



**Debreceni SZC Mechwart András Gépipari és Informatikai Technikum**

# **Debreceni jelmez webshop honlapja**

Készítette: Molnár Péter 14EC  
Győri Szabolcs 14EC  
Vajna Péter 14EC

Debrecen, 2025

## Tartalomjegyzék

<b>1.Bevezetés, a téma ismertetése, témaválasztás indoklása, szakmai célkitűzés .....</b>	<b>3</b>
<b>2. Fejlesztői dokumentáció.....</b>	<b>4</b>
2.1 Fejlesztőkörnyezet .....	4
2.2 Adatszerkezet .....	6
2.3 Algoritmusok és kódszerkezetek .....	11
2.4 Tesztelés .....	14
2.5 Fejlesztési lehetőségek.....	17
<b>3. Felhasználói dokumentáció .....</b>	<b>18</b>
3.1 Program célja és funkciói .....	18
3.2 Hardver és szoftver követelmények.....	18
3.3 Telepítés és indítás .....	18
3.4 A program használata.....	19
3.5 Szabadon elérhető és regisztrációhoz kötött funkciók.....	19
3.6 Szemléltetés .....	19
3.7 Hibajelzések és figyelmeztető üzenetek felhasználó a tevékenysége során .....	23
3.8 Admin felhasználó lehetőségei .....	23
3.9 Információkérés lehetősége. ....	23
<b>4. Összefoglaló.....</b>	<b>24</b>
4.1 Bevezetés .....	24
4.2 Technikai értékelés .....	24
4.3 Végeredmény értékelése .....	24
<b>4. Irodalom jegyzék .....</b>	<b>25</b>

## **1. Bevezetés, a téma ismertetése, témaválasztás indoklása, szakmai célkitűzés**

Az elkészült projekt egy debreceni fiktív jelmezkölcsönző üzlet weboldalát mutatja be, amelyet a vállalkozás online jelenlétének növelésére terveztünk. Elképzelésünk szerint jelenleg a kölcsönző csak a közösségi médiában van jelen, azonban az információk elérése nehézkes, a kínálat átláthatósága korlátozott, ami csökkenti a felhasználói élményt. Ennek megoldására egy átfogó, könnyen kezelhető és vizuálisan vonzó weboldalt hoztunk létre, amely segíti az ügyfeleket a jelmezek böngészésében, foglalásában, valamint megkönnyíti a kommunikációt és az üzleti folyamatokat.

A projekt témaválasztását indokolja az üzlet jelenlegi digitális jelenlétének hiányosságai, valamint az egyre növekvő igény egy jól strukturált, felhasználóbarát online felületre, mint ahogy szinte minden növekvő üzlet esetén. Személyes motivációnk között szerepel a kreatív önkifejezés és a tematikus rendezvények népszerűsítése, amelyet egy modern platform támogatásával szeretnénk elősegíteni. Lehetőséget adhat akár egy színházi együttműködésre.

Az elkészült weboldal a szoftverfejlesztő és tesztelő képzés részeként készült vizsgadokumentációként szolgál, amelynek célja a webfejlesztés teljes folyamatának bemutatása, a tervezéstől kezdve a megvalósításig és tesztelésig. Emellett részletesen dokumentáljuk a fejlesztés során felmerült kihívásokat és azok megoldásait, valamint az alkalmazott technológiákat és módszereket.

Szakmai célkitűzésünk egy modern, reszponzív és felhasználóbarát weboldal létrehozása, amely támogatja az üzlet online növekedését és segít elérni a potenciális ügyfeleket. Bár a projekt fiktív jellegű, úgy terveztük, hogy akár egy valódi vállalkozás is profitálhatna belőle, vagy alapul szolgálhatna egy későbbi fejlesztéshez.

Ez a vizsgaremek dokumentáció bemutatja a weboldal tervezési és kivitelezési folyamatát, valamint annak potenciális hasznosulási lehetőségeit a vállalkozás számára, tárgyalva az alkalmazott technológiákat és a fejlesztési lépéseket.

## 2. Fejlesztői dokumentáció

### 2.1 Fejlesztőkörnyezet

A jelmez webshop elkészítésének fő célja egy olyan digitális felület létrehozása volt, amely megkönnyíti a jelmezek vásárlását, miközben javítja a vállalkozás online jelenlétét. A hagyományos jelmezkölcsönzők és webáruházak gyakran széttagoltan működnek: az információk nehezen elérhetők, a rendelési folyamat körülményes, és sok esetben nincs hatékony keresési vagy kategorizálási rendszer

Ez a projekt modern, felhasználóbarát és könnyen kezelhető, amely ötvözi a gyors keresést, az átlátható termékkategóriákat és a könnyű rendelési folyamatot. A hagyományos megoldásokkal szemben az új webshop több szempontból is előremutató:

***Interaktív keresési rendszer*** – Egy fejlett szűrővel és keresőmotorral rendelkezik, amely segít a felhasználóknak gyorsan megtalálni a nekik megfelelő jelmezt.

***Reszponzív és mobilbarát*** – A webshop minden eszközön optimálisan működik, így a vásárlók akár telefonról is kényelmesen böngészhetnek és rendelhetnek.

***Fiktív, de valós alkalmazásra alkalmas koncepció*** – Bár a projekt jelenleg fiktívként van kezelve, de funkcionalitása miatt akár egy valós vállalkozás is könnyedén alkalmazhatná és továbbfejleszthetné.

A program cél a tehát nem csupán egy webshop létrehozása, hanem egy üzleti lehetőség modellezése, amely bemutatja, miként lehet egy hagyományos szolgáltatást digitális eszközökkel hatékonyabbá és vonzóbbá tenni.

**A weblap fejlesztése során a következő eszközöket és technológiákat használtuk:**

- Hardver: Számítógép, laptop
- Programozási technológiák: HTML, CSS, JavaScript, Python, Mysql, Django
- Szoftverek: Visual Studio Code , Python alkalmazás, Webböngésző (Google Chrome)

Hardver: PC erőforrások:

**Minimum rendszerkövetelmény:**

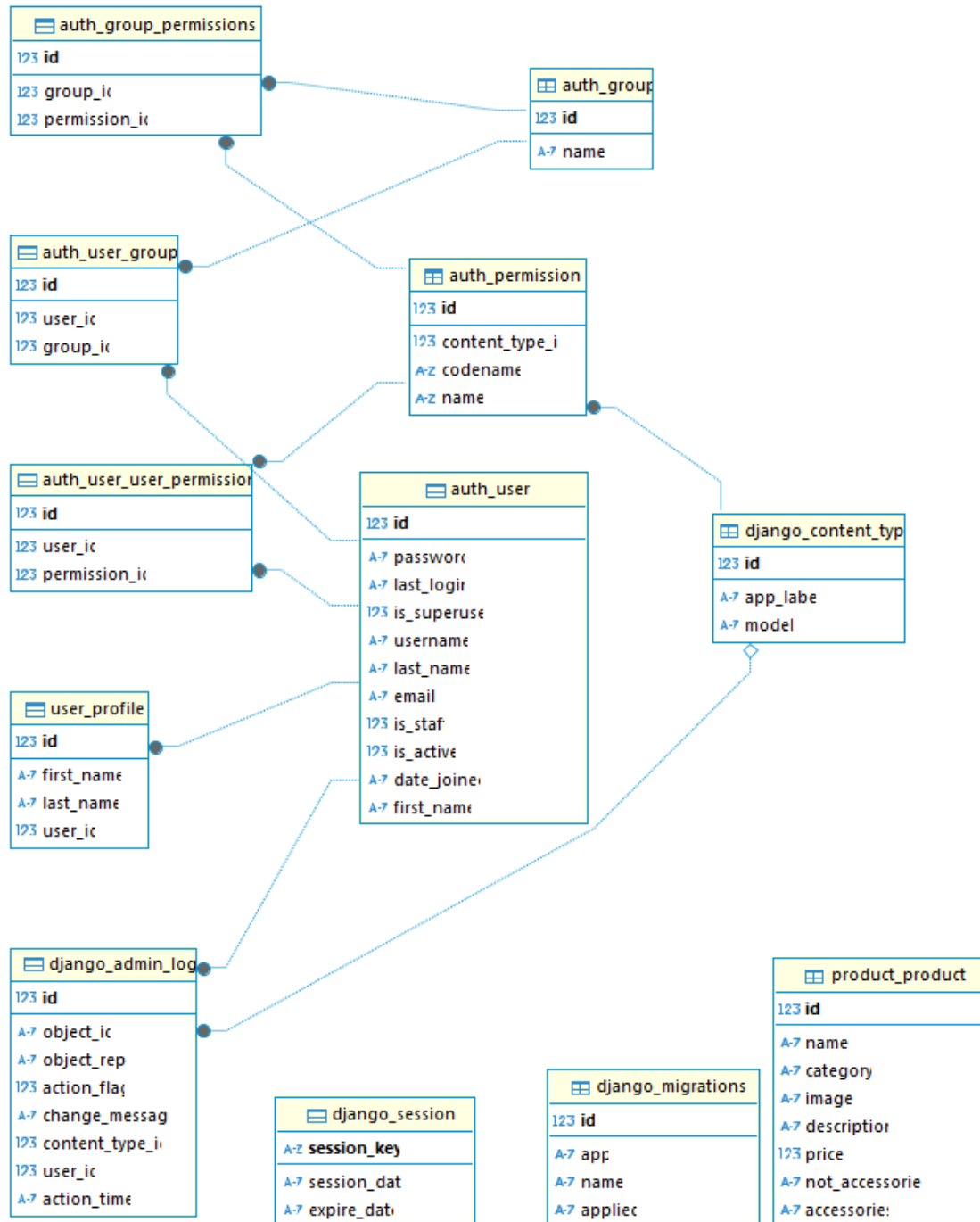
- **Processzor:** 1 gigahertzes (GHz) vagy gyorsabb processzor vagy SoC
- **RAM:** 1 gigabájt (GB) 32 bites rendszerhez vagy 2 GB 64 bites rendszerhez
- **Merevlemez-terület:** 16 GB 32 bites operációs rendszerhez vagy 20 GB 64 bites operációs rendszerhez
- **Videokártya:** A DirectX 9-es vagy újabb verziójával kompatibilis, WDDM 1.0 illesztőprogrammal
- **Kijelző:** 800x600

**Ajánlott rendszerkövetelmény:**

- **Processzor:** 1 GHz-es vagy gyorsabb, legalább 2 magos, kompatibilis, 64 bites processzor vagy egylapkás rendszer (SoC)
- **Memória:** 4 GB RAM
- **Tárhely:** legalább 64 GB
- **Videokártya:** DirectX 12-kompatibilis vagy újabb, WDDM 2.0 típusú illesztőprogrammal
- **Képernyő:** Legalább 9" képátlójú, színcsatornánként 8 bites, nagy felbontású (720p) kijelző

## 2.2 Adatszerkezet

Az alkalmazás adatszerkezete az alábbiak szerint alakult:



- Adatbázis: SQLite

## Tables (13)

Name	Type	Schema
<b>auth_group</b>		CREATE TABLE "auth_group" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(150) NOT NULL UNIQUE)
id	integer	"id" integer NOT NULL
name	varchar(150)	"name" varchar(150) NOT NULL UNIQUE
<b>auth_group_permissions</b>		CREATE TABLE "auth_group_permissions" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "group_id" integer NOT NULL REFERENCES "auth_group" ("id") DEFERRABLE INITIALLY DEFERRED, "permission_id" integer NOT NULL REFERENCES "auth_permission" ("id") DEFERRABLE INITIALLY DEFERRED)
id	integer	"id" integer NOT NULL
group_id	integer	"group_id" integer NOT NULL
permission_id	integer	"permission_id" integer NOT NULL
<b>auth_permission</b>		CREATE TABLE "auth_permission" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "content_type_id" integer NOT NULL REFERENCES "django_content_type" ("id") DEFERRABLE INITIALLY DEFERRED, "codename" varchar(100) NOT NULL, "name" varchar(255) NOT NULL)
id	integer	"id" integer NOT NULL
content_type_id	integer	"content_type_id" integer NOT NULL
codename	varchar(100)	"codename" varchar(100) NOT NULL
name	varchar(255)	"name" varchar(255) NOT NULL
<b>auth_user</b>		CREATE TABLE "auth_user" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "password" varchar(128) NOT NULL, "last_login" datetime NULL, "is_superuser" bool NOT NULL, "username" varchar(150) NOT NULL UNIQUE, "last_name" varchar(150) NOT NULL, "email" varchar(254) NOT NULL, "is_staff" bool NOT NULL, "is_active" bool NOT NULL, "date_joined" datetime NOT NULL, "first_name" varchar(150) NOT NULL)
id	integer	"id" integer NOT NULL
password	varchar(128)	"password" varchar(128) NOT NULL
last_login	datetime	"last_login" datetime
is_superuser	bool	"is_superuser" bool NOT NULL
username	varchar(150)	"username" varchar(150) NOT NULL UNIQUE
last_name	varchar(150)	"last_name" varchar(150) NOT NULL
email	varchar(254)	"email" varchar(254) NOT NULL
is_staff	bool	"is_staff" bool NOT NULL
is_active	bool	"is_active" bool NOT NULL
date_joined	datetime	"date_joined" datetime NOT NULL
first_name	varchar(150)	"first_name" varchar(150) NOT NULL
<b>auth_user_groups</b>		CREATE TABLE "auth_user_groups" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "user_id" integer NOT NULL REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED, "group_id" integer NOT NULL REFERENCES "auth_group" ("id") DEFERRABLE INITIALLY DEFERRED)



Name	Type	Schema
id	integer	"id" integer NOT NULL
user_id	integer	"user_id" integer NOT NULL
group_id	integer	"group_id" integer NOT NULL
<b>auth_user_user_permissions</b>		CREATE TABLE "auth_user_user_permissions" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "user_id" integer NOT NULL REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED, "permission_id" integer NOT NULL REFERENCES "auth_permission" ("id") DEFERRABLE INITIALLY DEFERRED)
id	integer	"id" integer NOT NULL
user_id	integer	"user_id" integer NOT NULL
permission_id	integer	"permission_id" integer NOT NULL
<b>django_admin_log</b>		CREATE TABLE "django_admin_log" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "object_id" text NULL, "object_repr" varchar(200) NOT NULL, "action_flag" smallint unsigned NOT NULL CHECK ("action_flag" >= 0), "change_message" text NOT NULL, "content_type_id" integer NULL REFERENCES "django_content_type" ("id") DEFERRABLE INITIALLY DEFERRED, "user_id" integer NOT NULL REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED, "action_time" datetime NOT NULL)
id	integer	"id" integer NOT NULL
object_id	text	"object_id" text
object_repr	varchar(200)	"object_repr" varchar(200) NOT NULL
action_flag	smallint unsigned	"action_flag" smallint unsigned NOT NULL CHECK("action_flag" >= 0)
change_message	text	"change_message" text NOT NULL
content_type_id	integer	"content_type_id" integer
user_id	integer	"user_id" integer NOT NULL
action_time	datetime	"action_time" datetime NOT NULL
<b>django_content_type</b>		CREATE TABLE "django_content_type" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "app_label" varchar(100) NOT NULL, "model" varchar(100) NOT NULL)
id	integer	"id" integer NOT NULL
app_label	varchar(100)	"app_label" varchar(100) NOT NULL
model	varchar(100)	"model" varchar(100) NOT NULL
<b>django_migrations</b>		CREATE TABLE "django_migrations" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "app" varchar(255) NOT NULL, "name" varchar(255) NOT NULL, "applied" datetime NOT NULL)
id	integer	"id" integer NOT NULL
app	varchar(255)	"app" varchar(255) NOT NULL
name	varchar(255)	"name" varchar(255) NOT NULL
applied	datetime	"applied" datetime NOT NULL
<b>django_session</b>		CREATE TABLE "django_session" ("session_key" varchar(40) NOT NULL PRIMARY KEY, "session_data" text NOT NULL, "expire_date" datetime NOT NULL)
session_key	varchar(40)	"session_key" varchar(40) NOT NULL



Name	Type	Schema
session_data	text	"session_data" text NOT NULL
expire_date	datetime	"expire_date" datetime NOT NULL
<b>product_product</b>		CREATE TABLE "product_product" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(255) NOT NULL, "category" varchar(10) NOT NULL, "image" varchar(100) NOT NULL, "description" varchar(255) NOT NULL, "price" integer NOT NULL, "not_accessories" text NOT NULL CHECK ((JSON_VALID("not_accessories") OR "not_accessories" IS NULL)), "accessories" text NOT NULL CHECK ((JSON_VALID("accessories") OR "accessories" IS NULL)))
id	integer	"id" integer NOT NULL
name	varchar(255)	"name" varchar(255) NOT NULL
category	varchar(10)	"category" varchar(10) NOT NULL
image	varchar(100)	"image" varchar(100) NOT NULL
description	varchar(255)	"description" varchar(255) NOT NULL
price	integer	"price" integer NOT NULL
not_accessories	text	"not_accessories" text NOT NULL CHECK((JSON_VALID("not_accessories") OR "not_accessories" IS NULL))
accessories	text	"accessories" text NOT NULL CHECK((JSON_VALID("accessories") OR "accessories" IS NULL))
<b>sqlite_sequence</b>		CREATE TABLE sqlite_sequence(name,seq)
name		"name"
seq		"seq"
<b>user_profile</b>		CREATE TABLE "user_profile" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "first_name" varchar(50) NOT NULL, "last_name" varchar(50) NOT NULL, "user_id" integer NOT NULL UNIQUE REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED)
id	integer	"id" integer NOT NULL
first_name	varchar(50)	"first_name" varchar(50) NOT NULL
last_name	varchar(50)	"last_name" varchar(50) NOT NULL
user_id	integer	"user_id" integer NOT NULL UNIQUE

## Indices (15)

Name	Type	Schema
<b>auth_group_permissions_group_id_b120cbf9</b>		CREATE INDEX "auth_group_permissions_group_id_b120cbf9" ON "auth_group_permissions" ("group_id")
group_id		"group_id"
<b>auth_group_permissions_group_id_permission_id_0cd325b0_uniq</b>		CREATE UNIQUE INDEX "auth_group_permissions_group_id_permission_id_0cd325b0_uniq" ON "auth_group_permissions" ("group_id", "permission_id")
group_id		"group_id"
permission_id		"permission_id"
<b>auth_group_permissions</b>		CREATE INDEX "auth_group_permissions_permission_id_84c5c92e" ON

Name	Type	Schema
<b>_permission_id_84c5c92e</b>		"auth_group_permissions" ("permission_id")
permission_id		"permission_id"
<b>auth_permission_content_type_id_2f476e4b</b>	CREATE INDEX	"auth_permission_content_type_id_2f476e4b" ON "auth_permission" ("content_type_id")
content_type_id		"content_type_id"
<b>auth_permission_content_type_id_codename_01ab375a_uniq</b>	CREATE UNIQUE INDEX	"auth_permission_content_type_id_codename_01ab375a_uniq" ON "auth_permission" ("content_type_id", "codename")
content_type_id		"content_type_id"
codename		"codename"
<b>auth_user_groups_group_id_97559544</b>	CREATE INDEX	"auth_user_groups_group_id_97559544" ON "auth_user_groups" ("group_id")
group_id		"group_id"
<b>auth_user_groups_user_id_6a12ed8b</b>	CREATE INDEX	"auth_user_groups_user_id_6a12ed8b" ON "auth_user_groups" ("user_id")
user_id		"user_id"
<b>auth_user_groups_user_id_group_id_94350c0c_uniq</b>	CREATE UNIQUE INDEX	"auth_user_groups_user_id_group_id_94350c0c_uniq" ON "auth_user_groups" ("user_id", "group_id")
user_id		"user_id"
group_id		"group_id"
<b>auth_user_user_permissions_permission_id_1fb5f2c</b>	CREATE INDEX	"auth_user_user_permissions_permission_id_1fb5f2c" ON "auth_user_user_permissions" ("permission_id")
permission_id		"permission_id"
<b>auth_user_user_permissions_user_id_a95ead1b</b>	CREATE INDEX	"auth_user_user_permissions_user_id_a95ead1b" ON "auth_user_user_permissions" ("user_id")
user_id		"user_id"
<b>auth_user_user_permissions_user_id_permission_id_14a6b632_uniq</b>	CREATE UNIQUE INDEX	"auth_user_user_permissions_user_id_permission_id_14a6b632_uniq" ON "auth_user_user_permissions" ("user_id", "permission_id")
user_id		"user_id"
permission_id		"permission_id"
<b>django_admin_log_content_type_id_c4bce8eb</b>	CREATE INDEX	"django_admin_log_content_type_id_c4bce8eb" ON "django_admin_log" ("content_type_id")
content_type_id		"content_type_id"
<b>django_admin_log_user_id_c564eba6</b>	CREATE INDEX	"django_admin_log_user_id_c564eba6" ON "django_admin_log" ("user_id")
user_id		"user_id"
<b>django_content_type_app_label_model_76bd3d3b_uniq</b>	CREATE UNIQUE INDEX	"django_content_type_app_label_model_76bd3d3b_uniq" ON "django_content_type" ("app_label", "model")
app_label		"app_label"



Name	Type	Schema
model		"model"
<b>django_session_expire_date_a5c62663</b>		CREATE INDEX "django_session_expire_date_a5c62663" ON "django_session" ("expire_date")
expire_date		"expire_date"

## Views (0)

Name	Type	Schema
------	------	--------

## Triggers (0)

Name	Type	Schema
------	------	--------

### Főbb változók:

- **auth\_user** (Felhasználói adatok)
- **user\_profile** (Felhasználói profilok),
- **product\_product** (Termékek)
- **django\_session** (Munkamenet adatok)

### Táblák közötti kapcsolatok

- **auth\_user** és **user\_profile** kapcsolata → Minden felhasználónak lehet saját profilja (user\_id külső kulcs az auth\_user-hoz)
- **product\_product** önállóan létező entitás, amely külön termékeket tárol
- **django\_session** a felhasználók munkamenetét kezeli

## 2.3 Algoritmusok és kódszerkezetek

Az alkalmazás egyedi és érdekesebb algoritmusai közé tartoznak:

- **Keresési algoritmus** – A productCards.jsx fájl tartalmaz egy méretválasztási logikát, amely segít kiválasztani a megfelelő méretet a kategória alapján.
- **Adatkezelési algoritmus** – A kosárba helyezés funkció ellenőrzi, hogy egy adott méret már szerepel-e a kosárban (isAlreadyInCart változó), így elkerüli a duplikációkat.

Megvalósítása:

```
ProductCard.jsx M X
frontend > src > components > ProductCard.jsx > ProductCard > confirmAddToCart > isAlreadyInCart
1 import React, { useState } from 'react';
2 import { Link } from 'react-router-dom';
3
4 const ProductCard = ({ product }) => {
5   const [showSizePrompt, setShowSizePrompt] = useState(false);
6   const [selectedSize, setSelectedSize] = useState('');
7
8   const sizeOptions = {
9     men: ['S', 'M', 'L', 'XL', 'XXL', '3XL'],
10    women: ['S', 'M', 'L', 'XL'],
11    children: [
12      '99-104', '105-110', '111-116', '117-122',
13      '123-128', '129-134', '135-144', '145-150'
14    ]
15  };
16
17  const availableSizes = sizeOptions[product.category] || [];
18
19  const handleRentClick = () => {
20    setShowSizePrompt(true);
21  };
22
23  const confirmAddToCart = () => {
24    if (!selectedSize) {
25      alert('Kérjük válassz méretet!');
26      return;
27    }
28
29    const cart = JSON.parse(localStorage.getItem('cart')) || [];
30    const isAlreadyInCart = cart.some(item => item.id === product.id && item.size === selectedSize);
31    if (isAlreadyInCart) {
32      alert('Ez a méret már a kosárban van!');
33      return;
34    }
35
36    const updatedCart = [...cart, { ...product, size: selectedSize }];
37    localStorage.setItem('cart', JSON.stringify(updatedCart));
38    alert(`${product.name} (${selectedSize}) hozzáadva a kosárhoz!`);
39    setShowSizePrompt(false);
40    setSelectedSize('');
41  };
42
43  const formatNumber = (num) => {
44    const parsedNum = parseFloat(num);
```

- **Felhasználói azonosítás és jogosultságkezelés** – A *Header.jsx* fájlban a bejelentkezés-kezelés és kijelentkezés funkció biztosítja a felhasználók közötti jogosultságokat, tokenek kezelésével.
- **Formázási algoritmus** – A *formatNumber* függvény átalakítja az árakat megfelelő formátumra (ez egy egyszerű számformázási algoritmus).

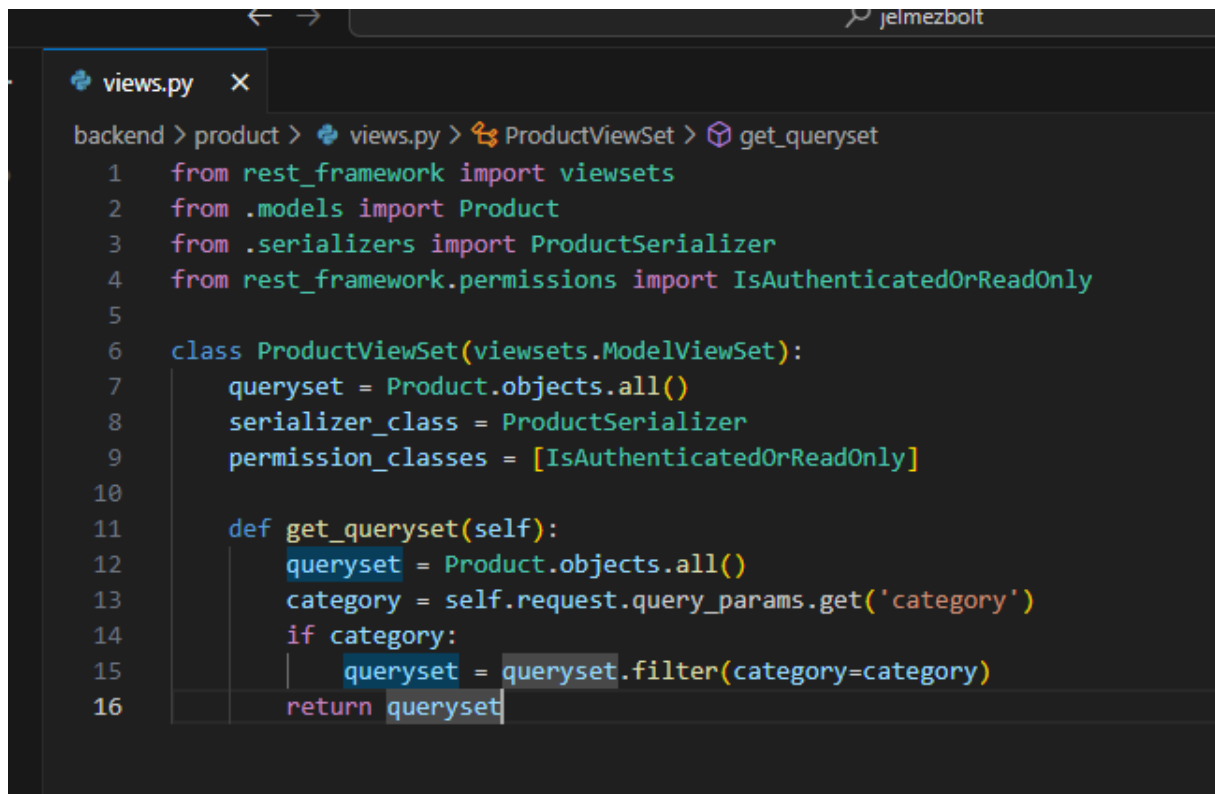
## Adatbázis-kezelési algoritmus (models.py)

```
models.py X
backend > product > models.py > ...
1  from django.db import models
2
3  class Product(models.Model):
4      CATEGORY_CHOICES = [
5          ('children', 'Children'),
6          ('men', 'Men'),
7          ('women', 'Women')
8      ]
9
10     id = models.AutoField(primary_key=True)
11     name = models.CharField(max_length=255)
12     description = models.CharField(max_length=255, blank=True, default='')
13     accessories = models.JSONField(blank=True, default=list)
14     not_accessories = models.JSONField(blank=True, default=list)
15     price = models.IntegerField(default=0)
16     category = models.CharField(max_length=10, choices=CATEGORY_CHOICES)
17     image = models.ImageField(upload_to='product_images/')
18
19     def __str__(self):
20         return self.name
21
```

Ebben a fájlban a **Django ORM** segítségével definiáltad a **Product** osztályt, amely az adatbázisban tárolt jelmezeket reprezentálja.

- **Adatszerkezetek és típusok kezelése** – Az egyes mezők (name, price, category, stb.) megkapják a megfelelő adattípust (CharField, IntegerField, JSONField).
- **Kategória-alapú adattárolás** – A **CATEGORY\_CHOICES** lista biztosítja, hogy csak előre meghatározott kategóriák kerüljenek az adatbázisba.
- **Adatbázis-lekérdezéshez optimalizált struktúra** – Az **id** mező automatikusan generált elsődleges kulcs (AutoField), ami hatékony adatkeresést tesz lehetővé.

## Lekérdezési algoritmus (views.py - get\_queryset)

A screenshot of a code editor with a dark theme. The editor shows a file named 'views.py' with a breadcrumb path: 'backend > product > views.py > ProductViewSet > get\_queryset'. The code is as follows:

```
1 from rest_framework import viewsets
2 from .models import Product
3 from .serializers import ProductSerializer
4 from rest_framework.permissions import IsAuthenticatedOrReadOnly
5
6 class ProductViewSet(viewsets.ModelViewSet):
7     queryset = Product.objects.all()
8     serializer_class = ProductSerializer
9     permission_classes = [IsAuthenticatedOrReadOnly]
10
11     def get_queryset(self):
12         queryset = Product.objects.all()
13         category = self.request.query_params.get('category')
14         if category:
15             queryset = queryset.filter(category=category)
16         return queryset
```

Ez a függvény a lekérdezések kezelésére szolgál, és biztosítja, hogy dinamikusan lehessen szűrni a termékeket kategória alapján.

- **Alaplekérdezés** (`Product.objects.all()`) – Minden termék lekérdezése az adatbázisból.
- **Dinamikus szűrés** (`queryset.filter(category=category)`) – Ha a felhasználó egy adott kategóriát szeretne látni, akkor az adatbázisban csak azokat az elemeket listázza, amelyek megfelelnek a kérésnek.
- **Hatékonyság növelése** – Az adatok előszűrése csökkenti az adatbázis terhelését és gyorsabb betöltést eredményez.

## 2.4 Tesztelés

Az alkalmazás tesztelése során a következő körülményeket vizsgáltuk:

- Különböző böngészők (Chrome, Firefox, Safari)
- Reszponzivitás teszt
- Adatbázis hibakezelés
- Sebesség teszt



A Debreceni jelmez webshop honlapja hibát tapasztaltunk két esetben is.

## 1. Hiba: Gombok nem voltak egy sorban a terméklistán

### Leírás

Az egyes termékek alatt található gombok (például "Kosárba", "Részletek") nem voltak megfelelően igazítva, és egyes esetekben **eltérő magasságban** helyezkedtek el az egyes termékeknél.

### Lehetséges okok

- A gombok **különböző padding és margin értékekkel** rendelkeztek, ezért nem illeszkedtek egy vonalba.
- A termékkártyák eltérő magasságúak voltak, ami okozta az elrendezés elcsúszását.
- **Flexbox vagy Grid hiánya** a komponensben (ProductList.jsx vagy ProductCard.jsx).
- **Nem megfelelő Tailwind vagy CSS beállítások** (például inline-block helyett flex kellett volna).

### Megoldás

✓ Az összes gomb konténerét (div vagy button elemeket) **Flexbox-szal vagy Grid-el** igazították egy sorba:

```
/* Gombok és méretválasztó legalul */  
<div className="mt-auto pt-4">  
  <div className="flex space-x-4">  
    <Link to={`/${product}/${product.id}`}>  
      <button className="px-4 py-2 bg-blue-500 text-white font-semibold rounded-lg shadow-md hover:bg-blue-600 transition">  
        Részletek  
      </button>  
    </Link>  
    <button  
      onClick={handleRentClick}  
      className="px-4 py-2 bg-green-500 text-white font-semibold rounded-lg shadow-md hover:bg-green-600 transition">  
      >  
      Kosárba  
    </button>  
  </div>  
</div>
```

✓ A kártyák magasságát egységesítették (min-h-[200px]).

✓ A CSS-ben eltávolították az egyedi margin és padding értékeket, hogy ne legyen eltérés.

## 2. Hiba: A kategória szűrés nem működött megfelelően

### Leírás

A ProductPage.jsx-ben található kategória szűrés (<select> menü) nem szűrte megfelelően a termékeket, amikor a "Mind", "Férfi", "Női", "Gyerek" opciók között váltott a felhasználó.

### Hiba oka

- A **fetchProducts API hívás** nem küldött megfelelő paramétereket az API-nak.
- A feltétel (`category !== 'all' ? { category } : {}`) nem kezelte jól az **összes termék listázását**, amikor a "Mind" volt kiválasztva.
- A **useEffect** nem frissítette helyesen az adatokat minden kategóriaváltáskor.

### Megoldás

✓ **API hívás logikájának módosítása**, hogy "Mind" esetén minden termék megjelenjen:

```
useEffect(() => {
  const fetchProducts = async () => {
    setLoading(true);
    try {
      const response = await api.get('/products/', {
        params: category !== 'all' ? { category } : {}
      });

      setProducts(response.data.results || response.data);
    } catch (error) {
      console.error('Hiba a termékek betöltésekor:', error);
      setProducts([]);
    } finally {
      setLoading(false);
    }
  };
});
```

✓ **UseEffect frissítése**, hogy minden kategóriaváltáskor biztosan frissüljön:

```
useEffect(() => {
  fetchProducts();
}, [category]);
```

## 2.5 Fejlesztési lehetőségek

### **Az alkalmazás továbbfejlesztésének néhány lehetséges területe:**

- Felhasználói élmény javítása: felhasználói felület továbbfejlesztése és optimalizálása például: Vélemények és értékelések a termékekhez
- További funkciók hozzáadása: Új funkciók bevezetése, például kosár automatikus mentése, promóciók kezelése stb.
- Teljesítményoptimalizálás: Az adatbázis lekérdezések és az alkalmazás teljesítményének optimalizálása.
- Kiterjesztett támogatás: Több nyelv támogatása, különböző fizetési módok bevezetése stb.

### **A jelen működés javítása tökéletesítése:**

- Oldal frissítése: A jelenlegi kosár csak a frontend állapotban tárolja az adatokat, így ha a felhasználó frissíti az oldalt, a kosár kiürül. – Ez a fejlesztés többek között például a Backend API endpoint hozzáadását igényli.
- Kosár állapot változása: A kosár piktogram, ha van valamilyen termék a kosárban nem változik. Ott is célszerű lenne jelölni, hogy van tétel a kosárba és hogy mennyi tétel van a kosárban.
- Szabadon elérhető és regisztrációhoz kötött funkcióban jelenleg nincs eltérés. Érdeemes lehet például az alábbi logikát érvényesíteni:  
Regisztráció nélkül – Termékek böngészése, kategóriák használata.  
Regisztráció után – Kosár mentése, vásárlás, kedvencek listázása.

### 3. Felhasználói dokumentáció

#### 3.1 Program célja és funkciói

A **Jelmez Webshop** egy online áruház, amelyben jelmezeket vásárolhatnak a felhasználók. A rendszer lehetővé teszi a termékek böngészését, kosárba helyezését, kategória szerinti szűrését és vásárlását. A webshop célja, hogy gyors és egyszerű módon tegye elérhetővé a jelmezek széles választékát.

#### 3.2 Hardver és szoftver követelmények

A program használatához szükséges:

**Hardver:**

- Asztali számítógép, laptop mobil vagy tablet eszköz
- Internetkapcsolat
- Kijelző (min. 720p felbontás javasolt)

**Szoftver:**

- Modern böngésző (Chrome, Edge, Firefox, Safari stb.)
- Operációs rendszer: Windows, macOS, Linux, Android, iOS

#### 3.3 Telepítés és indítás

A webshop egy webalkalmazás, így végeredményében **nincs szükség telepítésre**, melynek használta az alábbi:

1. **Nyissa meg a böngészőt.**
2. **Lépjen be a webshop URL-jére:** [www.jelmezwebshop.hu](http://www.jelmezwebshop.hu)
3. **Regisztráció nélkül böngészhet**, de vásárláshoz **bejelentkezés szükséges**.

Jelenlegi formájában az alábbiak szerint indítható el:

**Indítás:**

1. Projekt klónozása: (terminál) `cd C:\Users.... --> git clone https://github.com/petya1/jelmezbolt.git`

2. Backend beállítása és indítása (terminál) Navigáljunk a backend könyvtárba: --> pip install -r requirements.txt --> python manage.py runserver (<http://127.0.0.1:8000/>)
3. Frontend beállítása és indítása (terminál) Navigáljunk a frontend könyvtárba --> npm install --> npm run dev --> <http://localhost:5173/> címenen elérhető a web

### 3.4 A program használata

**Termékek böngészése** – A kategóriák és keresőmező segítségével.

**Kosárba helyezés** – Egy termék kiválasztásakor a „Kosárba” gombbal hozzáadható.

**Szűrés** – A kategóriák (férfi, női, gyerek, mind) alapján történő szűrés.

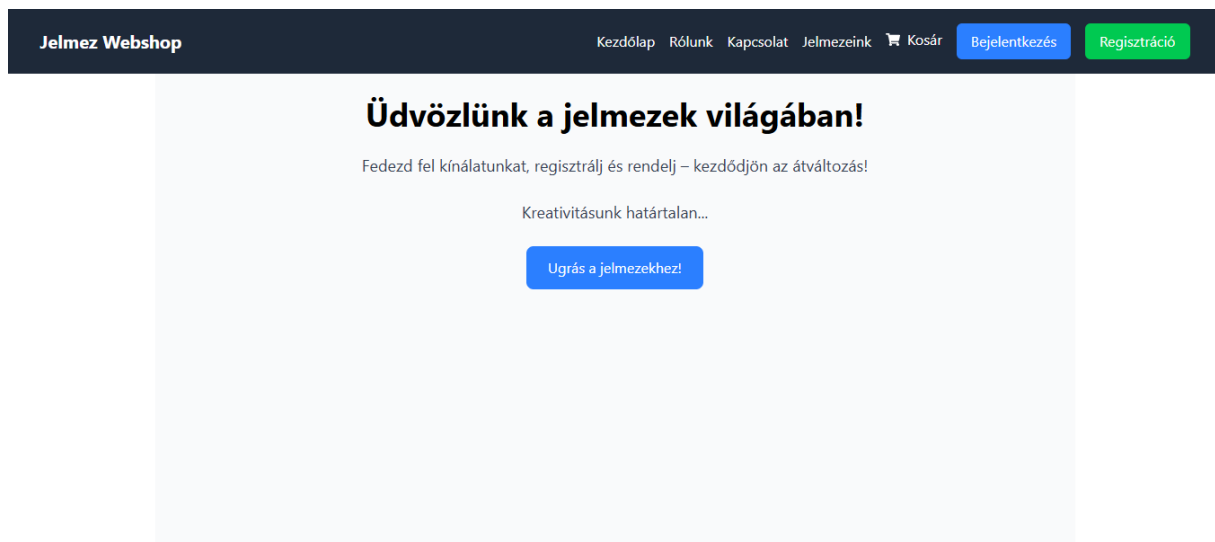
**Regisztráció** - Amennyiben szeretnének regisztrálhatnak is, így a későbbi rendelések során időt takaríthatnak meg.

### 3.5 Szabadon elérhető és regisztrációhoz kötött funkciók

Jelen formájában a program nem tér el funkciójában a regisztráció nélküli és a regisztráció utáni működésben.

### 3.6 Szemléltetés

Kezdőlap lap bemutatása:



## Rólunk rovat bemutatása:

**Jelmez Webshop**[Kezdőlap](#)[Rólunk](#)[Kapcsolat](#)[Jelmezeink](#) [Kosár](#)[Bejelentkezés](#)[Regisztráció](#)

## Rólunk


Üdvözlünk a legizgalmasabb debreceni jelmez webshop honlapján! Nálunk mindent megtalálsz, amire szükséged lehet ahhoz, hogy emlékeztetessé tedd a különleges alkalmakat. Webshopunkban széles választékban kínálunk jelmezeket gyerekeknek és felnőtteknek egyaránt, klasszikus mesehősöktől a modern filmes karakterekig, hogy mindenki megtalálja a saját stílusának megfelelő darabot.

Célunk, hogy a vásárlás nálunk gyors, egyszerű és kényelmes legyen. Csatunk gondosan válogatott, kiváló minőségű jelmezeket kínál, melyek garantáltan megidéznek az általad választott karakter varázsát. Nálunk nem csak termékeket vásárolhatsz, hanem olyan élményeket, amelyek mosolyt csálnak az arcokra.

Az aktuális trendeket folyamatosan figyelemmel kísérjük, hogy mindig friss és újszerű kínálattal várassunk. Akár születésnap, Halloween, farsang vagy tematikus buli, webshopunkban biztosan találsz valami igazán különlegeset. Büszkék vagyunk arra, hogy veled együtt tehetjük szebbé és színesebbé az ünnepi pillanatokat.

Lépj be webshopunk virtuális világába, és válogass kedvedre, hogy felejthetetlen élményben lehessen részed!

## Kapcsolati fül bemutatása:

**Jelmez Webshop**[Kezdőlap](#)[Rólunk](#)[Kapcsolat](#)[Jelmezeink](#) [Kosár](#)[Bejelentkezés](#)[Regisztráció](#)

## Kapcsolat

**Telefon:** +36(30)765-4321

**Címünk:** 4025 Debrecen, Széchenyi u. 58.

**Nyitvatartás:**

- H-P: 8:00-16:30
- Sz: 8:00-12:00
- V: Zárva

**E-mail cím:** jelmez@debrecen.hu



Jelmezeink felület áttekintése:


Jelmez Webshop

KezdőlapRólunkKapcsolatJelmezeinkKosár

BejelentkezésRegisztráció

### Jelmezeink

Mind




**Hókuszpók jelmez**  
Ár: 15 490 Ft  
Férfi

**Kiegészítők:**

- Tunika
- Kopasz paróka
- Lábmelegítők
- Cipőfedők

RészletekKosárba




**Joker jelmez**  
Ár: 18 290 Ft  
Férfi

**Kiegészítők:**

- Kabát
- Gallér
- Nyakkendő
- Mellény
- Nadrág
- Kesztyű

RészletekKosárba




**Vilma jelmez**  
Ár: 9 590 Ft  
Női

**Kiegészítők:**

- Ruha
- Paróka
- Nyaklánc
- Lábak

RészletekKosárba



**Piroska és a farkas jelmez**  
Ár: 9 890 Ft  
Női


**Kiegészítők:**

- Ruha
- Köpeny

**NEM tartozékok:**

- Harisnya
- Kosár

RészletekKosárba




**Kleopátra jelmez**  
Ár: 8 990 Ft  
Női

**Kiegészítők:**

- Ruha
- Gallér

RészletekKosárba




**Fogoly jelmez**  
Ár: 5 990 Ft  
Férfi

**Kiegészítők:**

- Ing
- Öv

RészletekKosárba




**Harley Quinn jelmez**  
Ár: 15 690 Ft  
Női

**Kiegészítők:**

- Dzseki hímzéssel és flitterrel
- Rövidnadrág

RészletekKosárba



**Macskanő jelmez**  
Ár: 17 990 Ft  
Női

**Kiegészítők:**

- Elastikus szublimált ruha
- Kesztyű

RészletekKosárba

Jelmez Webshop

KezdőlapRólunkKapcsolatJelmezeinkKosár

BejelentkezésRegisztráció



### Joker jelmez

Készülj egy kis káoszra! A Joker jelmez (A Sötét Lovag stílusában) zöld-lila színekombinációval, hosszú kabáttal és maszkkal érkezik. Egy igazán karizmatikus gonosztevőnek készült!

Méret kiválasztása:

-- Válassz méretet --

Kosárba

Bejelentkezés gomb bemutatása:

The screenshot shows the login page of the 'Jelmez Webshop'. The header is dark blue with the shop name on the left and navigation links on the right. The main content area is light gray and contains a white login form titled 'Bejelentkezés'. The form has two input fields: 'Felhasználónév' (Username) with the text 'teszt' and 'Jelszó' (Password) with masked characters. A blue 'Bejelentkezés' button is at the bottom of the form.

**Jelmez Webshop**

Kezdőlap Rólunk Kapcsolat Jelmezeink Kosár **Bejelentkezés** **Regisztráció**

### Bejelentkezés

Felhasználónév

Jelszó

**Bejelentkezés**

Regisztráció gomb bemutatása:

The screenshot shows the registration page of the 'Jelmez Webshop'. The header is identical to the login page. The main content area is light gray and contains a white registration form titled 'Regisztráció'. The form has three input fields: 'Felhasználónév' (Username) with placeholder text 'Írd be a felhasználóneved', 'Email' with placeholder text 'Írd be az email címed', and 'Jelszó' (Password) with placeholder text 'Írd be a jelszavad'. A green 'Regisztráció' button is at the bottom of the form.

**Jelmez Webshop**

Kezdőlap Rólunk Kapcsolat Jelmezeink Kosár **Bejelentkezés** **Regisztráció**

### Regisztráció

Felhasználónév

Email

Jelszó

**Regisztráció**

### **3.7 Hibajelzések és figyelmeztető üzenetek felhasználó a tevékenysége során**

- Kérjük válassz méretet!
- Joker jelmez (M) hozzáadva a kosárhoz!
- Még egy lépés hátra van – kérjük válassz méretet:
- Ez a méret már a kosárban van!
- Kérjük töltse ki a mezőt!

### **3.8 Admin felhasználó lehetőségei**

Az admin felhasználóknak lehetőségük van az összes rendelés kezelésére, felhasználói profilok módosítására és új termékek hozzáadására

### **3.9 információkérés lehetősége.**

Ahogy a webalapon is olvasható:

**Telefon: +36(30)765-4321**

**E-mail cím: [jelmez@debrecen.hu](mailto:jelmez@debrecen.hu)**

## 4. Összefoglaló

### 4.1 Bevezetés

A projekt célja egy felhasználóbarát jelmezek online vásárlásra szolgáló webshop létrehozása, amely lehetővé teszi a termékek kategória szerinti böngészését, keresést, kosárba helyezést és vásárlást. Az alkalmazás modern technológiákra épül, és nagy hangsúlyt fektet a felhasználói élményre.

### 4.2 Technikai értékelés

Frontend:

- A webshop React alapokra épül, így gyors és dinamikus.
- Az állapotkezelés jól strukturált, és a felhasználói interakciók azonnal láthatóak.
- A ProductPage.jsx jól szervezett, külön komponensekkel (ProductCard, Header).

Backend:

- A szerveroldali API megfelelően kezeli a termékek lekérdezését és szűrését.
- Az adatbázis SQLite alapú, és megfelelően tárolja az adatokat.
- A django\_session használata biztosítja a munkamenetek tárolását.

Design és UX:

- A webshop vizuálisan letisztult, mobilbarát, és jól igazodik a különböző képernyőméretekhez.
- A méretválasztó pontos és konkrét, egyértelműsített.
- A szűrési logika pontosan működik, miután az eredeti hibák javításra kerültek.

### 4.3 Végeredmény értékelése

A projekt **alaposanátgondolt**, és a technikai alapjai stabilak és modern megoldásokat használ. A megvalósítás során folyamatosan **finomhangolás** történt, ami az alkalmazás minőségét javította. A **projkt összességében működékpépes és használható**, de érdemes lehet további finomításokat végezni, például **termékértékelések és vélemények** bevezetése. **A webshop éles környezetben is jól működhet**, ha megfelelő skálázást és optimalizálást kap.

## 5. Irodalom jegyzék

<https://www.w3schools.com>

<https://docs.djangoproject.com/en/5.0/>

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

<https://infojegyzet.hu/webszerkesztes/dokumentacio/>

<https://docs.python.org/3/>