrendszerrel valósítottunk meg. Az alábbi képeken egy megnyitott hirdetés oldalát láthatjuk telefon és tablet nézetbe.

**iPhone 12 Pro**



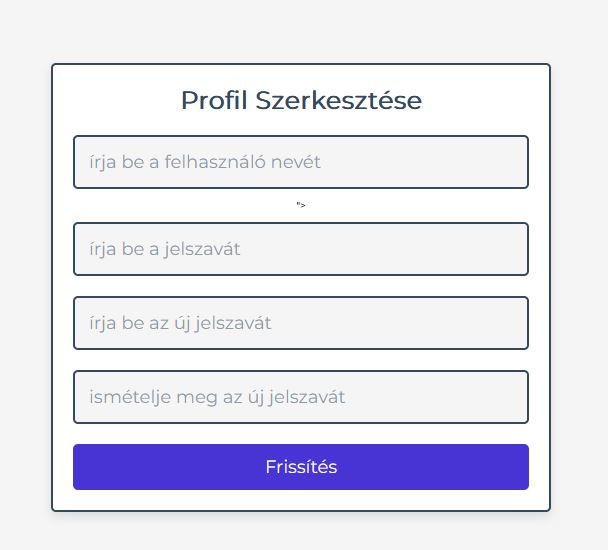
**iPad Pro**



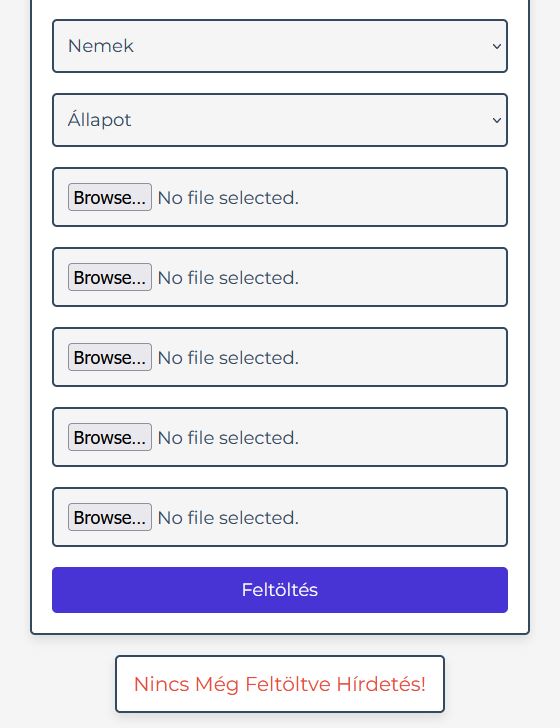
## Admin felület

Ahogy azt már a fejlesztői dokumentációban említettem az admin felület kinézete különbözik a főoldaláétól. A kezdőképernyőn ablakok találhatóak ahol a kívánt feladatainkat tudjuk ellátni.

### Profil szerkesztés

Ezen a fülön az éppen bejelentkezett adminnak változtathatjuk a jelszavát.

### Hirdetések

A hirdetéseknél adminként tölthetünk fel hirdetést vagy törölhetjük a már meglévőket, ha esetleg a bizonyos termék már elkelt.

### Felhasználói fiókok

Ezen a fülön tudjuk kezelni a már regisztrált felhasználóinkat.



### Adminok kezelése

Ezen a fülön tudunk törölni illetve regisztrálni új adminokat, illetve itt is tudjuk frissíteni a már bejelentkezett fiókot.

# **Irodalomjegyzék**

## Internetes források

1. Stack overflow 🡪 <https://stackoverflow.com/>
2. W3Schools 🡪 <https://www.w3schools.com/>
3. PHP Tutorial 🡪 <https://www.phptutorial.net/>
4. GitHub 🡪 <https://github.com/>
5. Code Forum 🡪 <https://codeforum.org/>
6. Geeks for Geeks 🡪 <https://www.geeksforgeeks.org>
7. Javatpoint 🡪 <https://www.javatpoint.com>

## Könyvek

***How To Design Programs*** – Robert Bruce Findler, Matthias Felleisen, Shriram Krishnamurthi, Matthew Flatt

***Web Design with HTML, Css, Javacsript and Jquery Set*** – Jon Duckett

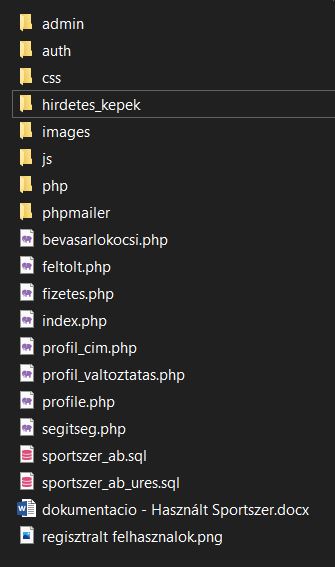
***Tiszta Kód*** – Robert C. Martin

# **Mellékletek**

Alkalmazásunk fontosabb kódjainak elérése githubon:

<https://github.com/SchonbergerDominikBoronkay/HasznaltSportszer>

Illetve a teljes állomány elérése GoogleDrive-on:

https://drive.google.com/drive/u/1/folders/1CoPtOKLU9ZpCrW2RnX8SHeFxAR5LC4HA

Az alkalmazásunk mappa szerkezete itt láttható.

1. Az *admin* mappában találhatóak az admin felülethez szükséges kódjaink,
2. Az *auth* mappában a regisztrációhoz és a bejelentkezéshez szükséges kódok,
3. A *css* mappában a designért felelős fájlok,
4. A *hirdetes\_kepek* illetve az *images* mappában az oldal működéséhez szükséges képek,
5. A *js* mappában a javascript kódok,
6. A *php* mappában pedig azon kódjaink amik nem tartalmaznak mást csak php funkciókat.

Illetve itt található meg a két adatbázis (üres és a teszt adatokkal feltöltött), a dokumentáció és a teszteléshez szükséges bejelentkezési adatok is.

1. https://hu.wikipedia.org/wiki/Visual\_Studio\_Code [↑](#footnote-ref-1)
2. https://hu.wikipedia.org/wiki/Brackets [↑](#footnote-ref-2)
3. https://www.inf.u-szeged.hu/~gnemeth/adatbgyak/exe/MySQL\_XAMPP\_JDBC/a\_php\_nyelvi\_elemeinek\_rvid\_ttekintse.html#Mi\_a\_PHP.3F [↑](#footnote-ref-3)
4. https://hu.wikipedia.org/wiki/HTML [↑](#footnote-ref-4)
5. https://prog.hu/cikkek/100430/bevezetes-a-css-alapjaiba [↑](#footnote-ref-5)
6. http://nyelvek.inf.elte.hu/leirasok/JavaScript/index.php?chapter=1 [↑](#footnote-ref-6)
7. https://en.wikipedia.org/wiki/PHPMailer [↑](#footnote-ref-7)
8. https://www.oracle.com/database/what-is-json/ [↑](#footnote-ref-8)
9. https://www.inf.u-szeged.hu/~gnemeth/adatbgyak/exe/MySQL\_XAMPP\_JDBC/a\_xampp\_keretrendszer\_s\_hasznlata.html [↑](#footnote-ref-9)
10. https://www.w3schools.com/bootstrap5/ [↑](#footnote-ref-10)
11. https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=0CAMQw7AJahcKEwjAroCtx4P-AhUAAAAAHQAAAAAQAw&url=https%3A%2F%2Faries.ektf.hu%2F~gkusper%2FSzoftverTeszteles.pdf&psig=AOvVaw00b\_KIZtBsVBJpon2ZoHg4&ust=1680262397248436 [↑](#footnote-ref-11)