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

**Vizsgaremek**

**Dallos Ruben, Herédi Gábor**

**2024.**

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



**Vizsgaremek**

TornadoPC

|  |  |
| --- | --- |
| Konzulens: Wiezl Csaba | Készítette: Dallos Ruben  Herédi Gábor |

# Hallgatói nyilatkozat

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

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Dallos Ruben |  | Herédi Gábor |  |

# Konzultációs lap

Vizsgázók neve: Dallos Ruben, Herédi Gábor

Vizsgaremek címe: TornadoPC

Program nyújtotta szolgáltatások:

* Regisztráció - bejelentkezés
* Asztali adminisztrátori felület
* Konfiguráció összeállítás és megosztás
* Kedvencek - Kosár
* Rendelés

|  |  |  |
| --- | --- | --- |
| Sorszám | A konzultáció időpontja | A konzulens aláírása |
| 1. | 2023.10.17. |  |
| 2. | 2023.11.21. |  |
| 3. | 2023.12.13. |  |
| 4. | 2024.01.16 |  |
| 5. | 2024.02.20. |  |
| 6. | 2024.03.12 |  |

|  |  |  |
| --- | --- | --- |
| A vizsgaremek beadható: |  | A vizsgaremeket átvettem: |
| Vác, 2024. |  | Vác, 2024. |
| Konzulens |  | A szakképzést folytató intézmény felelőse |

**Tartalomjegyzék**

[Hallgatói nyilatkozat 3](#_Toc165210483)

[Konzultációs lap 4](#_Toc165210484)

[Tartalomjegyzék 5](#_Toc165210485)

[Témaválasztás 9](#_Toc165210486)

[1 Fejlesztői dokumentáció 10](#_Toc165210487)

[1.1 Munkamegosztás 10](#_Toc165210488)

[1.2 Vizsgaremek bemutatása 11](#_Toc165210489)

[1.3 Fejlesztői környezet 11](#_Toc165210490)

[1.3.1 Visual Studio Code 11](#_Toc165210491)

[1.3.2 Visual Studio 2022 11](#_Toc165210492)

[1.3.3 XAMPP 11](#_Toc165210493)

[1.3.4 Teams 12](#_Toc165210494)

[1.3.5 Github 12](#_Toc165210495)

[1.4 Felhasznált technológiák 13](#_Toc165210496)

[1.4.1 HTML 13](#_Toc165210497)

[1.4.2 CSS 13](#_Toc165210498)

[1.4.3 PHP 13](#_Toc165210499)

[1.4.4 JavaScript 14](#_Toc165210500)

[1.4.5 jQuery 14](#_Toc165210501)

[1.4.6 AJAX 14](#_Toc165210502)

[1.4.7 REST API 15](#_Toc165210503)

[1.4.8 MySQL 15](#_Toc165210504)

[1.4.9 C# 16](#_Toc165210505)

[1.5 Használati esetmodell 16](#_Toc165210506)

[1.6 A program felépítése 17](#_Toc165210507)

[1.7 Adatbázis 18](#_Toc165210508)

[1.7.1 akciok tábla 20](#_Toc165210509)

[1.7.2 arkategoriak tábla 21](#_Toc165210510)

[1.7.3 ertekeles tábla 22](#_Toc165210511)

[1.7.4 felhasznalok tábla 23](#_Toc165210512)

[1.7.5 fizetesi\_modok tábla 24](#_Toc165210513)

[1.7.6 jellemzok tábla 25](#_Toc165210514)

[1.7.7 kedvencek tábla 26](#_Toc165210515)

[1.7.8 kepek tábla 27](#_Toc165210516)

[1.7.9 konfig tábla 28](#_Toc165210517)

[1.7.10 konfig\_id tábla 29](#_Toc165210518)

[1.7.11 kosar tábla 30](#_Toc165210519)

[1.7.12 markak tábla 31](#_Toc165210520)

[1.7.13 rendeles tábla 32](#_Toc165210521)

[1.7.14 termekek tábla 33](#_Toc165210522)

[1.7.15 termekek\_akciok 35](#_Toc165210523)

[1.7.16 termek\_kategoriak tabla 36](#_Toc165210524)

[1.7.17 tetelek tábla 37](#_Toc165210525)

[1.8 Rendszerkövetelmények 38](#_Toc165210526)

[1.9 Megfelelő indítás 39](#_Toc165210527)

[1.10 A weboldal felépítése 39](#_Toc165210528)

[1.10.1 Főoldal 40](#_Toc165210529)

[1.11 Termékek oldal 42](#_Toc165210530)

[1.11.1 termekekMegjelenit() 42](#_Toc165210531)

[1.11.2 Megjelenítés 45](#_Toc165210532)

[1.12 Kosár 46](#_Toc165210533)

[1.13 Kedvencek 47](#_Toc165210534)

[1.14 Belépés 48](#_Toc165210535)

[1.15 Regisztráció 49](#_Toc165210536)

[1.16 Konfiguráció összerakó 50](#_Toc165210537)

[1.17 A konfiguráció megtekintő, és értékelő oldal 51](#_Toc165210538)

[1.17.1 Card elemek 51](#_Toc165210539)

[1.17.2 Ajax hívások 53](#_Toc165210540)

[1.17.3 rating.php 54](#_Toc165210541)

[1.17.4 limit.php 55](#_Toc165210542)

[1.18 Tesztelés, tesztesetek 56](#_Toc165210543)

[1.18.1 Unit tesztek: 56](#_Toc165210544)

[1.18.2 Integrációs tesztek: 56](#_Toc165210545)

[1.18.3 Funkcionalitási tesztek: 56](#_Toc165210546)

[1.18.4 Megjelenés, felhasználói élmény tesztelése: 57](#_Toc165210547)

[1.18.5 Regisztráció, bejelentkezés 57](#_Toc165210548)

[1.18.6 Rendelés 59](#_Toc165210549)

[1.19 Továbbfejlesztési lehetőségek 60](#_Toc165210550)

[1.19.1 Felhasználói fiókok módosítása 60](#_Toc165210551)

[1.19.2 E-mail 60](#_Toc165210552)

[1.19.3 Ügyfélszolgálat 60](#_Toc165210553)

[1.19.4 Rendelés visszaküldése 60](#_Toc165210554)

[1.19.5 Akciók, leértékelések, kuponkód 61](#_Toc165210555)

[1.19.6 Fizetési módok 61](#_Toc165210556)

[1.19.7 Kompatibilitás és további elemek hozzáadása 61](#_Toc165210557)

[1.19.8 Konfiguráció megosztó oldal továbbfejlesztése 61](#_Toc165210558)

[2 Felhasználói dokumentáció 62](#_Toc165210559)

[2.1 Bevezetés 62](#_Toc165210560)

[2.2 Rendszerkövetelmények 62](#_Toc165210561)

[2.3 Főoldal 63](#_Toc165210562)

[2.4 Footer 63](#_Toc165210563)

[2.5 Navigációs sáv 64](#_Toc165210564)

[2.5.1 Termékek, Kategóriák 64](#_Toc165210565)

[2.5.2 Regisztráció, Bejelentkezés 64](#_Toc165210566)

[2.5.3 Konfigurációs rész 65](#_Toc165210567)

[2.5.4 Kosár, kedvencek 65](#_Toc165210568)

[2.6 Egy termék oldala 65](#_Toc165210569)

[2.7 Kosár 66](#_Toc165210570)

[2.8 Rendelés 66](#_Toc165210571)

[2.9 Rendelési előzmények 67](#_Toc165210572)

[2.10 Konfiguráció összeállító 68](#_Toc165210573)

[2.11 Konfiguráció értékelő 68](#_Toc165210574)

[3 Irodalomjegyzék 70](#_Toc165210575)

[3.1 Internetes tartalmak 70](#_Toc165210576)

[3.2 Könyvek 70](#_Toc165210577)

[4 Melléklet 71](#_Toc165210578)

# Témaválasztás

A vizsgaremekünk témájaként egy számítógép alkatrészeket, perifériákat, kiegészítőket és egyéb elektronikai eszközöket árusító webshopot választottunk. A tervezés első pillanataitól webes alkalmazásban gondolkodtunk. Rögtön egy számítógépes webshop volt az ötlete, ezért gyorsan neki tudtunk látni a munkálatoknak.

Nem akartuk, hogy a mi projektünk egy legyen a többezer webshop között, ami már elkészült, ezért találtuk ki a számítógép konfiguráció építő, és megosztó részt is, hogy valamivel egyedivé és maradandóvá tegyük. A megosztott konfigurációkat mások láthatják és tudnak rá reagálni like, dislike formájában, szóval egy kezdetleges közösségi média oldalra hasonlítana. A megosztott konfigurációk között lehetne nézelődni, inspirációt szerezni és ezek alapján vásárolni is.

A weboldalt webshop részét Ruben készítette el. Itt megtalálhatóak kategóriák, amire kattintva a megfelelő termékeket láthatjuk. A termékekre rá tudunk keresni. Tudunk kedvencek közé és kosárba helyezni, továbbá rendelni és rendelési előzményeket megtekinteni.

Gábor feladata volt az adatbázis összeállítása, konfiguráció összeállítására és megosztására szolgáló oldalak elkészítése, regisztrációs és bejelentkező oldalak, valamint az adminisztrációs asztali alkalmazás létrehozása is.

# Fejlesztői dokumentáció

## Munkamegosztás

**Dallos Ruben**

* navigációs sáv
* kategóriák oldal
* termékek oldal
* egy termék oldala
* rendezés
* keresés
* kedvencek
* kosár
* rendelés

**Herédi Gábor**

* regisztráció oldal
* bejelentkezés oldal
* konfiguráció összeállító oldal
* konfiguráció megosztó oldal
* adminisztrátori felület (C#)

## Vizsgaremek bemutatása

Vizsgaremekünket OOP (Objektumorientált Programozás) struktúra, és REST architektúra segítségével készítettük el. A fő szempont egy egyszerűen használható webshop és egy alap közösségi oldal. Törekedtünk a felhasználóbarát és modern megjelenésre.

## Fejlesztői környezet

### Visual Studio Code

|  |  |
| --- | --- |
| A weboldal forráskódját a Visual Studio Code IDE-ben írtuk, mely a Microsoft egy ingyenes, nyílt forráskódú IDE felülete. Alapértelmezetten támogatja az alábbiak szerkesztését: HTML, JavaScript. [[1]](#footnote-1) | https://upload.wikimedia.org/wikipedia/commons/thumb/9/9a/Visual_Studio_Code_1.35_icon.svg/75px-Visual_Studio_Code_1.35_icon.svg.png |

### Visual Studio 2022

|  |  |
| --- | --- |
| A Microsoft fejlesztőkörnyezete jelenleg számos programozási nyelvet támogat, például: F#, C++, C# és Visual Basic, valamint az XML-t is támogatja. [[2]](#footnote-2) |  |

### XAMPP

|  |  |
| --- | --- |
| Az XAMPP egy szoftvercsomag, amely segítségével a PHP nyelvhez szükséges helyi webszerveret hozza létre, amelyet az Apache old meg, továbbá a MySQL adatbázis elérése is a XAMPP használatával történik.[[3]](#footnote-3) |  |

### Teams

|  |  |
| --- | --- |
| A Microsoft Teams a Microsoft által a Microsoft 365 termékcsalád részeként kifejlesztett csapatmunka-alkalmazás, amely munkaterületi csevegést és videokonferenciát, fájltárolást, valamint saját és harmadik féltől származó alkalmazások integrációját kínálja. [[4]](#footnote-4) | undefined |

### Github

|  |  |
| --- | --- |
| A GitHub egy fejlesztői platform, amely lehetővé teszi a fejlesztők számára a kód létrehozását, tárolását, kezelését és megosztását. Git szoftvert használ, amely minden projekthez biztosítja a Git elosztott verziókezelését, plusz hozzáférés-vezérlést, hibakövetést, szoftverfunkciókéréseket, feladatkezelést, folyamatos integrációt és wikiket.  A kaliforniai székhelyű cég 2018 óta a Microsoft leányvállalata.  Általában nyílt forráskódú szoftverfejlesztési projektek fogadására használják. 2023 januárjában a GitHub arról számolt be, hogy több mint 100 millió fejlesztővel és több mint 420 millió adattárral rendelkezik, köztük legalább 28 millió nyilvános adattárral. 2023 júniusában ez a világ legnagyobb forráskód-gazdája. [[5]](#footnote-5) | https://github.githubassets.com/assets/GitHub-Mark-ea2971cee799.png |

## Felhasznált technológiák

### HTML

|  |  |
| --- | --- |
| A HTML (angolul: HyperText Markup Language, „hiperszöveges jelölőnyelv”) egy leíró nyelv, melyet weboldalak készítéséhez fejlesztettek ki, és mára már internetes szabvánnyá vált a W3C (World Wide Web Consortium) támogatásával. Az aktuális változata az 5.[[6]](#footnote-6) |  |

### CSS

|  |  |
| --- | --- |
| A CSS (Cascading Style Sheets, magyarul: „lépcsőzetes stíluslapok”) a számítástechnikában egy stílusleíró nyelv, mely a HTML vagy XHTML típusú strukturált dokumentumok megjelenését írja le. A CSS-t a weblapok szerkesztői és olvasói egyaránt használhatják, hogy átállítsák vele a lapok színét, betűtípusait, elrendezését, és más megjelenéshez kapcsolódó elemeit. A tervezése során a legfontosabb szempont az volt, hogy elkülönítsék a dokumentumok struktúráját (HTML vagy egy hasonló leíró nyelv) a dokumentum megjelenésétől (CSS).[[7]](#footnote-7) |  |

### PHP

|  |  |
| --- | --- |
| A PHP egy általános szerveroldali szkriptnyelv dinamikus weblapok készítésére. Az első szkriptnyelvek egyike, amely külső fájl használata helyett HTML oldalba ágyazható. A kódot a webszerver PHP feldolgozómodulja értelmezi, ezzel dinamikus weboldalakat hozva létre. Ma a The PHP Group tartja fenn és fejleszti. [[8]](#footnote-8) | undefined |

### JavaScript

|  |  |
| --- | --- |
| A JavaScript programozási nyelv egy objektumorientált, prototípus-alapú szkriptnyelv, amelyet weboldalakon elterjedten használnak. Ebből fejlődött ki a TypeScript, ami a JavaScript típusos változatának tekinthető. [[9]](#footnote-9) | undefined |

### jQuery

|  |  |
| --- | --- |
| A jQuery egy JavaScript-könyvtár, amelyet a HTML DOM-fa bejárásának és kezelésének, valamint az eseménykezelésnek, a CSS-animációknak és az AJAX-nak az egyszerűsítésére terveztek. Ez egy ingyenes, nyílt forráskódú szoftver, amely a megengedő MIT licencet használja. A jQuery szintaxisát úgy tervezték, hogy megkönnyítse a dokumentumban való navigálást, a DOM-elemek kiválasztását, az animációk létrehozását, az események kezelését és az AJAX-alkalmazások fejlesztését.[[10]](#footnote-10) | jQuery 4: A New Era. jQuery, the “old-fashioned” JavaScript… | by Alex  Efimenko | Feb, 2024 | Medium |

### AJAX

|  |  |
| --- | --- |
| Az Ajax (Asynchronous JavaScript and XML) interaktív webalkalmazások létrehozására szolgáló webfejlesztési technika. Segítségével a weblap kis mennyiségű adatot cserél a szerverrel a háttérben, így a lapot nem kell újratölteni minden egyes alkalommal, amikor a felhasználó módosít valamit. Ez növeli a honlap interaktivitását, sebességét és használhatóságát.[[11]](#footnote-11) |  |

### REST API

|  |  |
| --- | --- |
| A REST (Representational State Transfer) szoftverarchitektúra típus. Egy REST típusú architektúra kliensekből és szerverekből áll. A kliensek kéréseket indítanak a szerverek felé, a szerverek kéréseket dolgoznak fel és a megfelelő választ küldik vissza. A kérések és a válaszok erőforrás-reprezentációk szállítása köré épülnek. Az erőforrás lényegében bármilyen koherens és értelmesen címezhető koncepció lehet.[[12]](#footnote-12) |  |

### MySQL

|  |  |
| --- | --- |
| A MySQL egy többfelhasználós, többszálú, SQL-alapú relációs adatbázis-kezelő szerver.  A szoftver eredeti fejlesztője a svéd MySQL AB cég, amely kettős licenceléssel tette elérhetővé a MySQL-t; választható módon vagy a GPL szabad szoftver licenc, vagy egy zárt (tulajdonosi) licenc érvényes a felhasználásra. 2008 januárjában a Sun felvásárolta 800 millió dollárért a céget. 2010. január 27-én a Sunt felvásárolta az Oracle Corporation, így a MySQL is az Oracle tulajdonába került. [[13]](#footnote-13) | MySQL API Integrations - Pipedream |

### C#

|  |  |
| --- | --- |
| A C# egy általános célú, magas szintű programozási nyelv, amely több paradigmát is támogat. A C# magába foglalja a statikus gépelést, az erős gépelést, a lexikális hatókörű, kötelező, deklaratív, funkcionális, általános, objektum-orientált (osztály-alapú) és komponens-orientált programozási tudományokat. [[14]](#footnote-14) | Writing High-Quality C# Code: Key Principles and Essential Tools | by Alex  Maher | Medium |

## Használati esetmodell

A képen szöveg, képernyőkép, diagram, kör látható

Automatikusan generált leírás

## A program felépítése

A **’TornadoPC’** mappa tartalmazza a weboldal működéséhez létfontosságú alkotóelemeket például: minden szükséges PHP fájlt és több almappát is.

**’css’** almappában található egy CSS fájl, amely minden oldal modern és reszponzív megjelenéséért felel a Bootstrap 5-tel együttműködve.

**’js’** almappa tartalmazza a főként jQuery fájlokat, amelyek segítségével hozunk létre AJAX hívásokat, kosárba vagy kedvencek közé helyezést és törlést is általuk végezzük, illetve a kosárban található termék mennyiségének növelésére, csökkentésére is itt található jQuery scripteket használunk.

A **’kepek’** mappa további almappákat és képeket is tartalmaz. Az almappákban megtalálható képek vannak megjelenítve a weboldalon, az adatbázisban a képeknek csak az elérési útvonala van elmentve és megjelenítésre felhasználva. Egyik almappa a ’kategoriak’, amelyben az egyes kategóriák képei tekinthetők meg. Másik almappa a ’termekek’, amelyben a termék neve a termék azonosítóját kapta, amely egyszerűsíti a képek feltöltését, feldolgozását. A többi kép, amelyek nincsenek almappába helyezve, több helyen is megjelenhetnek, például a kosár, kedvencek, keresés gombok ikonjai és a háttérkép is itt szerepel.

## Adatbázis

10.4.25-MariaDB

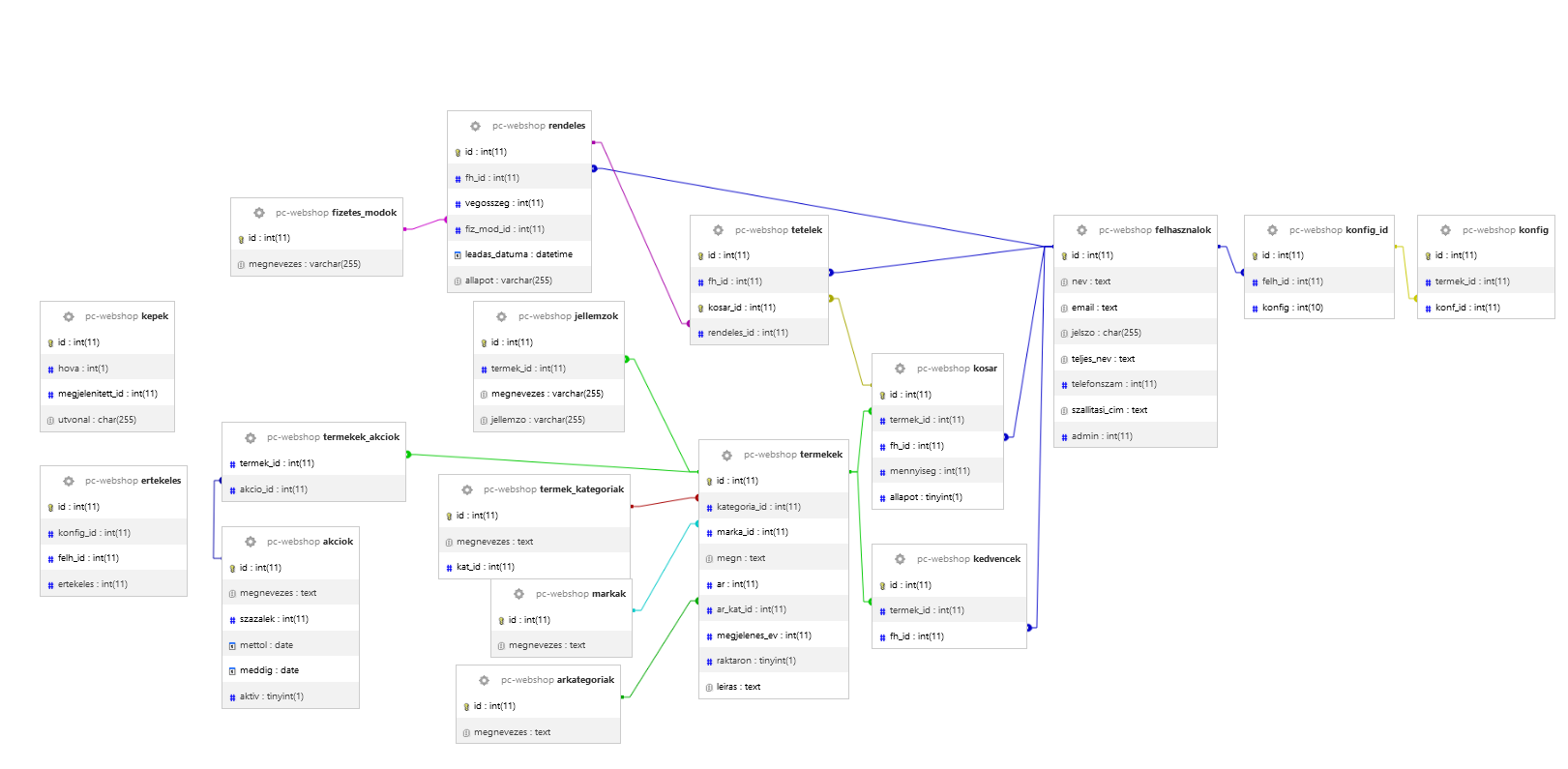
Adatbázis neve: tornado-pc

Tárolómotor: InnoDB

Alapértelmezett illesztés: utf8\_hungarian\_ci

Adattáblák:

* akciok
* arkategoriak
* ertekeles
* felhasznalok
* fizetes\_modok
* jellemzok
* kedvencek
* kepek
* konfig
* konfig\_id
* kosar
* markak
* rendeles
* termekek
* termekek\_akciok
* termek kategoriak
* tetelek



### akciok tábla



id: A tábla azonosítója, elsődleges kulcs

megnevezes: Az akció megnevezése

szazalek: Az akció értéke

mettol: Az akció kezdetének dátuma

meddig: Az akció végének dátuma

aktiv: Az akció érvényességének meghatározója

CREATE TABLE `akciok` (

`id` int(11) NOT NULL,

`megnevezes` text NOT NULL,

`szazalek` int(11) NOT NULL,

`mettol` date NOT NULL,

`meddig` date NOT NULL,

`aktiv` tinyint(1) NOT NULL

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

ALTER TABLE `akciok`

ADD PRIMARY KEY (`id`);

ALTER TABLE `akciok`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

*Ez a funkció még nincs használatban, de felkészítettük rá az adatbázist.*

### arkategoriak tábla



id: A tábla azonosítója, elsődleges kulcs

megnevezes: Az árkategória megnevezése

CREATE TABLE `arkategoriak` (

`id` int(11) NOT NULL,

`megnevezes` text NOT NULL

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

ALTER TABLE `arkategoriak`

ADD PRIMARY KEY (`id`);

ALTER TABLE `arkategoriak`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

### ertekeles tábla





id: A tábla azonosítója, elsődleges kulcs

konfig\_id: Az értékelt konfiguráció azonosítója

felh\_id: Az értékelő felhasználó azonosítója

ertekeles: Az értékelés értéke, 1 = like, 2 = dislike

CREATE TABLE `ertekeles` (

`id` int(11) NOT NULL,

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

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

`ertekeles` int(11) NOT NULL

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

ALTER TABLE `ertekeles`

ADD PRIMARY KEY (`id`);

ALTER TABLE `ertekeles`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

### felhasznalok tábla



id: A tábla azonosítója, elsődleges kulcs

nev: A felhasználó megadott felhasználóneve

email: A felhasználó e-mail címe

jelszo: A felhasználó jelszava sha1 algoritmussal titkosítva

teljes\_nev: A felhasználó teljes neve

telefonszam: A felhasználó telefonszáma

szallitasi\_cim: A felhasználó szállítási címe

admin: A felhasználó státusza, 0 = átlagos felhasználó, 1 = admin

CREATE TABLE `felhasznalok` (

`id` int(11) NOT NULL,

`nev` text NOT NULL,

`email` text NOT NULL,

`jelszo` char(255) NOT NULL,

`teljes\_nev` text NOT NULL,

`telefonszam` int(11) NOT NULL,

`szallitasi\_cim` text NOT NULL,

`admin` int(11) NOT NULL

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

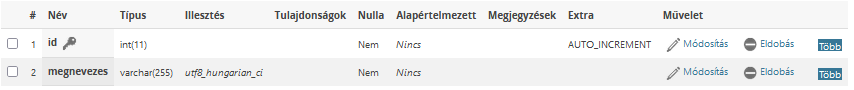
ALTER TABLE `felhasznalok`

ADD PRIMARY KEY (`id`);

ALTER TABLE `felhasznalok`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

### fizetesi\_modok tábla



id: A tábla azonosítója, elsődleges kulcs

megnevezes: A fizetési mód megnevezése

CREATE TABLE `fizetes\_modok` (

`id` int(11) NOT NULL,

`megnevezes` varchar(255) NOT NULL

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

INSERT INTO `fizetes\_modok` (`id`, `megnevezes`) VALUES

(1, 'készpénz - utánvét'),

(2, 'bankkártya - utánvét'),

ALTER TABLE `fizetes\_modok`

ADD PRIMARY KEY (`id`);

ALTER TABLE `fizetes\_modok`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

### jellemzok tábla



id: A tábla azonosítója, elsődleges kulcs

termek\_id: A termék azonosítója, külső kulcs

megnevezes: A jellemző megnevezése

jellemzo: A jellemző leírása

CREATE TABLE `jellemzok` (

`id` int(11) NOT NULL,

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

`megnevezes` varchar(255) NOT NULL,

`jellemzo` varchar(255) NOT NULL

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

ALTER TABLE `jellemzok`

ADD PRIMARY KEY (`id`),

ADD KEY `termek\_id` (`termek\_id`);

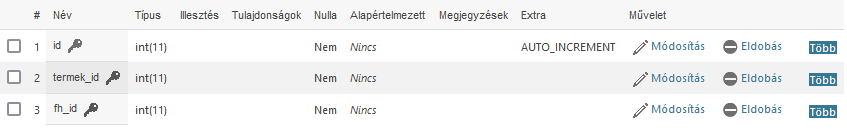
ALTER TABLE `jellemzok`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

ALTER TABLE `jellemzok`

ADD CONSTRAINT `jellemzok\_ibfk\_1` FOREIGN KEY (`termek\_id`) REFERENCES `termekek` (`id`);

### kedvencek tábla



id: A tábla azonosítója, elsődleges kulcs

termek\_id: A termék azonosítója, külső kulcs

fh\_id: A felhasználó azonosítója, külső kulcs

CREATE TABLE `kedvencek` (

`id` int(11) NOT NULL,

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

`fh\_id` int(11) NOT NULL

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

ALTER TABLE `kedvencek`

ADD PRIMARY KEY (`id`),

ADD KEY `termek\_id` (`termek\_id`),

ADD KEY `fh\_id` (`fh\_id`);

ALTER TABLE `kedvencek`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

ALTER TABLE `kedvencek`

ADD CONSTRAINT `kedvencek\_ibfk\_1` FOREIGN KEY (`termek\_id`) REFERENCES `termekek` (`id`),

ADD CONSTRAINT `kedvencek\_ibfk\_2` FOREIGN KEY (`fh\_id`) REFERENCES `felhasznalok` (`id`);

### kepek tábla



id: A tábla azonosítója, elsődleges kulcs

hova: A kép megjelenítésének helye, 1 = kategória, 2 = termék

megjelenitett\_id: A kép megjelenítéséhez szükséges azonosító

utvonal: A kép elérési útvonala

CREATE TABLE `kepek` (

`id` int(11) NOT NULL,

`hova` int(1) NOT NULL,

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

`utvonal` char(255) NOT NULL

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

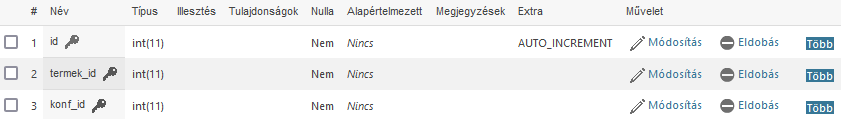
ALTER TABLE `kepek`

ADD PRIMARY KEY (`id`);

ALTER TABLE `kepek`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

### konfig tábla



id: A tábla azonosítója, elsődleges kulcs

termek\_id: A konfigurációhoz tartozó termék azonosítója, külső kulcs

konf\_id: A konfiguráció azonosítója, amihez a termék tartozik, külső kulcs

CREATE TABLE `konfig` (

`id` int(11) NOT NULL,

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

`konf\_id` int(11) NOT NULL

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

ALTER TABLE `konfig`

ADD PRIMARY KEY (`id`),

ADD KEY `termek\_id` (`termek\_id`, `konf\_id`),

ADD KEY `konf\_id` (`konf\_id`);

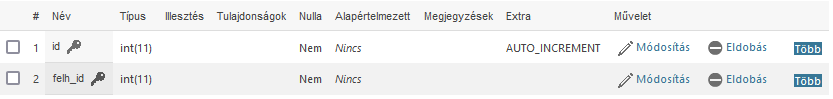
ALTER TABLE `konfig`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

ALTER TABLE `konfig`

ADD CONSTRAINT `konfig\_ibfk\_1` FOREIGN KEY (`konf\_id`) REFERENCES `konfig\_id` (`id`);

### konfig\_id tábla



id: A konfiguráció azonosítója, elsődleges kulcs

felh\_id: A konfiguráció melyik felhasználóhoz tartozik, külső kulcs

CREATE TABLE `konfig\_id` (

`id` int(11) NOT NULL,

`felh\_id` int(11) NOT NULL

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

ALTER TABLE `konfig\_id`

ADD PRIMARY KEY (`id`),

ADD KEY `felh\_id` (`felh\_id`);

ALTER TABLE `konfig\_id`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

ALTER TABLE `konfig\_id`

ADD CONSTRAINT `konfig\_id\_ibfk\_1` FOREIGN KEY (`felh\_id`) REFERENCES `felhasznalok` (`id`);

### kosar tábla



id: A tábla azonosítója, elsődleges kulcs

termek\_id: A termék azonosítója, külső kulcs

fh\_id: A felhasználó azonosítója, külső kulcs

mennyiseg: Az adott termék mennyisége

allapot: A kosárban levő termék állapota, 0 = nem megrendelt (megjelenik kosárban), 1 = megrendelt

CREATE TABLE `kosar` (

`id` int(11) NOT NULL,

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

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

`mennyiseg` int(11) NOT NULL,

`allapot` tinyint(1) NOT NULL

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

ALTER TABLE `kosar`

ADD PRIMARY KEY (`id`),

ADD KEY `termek\_id` (`termek\_id`),

ADD KEY `fh\_id` (`fh\_id`);

ALTER TABLE `kosar`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

ALTER TABLE `kosar`

ADD CONSTRAINT `kosar\_ibfk\_3` FOREIGN KEY (`termek\_id`) REFERENCES `termekek` (`id`),

ADD CONSTRAINT `kosar\_ibfk\_4` FOREIGN KEY (`fh\_id`) REFERENCES `felhasznalok` (`id`);

### markak tábla



id: A tábla azonosítója, elsődleges kulcs

megnevezes: A márka megnevezése

CREATE TABLE `markak` (

`id` int(11) NOT NULL,

`megnevezes` text NOT NULL

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

ALTER TABLE `markak`

ADD PRIMARY KEY (`id`);

ALTER TABLE `markak`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

### rendeles tábla



id: A tábla azonosítója, elsődleges kulcs

fh\_id: A felhasználó azonosítója, külső kulcs

vegosszeg: A fizetendő végösszeg

fiz\_mod\_id: A fizetési mód azonosítója, külső kulcs

leadas\_datuma: A leadás dátuma

allapot: A rendelés állapota

CREATE TABLE `rendeles` (

`id` int(11) NOT NULL,

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

`vegosszeg` int(11) NOT NULL,

`fiz\_mod\_id` int(11) NOT NULL,

`leadas\_datuma` datetime NOT NULL,

`allapot` varchar(255) NOT NULL

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

ALTER TABLE `rendeles`

ADD PRIMARY KEY (`id`),

ADD KEY `fiz\_mod\_id` (`fiz\_mod\_id`),

ADD KEY `fh\_id` (`fh\_id`);

ALTER TABLE `rendeles`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

ALTER TABLE `rendeles`

ADD CONSTRAINT `rendeles\_ibfk\_3` FOREIGN KEY (`fiz\_mod\_id`) REFERENCES `fizetes\_modok` (`id`),

ADD CONSTRAINT `rendeles\_ibfk\_4` FOREIGN KEY (`fh\_id`) REFERENCES `felhasznalok` (`id`);

### termekek tábla



id: A tábla azonosítója, elsődleges kulcs

kategoria\_id: A kategória azonosítója, külső kulcs

marka\_id: A termék márkájának azonosítója, külső kulcs

megn: A termék megnevezése

ar\_kat\_id: A termék árkategóriájának azonosítója, külső kulcs

megjelenes\_ev: A termék megjelenésének éve

raktaron: A termékből van-e raktáron? (Még nem működő funkció, fejlesztés alatt)

leiras: A termék leírása

CREATE TABLE `termekek` (

`id` int(11) NOT NULL,

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

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

`megn` text NOT NULL,

`ar` int(11) NOT NULL,

`ar\_kat\_id` int(11) NOT NULL,

`megjelenes\_ev` int(11) NOT NULL,

`raktaron` tinyint(1) NOT NULL,

`leiras` text NOT NULL

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

ALTER TABLE `termekek`

ADD PRIMARY KEY (`id`),

ADD KEY `kategoria\_id` (`kategoria\_id`),

ADD KEY `marka\_id` (`marka\_id`),

ADD KEY `ar\_kat\_id` (`ar\_kat\_id`);

ALTER TABLE `termekek`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

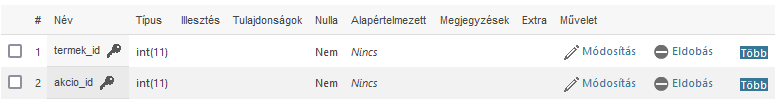
ALTER TABLE `termekek`

ADD CONSTRAINT `termekek\_ibfk\_1` FOREIGN KEY (`marka\_id`) REFERENCES `markak` (`id`),

ADD CONSTRAINT `termekek\_ibfk\_3` FOREIGN KEY (`kategoria\_id`) REFERENCES `termek\_kategoriak` (`id`),

ADD CONSTRAINT `termekek\_ibfk\_4` FOREIGN KEY (`ar\_kat\_id`) REFERENCES `arkategoriak` (`id`);

### termekek\_akciok



termek\_id: A termék azonosítója, külső kulcs

akcio\_id: Az akció azonosítója, külső kulcs

CREATE TABLE `termekek\_akciok` (

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

`akcio\_id` int(11) NOT NULL

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

ALTER TABLE `termekek\_akciok`

ADD KEY `termek\_id` (`termek\_id`),

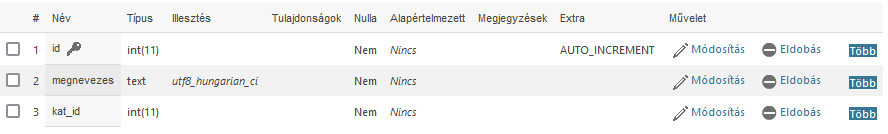
ADD KEY `akcio\_id` (`akcio\_id`);

ALTER TABLE `termekek\_akciok`

ADD CONSTRAINT `termekek\_akciok\_ibfk\_2` FOREIGN KEY (`termek\_id`) REFERENCES `termekek` (`id`),

ADD CONSTRAINT `termekek\_akciok\_ibfk\_3` FOREIGN KEY (`akcio\_id`) REFERENCES `akciok` (`id`);

### termek\_kategoriak tabla



id: A tábla azonosítója, elsődleges kulcs

megnevezes: A kategória megnevezése

kat\_id: A kategóriának megadott azonosító

CREATE TABLE `termek\_kategoriak` (

`id` int(11) NOT NULL,

`megnevezes` text NOT NULL,

`kat\_id` int(11) NOT NULL

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

ALTER TABLE `termek\_kategoriak`

ADD PRIMARY KEY (`id`);

ALTER TABLE `termek\_kategoriak`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

### tetelek tábla



id: A tábla azonosítója, elsődleges kulcs

fh\_id: A felhasználó azonosítója, külső kulcs

kosar\_id: A kosár azonosítója, külső kulcs

rendeles\_id: A rendelés azonosítója, külső kulcs

CREATE TABLE `tetelek` (

`id` int(11) NOT NULL,

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

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

`rendeles\_id` int(11) NOT NULL

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

ALTER TABLE `tetelek`

ADD PRIMARY KEY (`id`),

ADD UNIQUE KEY `kosar\_id` (`kosar\_id`),

ADD KEY `rendeles\_id` (`rendeles\_id`),

ADD KEY `fh\_id` (`fh\_id`);

ALTER TABLE `tetelek`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

ALTER TABLE `tetelek`

ADD CONSTRAINT `tetelek\_ibfk\_1` FOREIGN KEY (`kosar\_id`) REFERENCES `kosar` (`id`),

ADD CONSTRAINT `tetelek\_ibfk\_3` FOREIGN KEY (`rendeles\_id`) REFERENCES `rendeles` (`id`),

ADD CONSTRAINT `tetelek\_ibfk\_4` FOREIGN KEY (`fh\_id`) REFERENCES `felhasznalok` (`id`);

## Rendszerkövetelmények

Mesterremekünk webshop lévén, szükség van egy mai modern böngészőre, lehet ez Google Chrome, Mozilla Firefox, Opera és Opera GX is. Szükséges lehet néhány bővítmény telepítése a technológiák miatt, és a JavaScript futtatását is engedélyezni kell. A webes alkalmazás egy adatbázis-kezelő rendszert is használ, amely telepítése szükséges a helyi vagy webszerverre, amiről az alkalmazást szeretnénk futtatni. Futtatáshoz szükséges vezetékes, vagy vezeték nélküli internetkapcsolat. Alkalmazásunkat helyi szerveren fogjuk bemutatni, ezért szükségünk van egy webalkalmazás kiszolgáló helyi alkalmazásra, ami az Apache http szerver, a MySQL adatbázis kezelő rendszer és ezen kívüli más komponensek integrált csomagja, ami jelen esetben a Xampp lesz.

Követelmények a Xampp által biztosított webszerverhez:

* Windows, Linux vagy Mac OS operációs rendszer
* 5-8 GB tárhely (Minél több új terméket adunk kínálatunkhoz, úgy változhat a

tárhely igény)

* Internetböngésző, javasolt a Mozilla Firefox vagy Google Chrome
* MySQL legfrissebb verziója
* PHP legfrissebb verziója

## Megfelelő indítás

Alkalmazásunk a dokumentáció írásakor még nem található meg semmilyen webszerveren, ezért egy GitHub link szükséges a letöltéshez. Az előbb említett Xampp legfrissebb verzióját is szükséges telepíteni az alkalmazásunk pontos és megfelelő működéséhez. A Xampp magában foglalja az Apache http szervert, és a MySQL adatbáziskezelő rendszert is. Ez egy egyszerű telepítő-programmal, néhány kattintással telepíthető egyszerű folyamat.

1. A GitHubos linkről letöltjük a forrásfájlokat, amiket a Xampp miatt szigorúan, a C meghajtó ’xampp’ mappáján belül a ’htdocs’ mappába helyezzük el.
2. Elindítjuk a Xampp Control Panelt, ezen belül pedig, az Apache és MySQL modulokat.
3. Böngészőben meg kell nyitnunk a [http://localhost/phpmyadmin/](http://localhost:84/phpmyadmin/) oldalt, és importálnunk kell a ’tornado-pc.sql’ fájlt.
4. Végül elindítjuk az alkalmazásunkat. Beírjuk a címet: ’localhost’, és kész. Webes alkalmazásunk készen áll a bemutatásra!

A webszerveren vagy localhost-on való futtatás ugyanúgy működik, nincs köztük működésbeli különbség, csak furcsa lehet, hogy egy gépről megy a rendelés, és az adminisztrátori funkciók is ott működnek. A védés idejére igyekszünk oldalunk egy webcímen való megosztására.

## A weboldal felépítése

A weboldal az objektum orientált programozás és a tiszta kód elvei szerint készült. Több osztályt használunk a webshop működéséhez. ’Webshop’ osztály tartalmazza a webshop működéséhez szolgáló metódusokat. ’DB’ osztályban találhatóak az adatbázis csatlakozást létrehozó és az SQL utasítások futtatására használható metódusok. ’Felhasznalo’ osztály a regisztráló és bejelentkező űrlapok megjelenítésére és feldolgozására szolgáló metódusokat tartalmazza. A ’Rendeles’ osztályban pedig a rendeléshez szükséges adatok bekérő és feldolgozó, illetve a rendelési előzményeket megjelenítő metódusok vannak elhelyezve. HTML, PHP és JavaScript nyelveket használtunk, a frontend saját CSS és Bootstrap 5 segítségével valósítottuk meg, a reszponzivitásra is odafigyelve.

### Főoldal

Létrehoztunk az ’index.php’ fájlban egy ’$webshop’ objektumot és ezen keresztül illesztjük be a ’head()’ metódus által a <head> részben szükséges adatok és például a saját CSS és Bootstrap linkelés is, így, ha bármilyen változtatást szükséges eszközölnünk, akkor elég lesz csak a metódusban egyszer módosítani, nem kell minden oldalon. Továbbá ilyen módon jelenik meg a navigációs sáv, a kategóriák és a lábléc is.

A kategóriák három nagyobb főkategóriába vannak osztva, a három főkategória alá tartozó kategóriák kártyái egy-egy lenyíló részen belül tekinthetőek meg, ahonnan tovább lehet navigálni az adott kategóriába tartozó termékekhez. Ugyanez a három főkategória a további kategóriáival a navigációs sávon is elérhető. Minden kategória egyedi ID-vel, azonosítóval rendelkezik, GET kérésen keresztül paraméterként átadásra kerül ez az ID és megjelennek a termékek egy-egy kártyán.

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

## Termékek oldal

### termekekMegjelenit()

Ez a metódus a következő funkciókat látja el:

* rendezés
* lapozás
* adatbázisból lekérdezzük a termék adatait
* keresés
* megjelenítés kártyákkal

Bemeneti paraméternek meg kell adni a jelenlegi oldal számát ($p). Két opcionális paramétert fogad: ’$kategoria\_id’, a kiválasztott kategória ID-je, és ’$keresett’, ami a keresett kulcsszó lesz.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírásA képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírásTartalmaz három különböző lekérdezést, amelyek azt a célt szolgálják, hogy három funkció közül a megfelelőt tudjuk hasznosítani (összes, kiválasztott kategória és keresett kulcsszó termékei).

#### Rendezés

Az alapértelmezetten ABC sorrendben jelennek meg a termékek, de a felhasználó egy legördülő listából tud választani további kettő rendezési mód közül: ár csökkenő, ár növekvő. Annak érdekében, hogy megmaradjon a rendezés lapozás vagy oldalfrissítés esetén is, el kell menteni a választott módot egy SESSION változóba. Ezután a megfelelő rész a ’$rendez’ változóban kerül elmentésre, amely pedig az SQL utasításokban lesz felhasználva.

#### LapozásA képen szöveg, képernyőkép, Betűtípus, szám látható Automatikusan generált leírás

A lapozás lehetséges az első, utolsó oldalra, valamint az oldalszámokra kattintva is. Ki van számolva a kezdőérték, ami mindig változik a lapozás által és az oldalak száma. Ki van nyerve az URL olyan formában, hogy utána lehessen illeszteni a lapozás oldalszámát. Ha oldalszámra kattintunk, akkor kiszámolja hány oldalra lesz szükség és megjeleníti ezeket. Visszatérési értéke egy tömb, amely tartalmazza, hogy egy oldalon mennyinek kell megjelenni és a kezdőértéket, amelyek az SQL utasítás LIMIT záradékában lesznek felhasználva.

#### Összes termék lekérdezése

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

Amikor a navigációs sávon kiválasztjuk az ’Összes termék’ menüpontot. Mint a neve is mutatja az összes adatbázisban tárolt termék megjelenik, ekkor átadásra kerül a kategória ID, ami nulla lesz. Jelen esetben 12 darab termék lesz látható egy oldalon, eszerint a beállítás szerint.

#### Kiválasztott kategória termékei

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Amennyiben a metódus megkapott kategória azonosítója nagyobb, mint nulla, akkor az adott kategória termékei lesznek lekérdezve.

#### Keresett kulcsszó termékei

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

Ha a navigációs sáv keresősávját használjuk, akkor a keresett termékek lekérdezésére az itt látható módon kerül sor.

A termékek a megfelelő SQL utasítás kiválasztása után végezetül egy asszociatív tömbbe kerülnek, a saját metódusunk által, amely így használható:

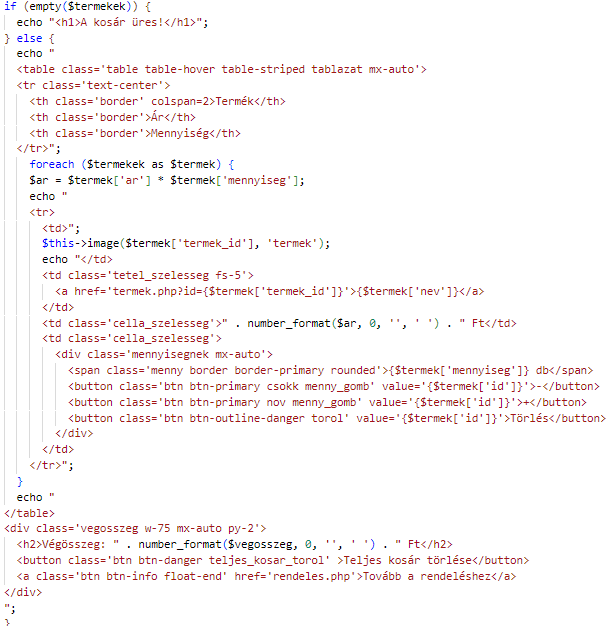
### Megjelenítés

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

A termékek kártyán jelennek meg képpel, megnevezéssel és árral. Kosárba és kedvencek közé innen is helyezhetők. Minden termékről megjeleníthető egy részletes, külön oldal, amelyhez a ’Tovább’ gombbal navigálhatunk el.

## Kosár



A kosár elemei adatbázisban vannak elmentve, ezek vannak megjelenítve egy táblázatban. Abban az esetben, ha a bejelentkezett felhasználó nem helyezett semmit a kosárba, ’A kosár üres!’ felirat jelenik meg. Megjeleníti a táblázatban a termék képét, megnevezését, árát, mennyiségét. A táblázat alatt megtalálható a végösszeg, egy ’Teljes kosár törlése’ és egy ’Tovább a rendeléshez’ gomb. A termék ára és a végösszeg dinamikusan változnak a darabszám módosításának függvényében. Van lehetőség a mennyiség növelésére, csökkentésére 1-99 között, egy termék törlésére és a teljes kosár törlésére, ezen funkciók egy-egy gomb segítségével érhetők el.

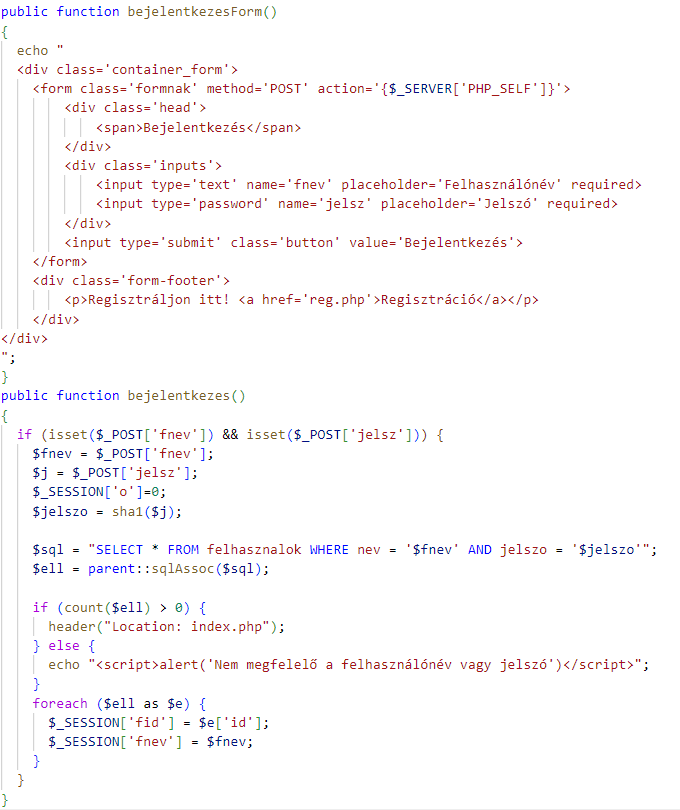
A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírásA kosárba helyezés a termék ID-je alapján történik, mennyiséggel együtt. A termék mennyisége a ’menny’ azonosítójú beviteli mezőtől függ. Amennyiben a beírt érték nulla vagy negatív szám, esetleg nincs megadva érték, akkor mindenféleképpen 1 lesz. Akkor is ez lesz érvényes, ha csak egy kosárba helyez gomb található, nincsen beviteli mező a darabszámnak, például kedvencek közül kosárba helyezés. Ha érvényes a szám akkor ellenőrizzük, hogy ne legyen 99-től nagyobb az érték, végül, ha megfelelő a beírt darabszám, akkor lesz elmentve. Egy AJAX hívás segítségével átadásra kerül a termék azonosítója és a mennyiség is, ezután következik az adatbázisba mentés. Amennyiben sikeres vagy sikertelen, megjelenik ’alert()’ formájában egy üzenet.

## Kedvencek

A kedvencek közé helyezett termékek szintén adatbázisban vannak elmentve és táblázatban megjelenítve, hasonlóan a kosárhoz, csak egyszerűbben. Nincsen mennyiség eltárolva, ebből adódóan nem lehet módosítani sem a darabszámot. A kedvencek között lévő termékeket el lehet távolítani és kosárba helyezni, ebben az esetben egy darab kerül át a kosárba, viszont továbbra is megmarad a kedvencek között.

## Belépés

Ez a kódrészlet a bejelentkezés oldal fő részét hozza létre. A bejelentkezesForm() eljárás létrehozza a bejelentkezési adatok beírására alkalmas űrlapot, és a bejelentkezés gombot. A bejelentkezes() metódus magát a bejelentkezési folyamatot hajtja végre. Először is ellenőrzi, hogy a felhasználó megadta-e a felhasználó nevét, és jelszavát. Ha ez a feltétel teljesült, akkor fut tovább a kód. A megadott felhasználónevet, és jelszót elmenti egy-egy változóba, ezután egy sql parancs futtatásával megnézi, hogy az adatbázis tartalmaz-e ilyen néven regisztrált, és ezzel a jelszóval rendelkező felhasználót. Ha tartalmaz, akkor átirányít az ’index.php’ oldalra, és a felhasználónevet elmenti munkamenet változóként.

## Regisztráció

Ez a kódrészlet a regisztráció oldal fő részét hozza létre. A regisztracioForm() eljárás létrehozza a regisztráláshoz szükséges adatok beírására alkalmas űrlapot, és a regisztráció gombot. A regisztracio() metódus ellenőrzi, hogy a felhasználó megadott-e minden adatot, ha ez a feltétel teljesült, akkor fut tovább. A megadott adatokat átadja egy-egy változónak, a jelszót titkosítja sha1 hash titkosításban. Ezután lefuttat egy paraméteres sql parancsot, ami siker esetén felugró ablakban megjeleníti ’Regisztráció sikeres’, hiba esetén pedig ugyanilyen felugró ablakban megjeleníti a hibát.

## Konfiguráció összerakó

A konfigurációépítő oldalon a felhasználók összeállíthatják álmaik számítógépét, vagy azt a mit a pénztárcájuk megenged. Ezután menthetik az adatbázisba, és a kosárhoz is hozzáadhatják.

A fenti kódrészlet létrehoz egy űrlapfelületet, ebben egy kettő oszlopos táblázattal. A táblázat bal oldali oszlopában a kategóriák jelennek meg, jobb oldalt pedig a termékek egy lenyíló select input mezőben. A megjelenítéshez kettő darab foreach ciklust használtam. A külső foreach ciklus a táblázat adott sorának kettő oszlopáért felelős, a belső foreach ciklus pedig a select input mezőkhöz való opciók megadásáról gondoskodik. Az űrlap beküldésekor lefut a ’bekuld.php’ fájl. Ez felelős az adatbázisban való rögzítésért.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírásA beküldés gombra nyomás után a ’Post’ tömb ’sub’ mezője kap egy értéket. A feltételvizsgálat azt vizsgálja, hogy a felhasználó megnyomta-e a gombot. Az első sql parancs hozzáad egy új adatsort a ’konfig\_id’ táblához. A for ciklusban lévő többször lefutó sql kód a ’konfig’ táblához ad hozzá adatot. Minden lefutáskor rögzíti a konfiguráció azonosítóját, és a hozzáadott alkatrész azonosítóját is.

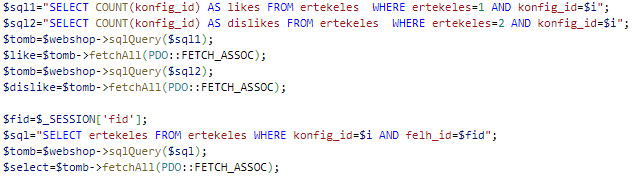
## A konfiguráció megtekintő, és értékelő oldal

### Card elemek

Ezt az oldalt csak a bejelentkezett felhasználók tudják elérni egy menüpontról. Itt saját, és más felhasználók által összerakott konfigurációkat lehet megtekinteni, és értékelni. A központi div tárolóban oldalanként négy darab konfiguráció jelenik meg. Minden konfiguráció egy külön Bootstrap-es ’card’ elemként jelenik meg, a like és dislike gombbal együtt.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírásA fenti kódsor adja hozzá a kártyákhoz a konfiguráció alkatrészeit táblázatos formában. Hasonlóan, mint a konfiguráció összerakó oldalon, de a legnagyobb különbség az, hogy itt nincsenek ’select’ mezők.



Ez a kódrészlet felelős azért, hogy lekérje az adott konfigurációhoz a like-ok és dislike-ok számát, illetve meghatározni, hogy a jelenleg bejelentkezett felhasználó értékelte-e a konfigurációt, és ha igen, mivel értéke.

### Ajax hívások

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírásA like - dislike funkció működését AJAX hívásokkal oldottuk meg. Egy adott konfigurációhoz tartozó gombok kapnak egy külön ’like’ és ’dislike’ osztályt, innen tudja a program, hogy melyik AJAX hívást kell végrehajtani.

A ’fid’ változó a bejelentkezett felhasználó azonosítója az adatbázisban, az ’id[0]’ a megnyomott gomb azonosítójának első karaktere, ezt adja tovább, mint a konfiguráció azonosítója. A mentett értékelés értéke kettő lehet. Ha a felhasználó like-olt, akkor ’1’, ha dislike-olt, akkor ’2’.

### rating.php

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás Mindkettő AJAX hívás ugyanarra a PHP fájlra küldi tovább az adatokat, és ebből a fájlból indulnak az adatbázisműveletek is. A felhasználó módosíthatja az értékelését, vagy törölheti, ha ugyanarra a gombra nyom, amit már választott.

A kód kezdetben megnyitja, a munkamenetet, és külön változókba teszi a ’$\_POST’ tömb elemeit, ezután kettő adatbázis-lekérdezést hajt végre. Az első lekérdezés lekérdezi, hogy az adott felhasználó értékelte-e már az adott konfigurációt, a második lekérdezés az értékelt konfiguráció értékelését kérdezi le. Ezután feltételvizsgálattal eldönti az adott felhasználó értékelésének számát az adott konfiguráción. 0 érték esetén egyszerűen hozzáadja az adatbázishoz a megfelelő adatokat. Ha a felhasználó már értékelte az adott konfigurációt, akkor frissíti az adatbázis adatsorát, ahol a konfiguráció és a felhasználó azonosítója azonos a megadottakkal. Ellenkező esetben törli az adatbázis azon adatsorát, ahol az előző feltételek teljesülnek.

### limit.php

A nagyon sok konfiguráció betöltésénél való problémákat, a megjelenített konfigurációk limitálásával oldottuk meg. Mindig csak négy darab konfigurációt jelenít meg az oldalon. Bejelentkezéskor létrejön egy ’o’ változó a munkamenetben. Ez adja meg az oldalszámot, ez változik minden lapozásnál, vagyis, ha a felhasználó átmegy az oldal más részére, és visszatér, ugyanott folytatja a konfigurációk böngészését.



Az oldalváltó gombok egy ’e’ és ’k’ azonosítót kaptak, ezeket adják tovább a ’limit.php’ fájlnak.

## Tesztelés, tesztesetek

A munkavégzés közben többször teszteltük az elkészített kódokat. Az alábbi tesztelési módszereket, technikákat végeztük el weboldalunk kódjainak tesztelésére.

### Unit tesztek:

A unit tesztek arra összpontosítanak, hogy az egyes komponensek vagy egységek (modulok, osztályok, objektumok) jól működnek-e elkülönülve. A unit tesztek segítenek ellenőrizni az egyes funkciók helyes működését és stabilitását, könnyebb hibákat találni az egyes komponensekben.

### Integrációs tesztek:

Az integrációs tesztek azt vizsgálják, hogy a különálló egységek együttesen tudnak-e helyesen működni. Az integrációs tesztek segítenek abban, hogy az egyes komponensek közötti interakciók megfelelően működjenek, és elkerüljék az esetleges összeütközéseket vagy hibákat.

### Funkcionalitási tesztek:

A weboldalunkon létrehozott funkciókat is teszteltük, annak érdekében, hogy minden hibát kiszűrjünk, úgy működnek-e, ahogyan elterveztük.

* **Kategóriák és termékek:** Ellenőrizni kell, hogy a termékek jól vannak-e csoportosítva kategóriákba, illetve mindig a megfelelő termékek jelennek-e meg.
* **Keresés, rendezés, lapozás:** Termékek között lehetősége van a felhasználónak keresni, a termékeket rendezni, például: ár szerint növekvő, csökkenő rendezés. Lapozás, annak érdekében, hogy ne legyen túlterhelve a weboldal, amiért az összes termék meg lenne jelenítve, ezért csak néhány található meg egy oldalon. Ezeknek a funkcióknak a tesztelését is elvégeztük.
* **Kosár, kedvencek és megrendelés:** A tesztelések alkalmával meg kell győződni róla, hogy a vásárlók sikeresen hozzáadhatnak-e termékeket a kosárhoz, kedvencekhez, megtekinthetik-e a tartalmát ezeknek, és sikeresen leadhatnak-e megrendeléseket.

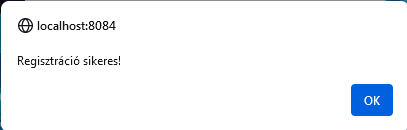
### Megjelenés, felhasználói élmény tesztelése:

A megjelenésre is próbáltunk odafigyelni, egy modern, letisztult, felhasználóbarát kinézetet szerettünk volna létrehozni. Nem szerettük volna, hogy legyen olyan, ami a felhasználói élményt leronthatná.

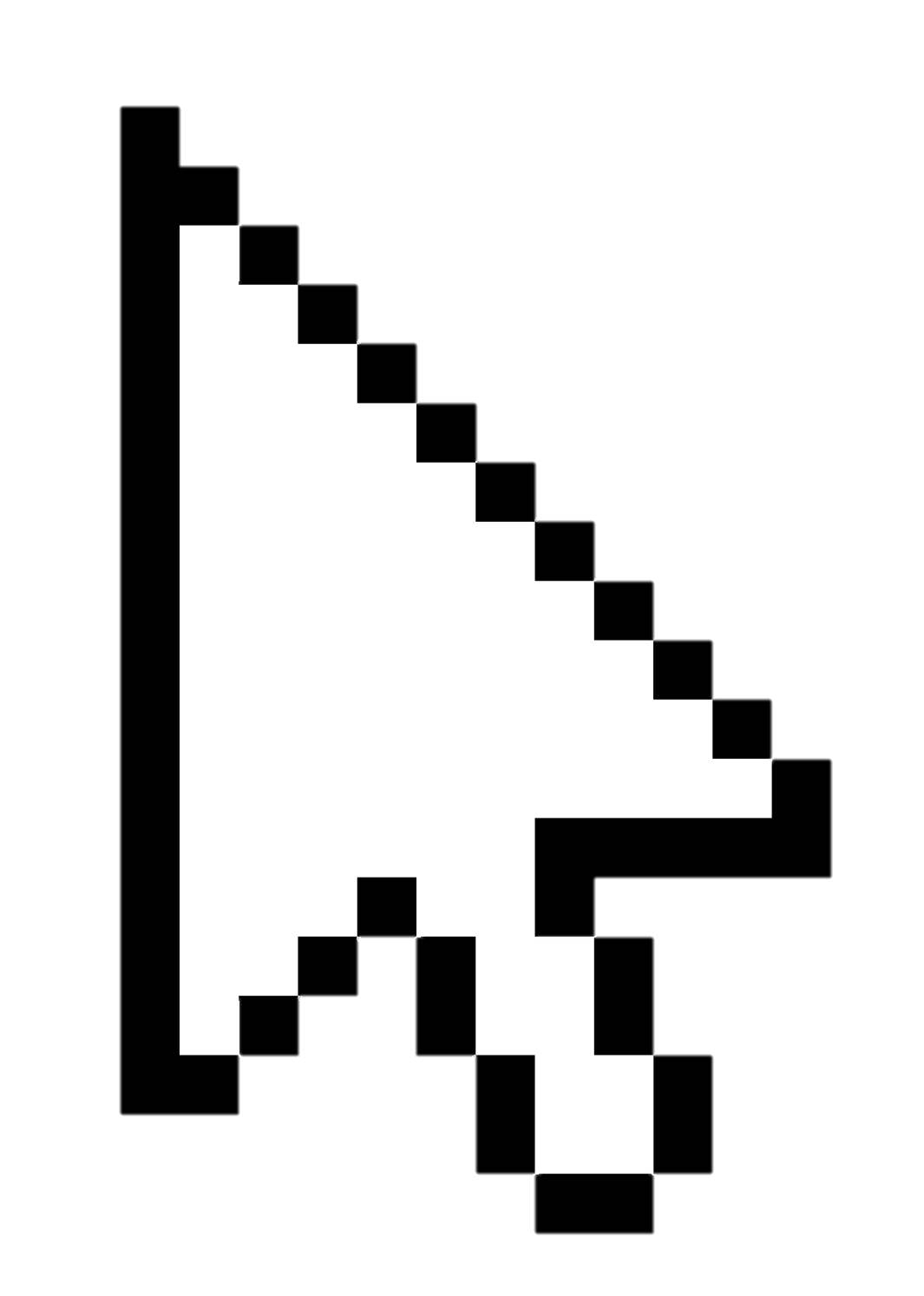
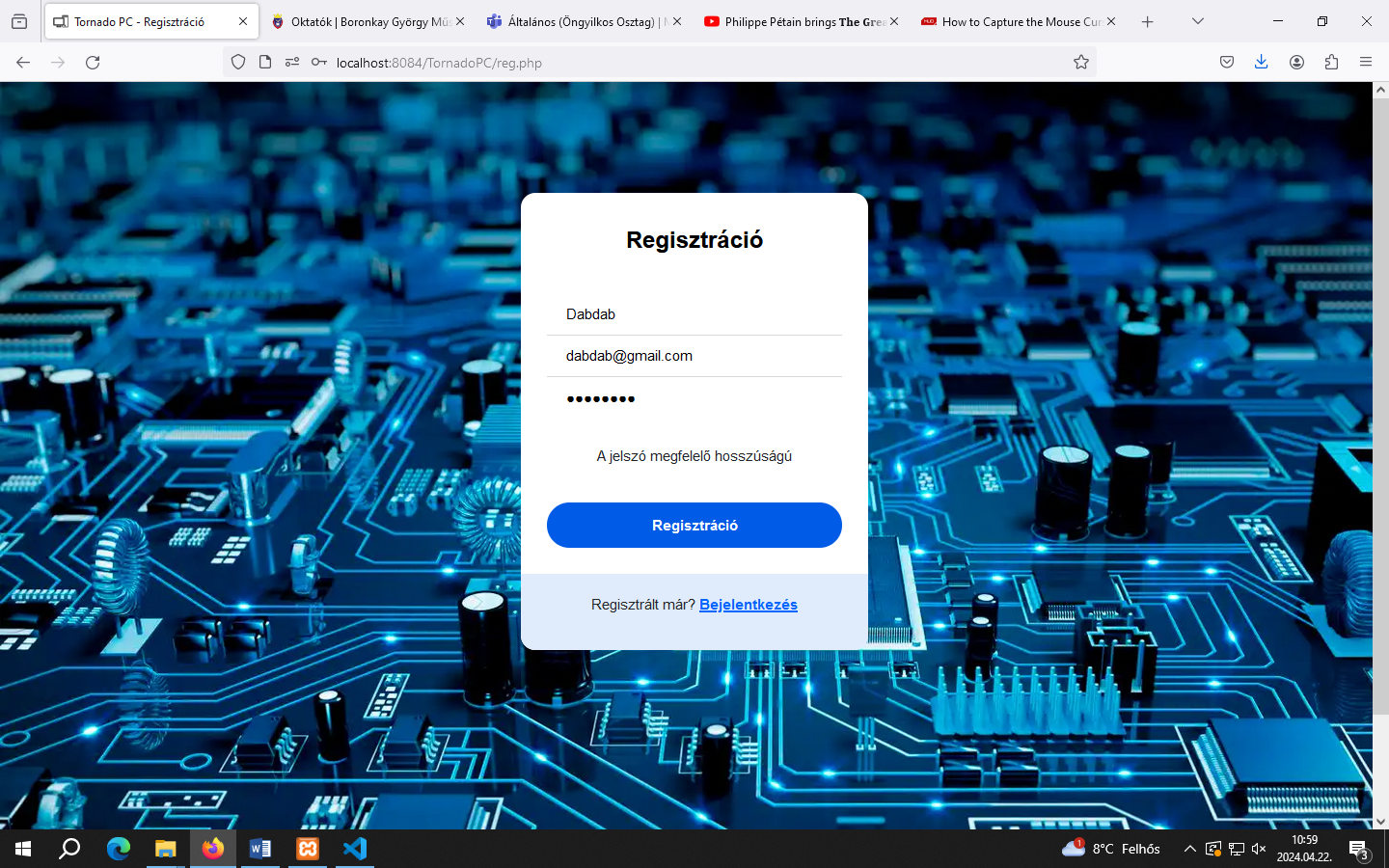
* **Reszponzivitás:** Weboldalunkat minden eszközökön és különböző méretű képernyőkön is leteszteltük. Azt néztük, hogy úgy jelenik-e meg az oldal, hogy semmi ne legyen takarásban, elcsúszva, nem esztétikusan megjelenítve vagy a funkciók működése hibás lenne-e.
* **Használhatóság:** Arra törekedtünk, hogy bizonyos funkciók több helyről is elérhetőek legyenek. A kategóriák megjelennek a navigációs sávon és a főoldalon is, ezzel is segítve a vásárlót, hogy a megfelelő kategória, termék megtalálása gyorsabb legyen, több helyről is elérhető legyen. A kereső használatával is segíteni próbáljuk a vásárlást. Ha megtalálta a terméket, akkor kosárba és kedvencek közé is helyezheti anélkül, hogy a termék részletes oldalát megnyitná, ahol természetesen szintén megtalálhatóak ezek a funkciók. Ezeket is mind le kellett ellenőrizni, hogy úgy működjenek, ahogyan szerettük volna.

### Regisztráció, bejelentkezés

Nagy hangsúlyt fektettünk a regisztráció szolgáltatás, és az adatbáziskezelő asztali alkalmazás bejelentkezési felületének tesztelésére. Az oldalra való regisztráció oldalán a jelszónak minimum 8 karakter hosszúnak kell lennie, és tartalmaznia kell legalább egy kisbetűt, egy nagybetűt, egy számot és egy különleges karaktert. Az oldal kiírja, a jelszó jelenlegi hosszát. Tesztelésként ’Gabooo’ néven fogok regisztrálni, a ’gabooo@gmail.com’ email címmel, és ’Gabooo\_20’ jelszóval.

A ’Minden jó’ felirat a jelszóra utal, ezt azt jelenti, hogy minden feltétel teljesült, és minden szükséges dolgot tartalmaz a jelszó. A regisztráció gombra nyomva, egy JavaScript ’alert()’ ablakot fog megjeleníteni, ebben a regisztráció sikerességéről szóló üzenettel.

Abban az esetben, amikor a jelszó nem felel meg a megadott követelményeknek, egy másik tesztesetre is szükség van, mert meg kell vizsgálni a teljesség kedvéért. Felhasználónévként megadott adat a ’Dabdab’ lesz, ’dabdab@gmail.com’ email címmel, és ’dabdab11’ jelszóval. Ebben az esetben az oldal nem változik, de a ’Regisztráció’ feliratú gomb nem lesz kattintható.



### Rendelés

A képen szöveg, képernyőkép, Betűtípus, diagram látható

Automatikusan generált leírásA kosárba helyezett termékeket meg szeretnénk rendelni, akkor a személyes adatok, szállítási cím megadása után, abban az esetben, ha a rendelés sikeres, akkor egy pipa kell, hogy fogadja a vásárlót és az adatai a jobb oldalon meg kell, hogy jelenjenek.

Viszont, ha valami probléma adódik a rendelés közben, akkor ez az oldal lesz látható. Ekkor csak a főoldalra van lehetőség visszamenni.

A képen embléma, szimbólum, Grafika, Betűtípus látható

Automatikusan generált leírás

## Továbbfejlesztési lehetőségek

Sok funkciót szerettünk volna még implementálni a weboldalunkba, csak idő hiányában és egyéb okokból kifolyólag nem sikerült véghez vinnünk, ezért a továbbfejlesztések közé kerültek.

### Felhasználói fiókok módosítása

Szerettünk volna egy menüpontot, ahol egy űrlapon a felhasználó tudná módosítani a személyes adatai, szállítási címet, elérhetőségeit. Profilkép feltöltésére is szeretnénk biztosítani lehetőséget, amely megjelenne a navigációs sávon és számítógép konfiguráció megosztása esetén a többi konfiguráció között is, a feltöltő neve mellett.

### E-mail

E-mailen keresztül értesítenénk a felhasználót például: regisztrálás esetén egy visszaigazolás lenne szükséges, adatok módosítása, rendelés, visszaküldés esetén egy visszajelzést kapna.

### Ügyfélszolgálat

A legtöbb webáruház rendelkezik saját ügyfélszolgálattal, ahol lehet segítséget kérni, észrevételt, problémát, panaszt jelezni, érdeklődni a megrendelt termékek iránt, visszaküldést lebonyolítani. Ilyen és hasonló műveletek elvégzésére nyújtana megoldást egy webes felület létrehozása.

### Rendelés visszaküldése

Amennyiben a vásárolt eszköz hibás, megsérült szállítás közben, vagy egyéb okból adódóan nem működik megfelelően. Biztosítanánk lehetőséget a termék visszaküldésére ügyfélszolgálaton keresztül, esetleg egy megfelelő menüpont kiválasztásával. Visszaküldés esetén a felhasználó visszakapná a termék összegét vagy egy kupont, amely a következő rendelésnél jelentene kedvezményt.

### Akciók, leértékelések, kuponkód

Az adatbázis fel lett készítve az akciók létrehozására, kezelésére, csak idő hiányában nem került kidolgozásra. Meg lehetne adni százalékkal a kedvezmény mértékét, egy időintervallumot, ami között lenne érvényes vagy pedig egy aktív mezővel is lehetne inaktiválni.

Kuponkód járhatna az első rendelés után, visszaküldött termék után, szezonálisan. Több fajta is lenne, bizonyos mértékű kedvezmény a végösszegből, ingyenes szállítás biztosítása, esetleg több hasonló termék vásárlása esetén alacsonyabb áron megszerezhetővé válnának.

### Fizetési módok

Szeretnénk a későbbiekben bővíteni fizetési lehetőségeinket, bankkártyát szeretnénk elfogadni, esetleg API felhasználásával újabb módszereket biztosítani, például: PayPal, SimplePay.

### Kompatibilitás és további elemek hozzáadása

Konfiguráció összeállítás esetén kompatibilitás ellenőrzésére is lenne lehetőség, adatbázisba mentenénk a szükséges paramétereket, illetve hozzáadhatók lennének nem kötelező elemek, például: perifériák, kiegészítők, kötelezők közül több mint egy elem is.

### Konfiguráció megosztó oldal továbbfejlesztése

Hasznos lenne egy keresés funkció egy felhasználóra vagy a keresett terméket tartalmazó konfigurációk szűkítéséhez, szűrési lehetőség alapján dátumnak, márkának, kompatibilitáshoz szükséges adatoknak megfelelően jelennének meg az oldalon, rendezés történne dátum és népszerűség szerint (like, dislike aránya, megtekintések száma alapján).

# Felhasználói dokumentáció

## Bevezetés

Alkalmazásunk elindításához, és beüzemeléséhez, a már említett, és bemutatott telepítési folyamatot kell végrehajtani. A telepítéshez lapozzon vissza a ’Megfelelő indítás’ bekezdésre. Itt részletesen, és érthetően megtalálható a telepítés folyamata.

## Rendszerkövetelmények

Mesterremekünk webshop lévén, szükség van egy mai modern böngészőre, lehet ez Google Chrome, Mozilla Firefox, Opera és Opera GX is. Szükséges lehet néhány bővítmény telepítése a technológiák miatt, és a JavaScript futtatását is engedélyezni kell. A webes alkalmazás egy adatbázis-kezelő rendszert is használ, amely telepítése szükséges a helyi vagy webszerverre, amiről az alkalmazást szeretnénk futtatni. Futtatáshoz szükséges vezetékes, vagy vezeték nélküli internetkapcsolat. Alkalmazásunkat helyi szerveren fogjuk bemutatni, ezért szükségünk van egy webalkalmazás kiszolgáló helyi alkalmazásra, ami az Apache http szerver, a MySQL adatbázis kezelő rendszer és ezen kívüli más komponensek integrált csomagja, ami jelen esetben a Xampp lesz.

Követelmények a Xampp által biztosított webszerverhez:

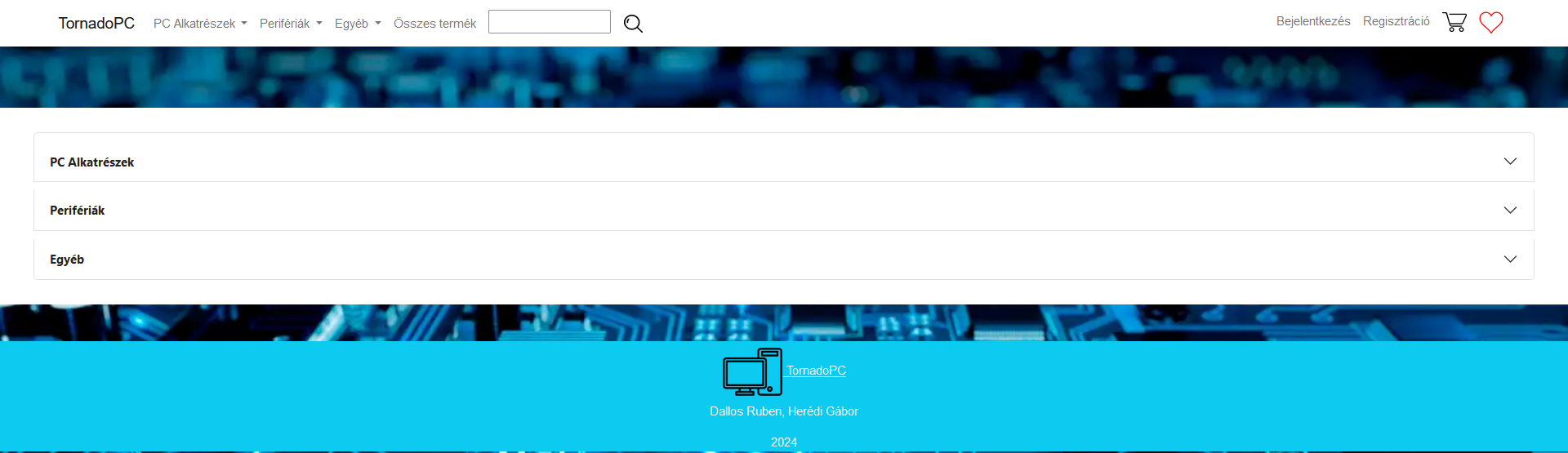
* Windows, Linux vagy Mac OS operációs rendszer
* 5-8 GB tárhely (Minél több új terméket adunk kínálatunkhoz, úgy változhat a

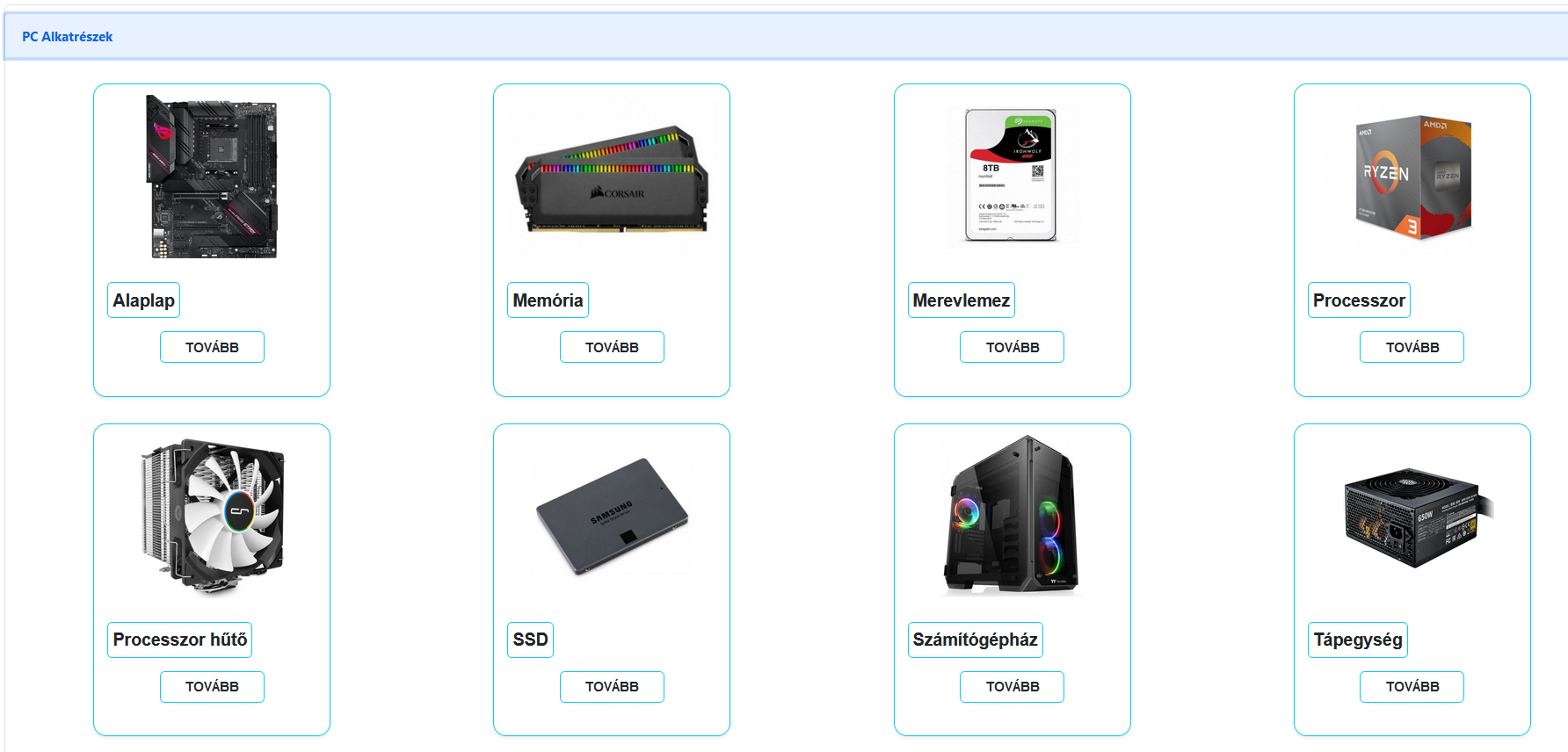
tárhely igény)

* Internetböngésző, javasolt a Mozilla Firefox vagy Google Chrome
* MySQL legfrissebb verziója
* PHP legfrissebb verziója

## Főoldal

A weblapunk megnyitásakor, a főoldal az első dolog, ami a szemünk elé tárul. Itt több fontos dolog is található, mint például a navigációs sor, vagy szakmai néven hívva ’navbar’ is. Ez a navigációs sáv majdnem az összes oldalon elérhető, mert egy fontos része az oldalon való navigálásnak. A főoldalon jelennek meg a termékkategóriák, az adott kategóriára kattintva átugorhatunk az adott kategóriába tartozó termékek megtekintésére.





## Footer

Az összes oldalon található egy lábléc, azaz ’footer’ is. Ezen található a készítők neve, a készítés éve, és egy hivatkozás a főoldalra.

## Navigációs sáv

A navigációs soron található az oldalon való navigáció fő része. Kettő változata van, egy akkor, amikor a felhasználó nincs bejelentkezve, és amikor a felhasználó bejelentkezett.

### Termékek, Kategóriák



A ’navbar’ első része a termékek, és a kategóriák közti navigációra használható. A kategóriáknál lenyíló elemekből lehet választani, vagy az összes termék is megtekinthető, az ’Összes termék’ menüpontra kattintva.



A termékek közötti keresésre is van lehetőség, a ’navbar’ ezen részével. Ez egy keresősáv, amibe a felhasználó beírja egy termék nevét, vagy márkáját, és rányom a nagyító gombra, majd előhozza a keresésnek megfelelő termékeket.

### Regisztráció, Bejelentkezés



A Bejelentkezés vagy Regisztráció menüpontra kattintva, a megfelelő oldalra ugrik. Ez a rész csak akkor jelenik meg, ha a felhasználó nincs bejelentkezve.

### Konfigurációs rész



Erről a részről a számítógépkonfigurációkkal kapcsolatos oldalakra lehet látogatni. A ’Konfigurációk’ menüpontra kattintva a saját, és más felhasználók által megosztott konfigurációk tekinthetők meg, a ’Konfiguráció összerakó’ oldalon pedig saját konfigurációt lehet összeállítani.

### Kosár, kedvencek



Ezen gombokkal juthatunk el a kosárba, vagy a kedvenc termékek oldalra. Ezek csak bejelentkezett felhasználók által használhatók.

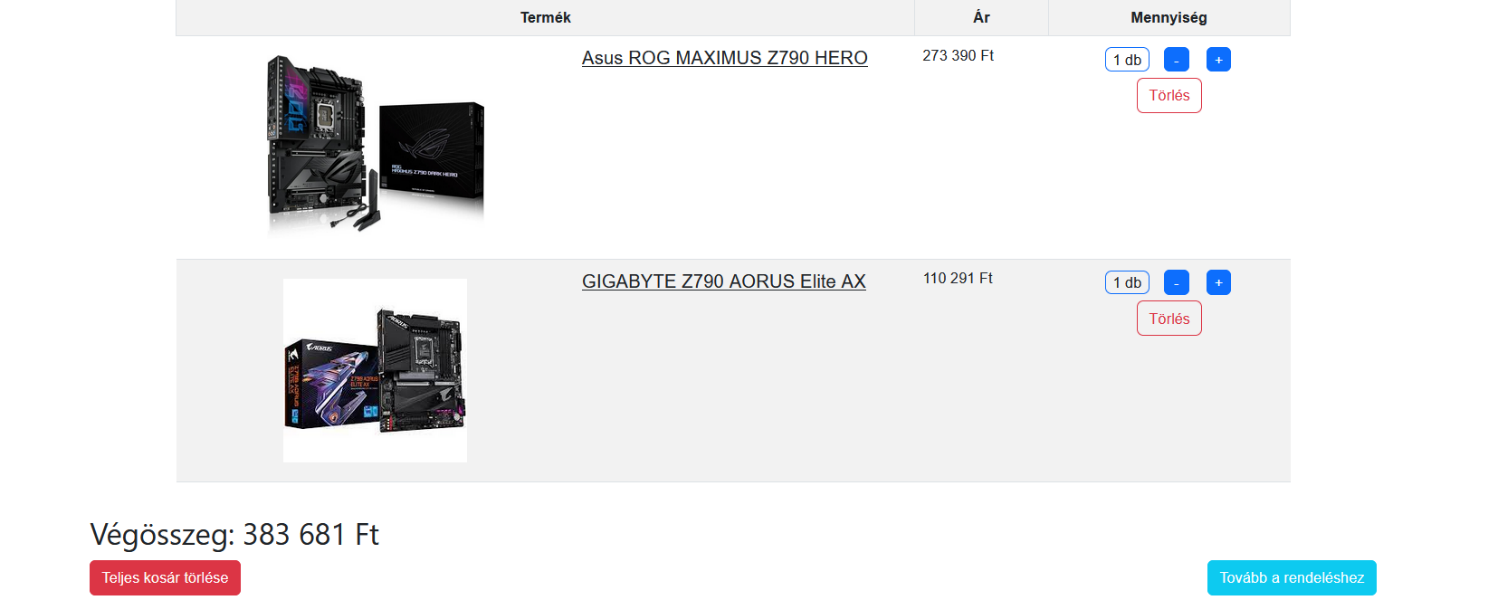
## Egy termék oldala

Egy adott termékre kattintva az oldal átvisz a termék oldalára. Itt látható egy kép a termékről, az ára, illetve az adott termék megadott tulajdonságai is. Kedvencek közé mentheti ezt a terméket és megadható egy darabszám is, amennyire a vásárlónak szüksége lenne, annyit helyezhet a kosárba.

A képen szöveg, Betűtípus, szám, szoftver látható

Automatikusan generált leírás

## Kosár

A kosárban a felhasználó megtekintheti azokat a termékeket, amelyeket hozzáadott a kosárhoz. Itt tud törölni egy-egy terméket és tudja a mennyiséget növelni és csökkenteni. Ennek megfelelően dinamikusan változik a termék ára és a végösszeg is.

A ’Teljes kosár törlése’ gombra kattintva egy kattintással a teljes tartalom törölhető, a ’Tovább a rendelésre’ gombra kattintva, pedig a rendelés oldalra dob át.

## Rendelés

Ezen az oldalon adhatja meg a felhasználó a szállítási címet, telefonszámot és választhatja ki a fizetési módot.

A képen szöveg, Betűtípus, szám, szoftver látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

Sikeres rendelés esetén ez az oldal fogadja a felhasználót, itt megtekintheti a megadott adatait és visszatérhet a főoldalra.

## Rendelési előzmények

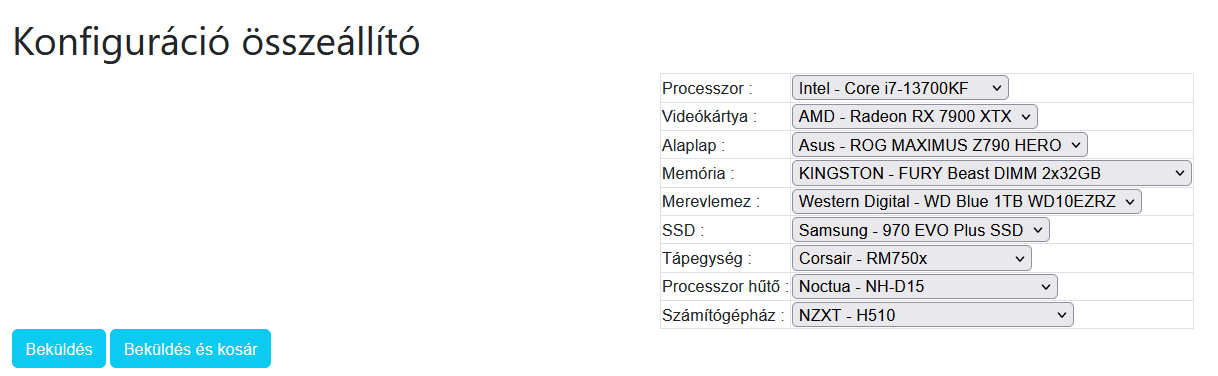
A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás’Rendelési előzmények’ menüpontra kattintva megtekinthetővé válnak a leadott rendelések. Ezekről néhány információ és a rendelt termékek láthatóak.

## Konfiguráció összeállító

A képen szöveg, képernyőkép, Betűtípus, szám látható

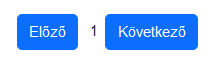
Automatikusan generált leírásEz az oldal csak bejelentkezett felhasználóknak elérhető. Itt tudnak saját számítógépkonfigurációt összeállítani a webshop kínálatában levő hardverelemekből. Az elemeket egy legördülő menüből lehet kiválasztani minden egyes típusnál.

Az összeállítás után kettő lehetősége van a felhasználónak. Hozzáadni a megosztott konfigurációkhoz a ’Beküldés’ gombra kattintva, vagy beküldeni, és a hardverelemeket a kosárhoz adni a ’Beküldés és kosár’ gombra kattintva.

## Konfiguráció értékelő

Ez az oldal csak bejelentkezett felhasználók számára elérhető. Itt lehet megtekinteni a saját és más felhasználók által létrehozott konfigurációkat, és értékelni egy like-al, vagy dislike-al. Egyszerre maximum négy darab konfiguráció jelenhet meg. Mindegyik konfiguráció külön elemként jelenik meg az oldalon. Az oldal jelzi, hogy a felhasználó mivel A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírásértékelte az adott konfigurációt, és annak megfelelően színezi az adott gombot.

Az oldal alján található kettő gomb, amelyek a megjelenített konfigurációk váltására szolgálnak. Ezekkel lehet lapozni az előző, vagy következő oldalra. Középen pedig megjelenik, hogy a felhasználó hányadik oldalon van jelenleg.

# Irodalomjegyzék

## Internetes tartalmak

* **W3schools:** https://www.w3schools.com/
* **Stackoverflow:** https://stackoverflow.com/
* **Bootstrap:** https://getbootstrap.com/
* **PHP:** https://www.php.net/
* **JavaScript:** https://www.javascript.com/
* **jQuery:** https://jquery.com/
* **MySQL:** https://www.mysql.com/

## Könyvek

* Robert C. Martin: **Tiszta kód** (2008)
* David Powers: **PHP Object-Oriented Solutions** (2010)
* Matt Beaumont: **PHP eCommerce** (2010)
* Richard Blum: **PHP, MySQL, & JavaScript All-in-One For Dummies** (2018)
* Jennifer Niederst Robbin: **Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics** (2018)
* Christian Nagel: **Professional C# and .NET - 2021 Edition** (2021)
* Rudolf Pecinovský: **OOP - Learn Object Oriented Thinking and Programming** (2013)

# Melléklet

* **Projektünk GitHub linkje:**

https://github.com/DallosRub/TornadoPC

1. Forrás: https://hu.wikipedia.org/wiki/Visual\_Studio\_Code [↑](#footnote-ref-1)
2. Forrás: https://hu.wikipedia.org/wiki/Microsoft\_Visual\_Studio [↑](#footnote-ref-2)
3. Forrás: https://hu.wikipedia.org/wiki/XAMPP [↑](#footnote-ref-3)
4. Forrás: https://en.wikipedia.org/wiki/Microsoft\_Teams [↑](#footnote-ref-4)
5. Forrás: https://hu.wikipedia.org/wiki/GitHub [↑](#footnote-ref-5)
6. Forrás: https://hu.wikipedia.org/wiki/HTML [↑](#footnote-ref-6)
7. Forrás: https://hu.wikipedia.org/wiki/CSS [↑](#footnote-ref-7)
8. Forrás: https://hu.wikipedia.org/wiki/PHP [↑](#footnote-ref-8)
9. Forrás: https://hu.wikipedia.org/wiki/JavaScript [↑](#footnote-ref-9)
10. Forrás: https://hu.wikipedia.org/wiki/JQuery [↑](#footnote-ref-10)
11. Forrás: https://hu.wikipedia.org/wiki/Ajax\_(programoz%C3%A1s) [↑](#footnote-ref-11)
12. Forrás: https://hu.wikipedia.org/wiki/REST [↑](#footnote-ref-12)
13. Forrás: https://hu.wikipedia.org/wiki/MySQL [↑](#footnote-ref-13)
14. Forrás https://hu.wikipedia.org/wiki/C\_Sharp [↑](#footnote-ref-14)