

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

**Szoftverfejlesztő és –tesztelő projektfeladat**

**Raktárkezelő rendszer**

**Készítették:**

**Kovács Tamás 14EC**

**Barcza Máté 14EC**

**Fehér Ádám 14EC**

**2024.**

Tartalomjegyzék

[1. Bevezetés, a téma ismertetése 4](#_Toc165130101)

[2. Fejlesztői dokumentáció 5](#_Toc165130102)

[2.1. A fejlesztőkörnyezet 5](#_Toc165130103)

[2.2. A kialakított adatszerkezet 5](#_Toc165130104)

[2.3. A program érdekesebb algoritmusai, kódrészletei 13](#_Toc165130105)

[2.4. Tesztelés 15](#_Toc165130106)

[2.5. Fejlesztési lehetőségek 15](#_Toc165130107)

[3. Felhasználói dokumentáció 16](#_Toc165130108)

[3.1. A program célja, lényegesebb funkciói 16](#_Toc165130109)

[3.2. Szükséges hardvereszközök és szoftver 16](#_Toc165130110)

[3.3. Telepítés és indítás 16](#_Toc165130111)

[3.4. A program részletes bemutatása 16](#_Toc165130112)

[3.4.1. Rendelések nyilvántartása 19](#_Toc165130113)

[3.4.2. Termékek hozzáadása a raktárkészlethez 21](#_Toc165130114)

[3.5. Hibajelzések 22](#_Toc165130115)

[3.6. Az admin felhasználó lehetőségei 23](#_Toc165130116)

[4. Információkérés lehetősége 24](#_Toc165130117)

[5. Összefoglalás, köszönetnyilvánítás 25](#_Toc165130118)

[6. Irodalomjegyzék 26](#_Toc165130119)

Ábrajegyzék

[1. ábra: A termékkategóriák kiválasztása 14](#_Toc165130120)

[2. ábra: A termék képének hozzáadására szolgáló felület 15](#_Toc165130121)

[3. ábra: Az admin felhasználó bejelentkezési felülete 16](#_Toc165130122)

[4. ábra: Az admin felhasználói felület 17](#_Toc165130123)

[5. ábra: A raktárban nyilvántartott termékek listája 17](#_Toc165130124)

[6. ábra: Keresés a termék teljes nevének megadásával 18](#_Toc165130125)

[7. ábra: Keresés a kezdőszó megadásával 18](#_Toc165130126)

[8. ábra: Keresés egy szövegrészlet megadásával 18](#_Toc165130127)

[9. ábra: Keresés a termék ára szerint 18](#_Toc165130128)

[10. ábra: A jelölőnégyzet működése 19](#_Toc165130129)

[11. ábra: A rendelések nyilvántartó felülete 19](#_Toc165130130)

[12. ábra: Rendelés hozzáadására szolgáló felület 19](#_Toc165130131)

[13. ábra: Példa rendelés hozzáadására 20](#_Toc165130132)

[14. ábra: A termékek listája a rendelés felületen 20](#_Toc165130133)

[15. ábra: A rendelés állapota 20](#_Toc165130134)

[16. ábra: A Szállítva lenyíló lista választható lehetőségei 20](#_Toc165130135)

[17. ábra: A rendelés véglegesítésére szolgáló felület 21](#_Toc165130136)

[18. ábra: A felvezetett rendelés az utolsó sorban 21](#_Toc165130137)

[19. ábra: Az adatbázisban szereplő termékek hozzáadására szolgáló felhasználói felület 21](#_Toc165130138)

[20. ábra: Példa termék hozzáadására 21](#_Toc165130139)

[21. ábra: A termék elérhetősége a raktárban 21](#_Toc165130140)

[22. ábra: A raktárban szereplő termék kategorizálása 22](#_Toc165130141)

[23. ábra: A felvezetett termékadatok véglegesítésére szolgáló felület 22](#_Toc165130142)

[24. ábra: Hibaüzenet, amennyiben egy termék nem szerepel az adatbázisban 23](#_Toc165130143)

1. Bevezetés, a téma ismertetése

Napjainkban az interneten történő vásárlás egyre megszokottabbá válik főként annak kényelméből adódóan. A kizárólag az interneten árusító cégek (webáruházak) komoly vetélytársai lettek a fizikai valóságban is létező üzleteknek, így az utóbbiak is sok esetben az üzlet mellett egy online áruházat is üzemeltetnek a versenyben maradás- és a vásárlók kényelmének érdekében.

Legyen szó akár internetes áruházról, akár egy webshopot üzemeltető boltról, a honlapon feltűntetett termékeket fizikailag is tárolni kell. Széles termékportfólió esetén a kézi nyilvántartás nehezen kivitelezhető, pontatlanságokat eredményezhet. A raktárban felhalmozott, ott tárolt, az oda beérkező-, az onnan kimenő-, az esetlegesen visszaérkező-, valamint a hiány állapotban lévő termékek nyilvántartására az egyes vállalatok raktárkezelő nyilvántartó programokat alkalmaznak.

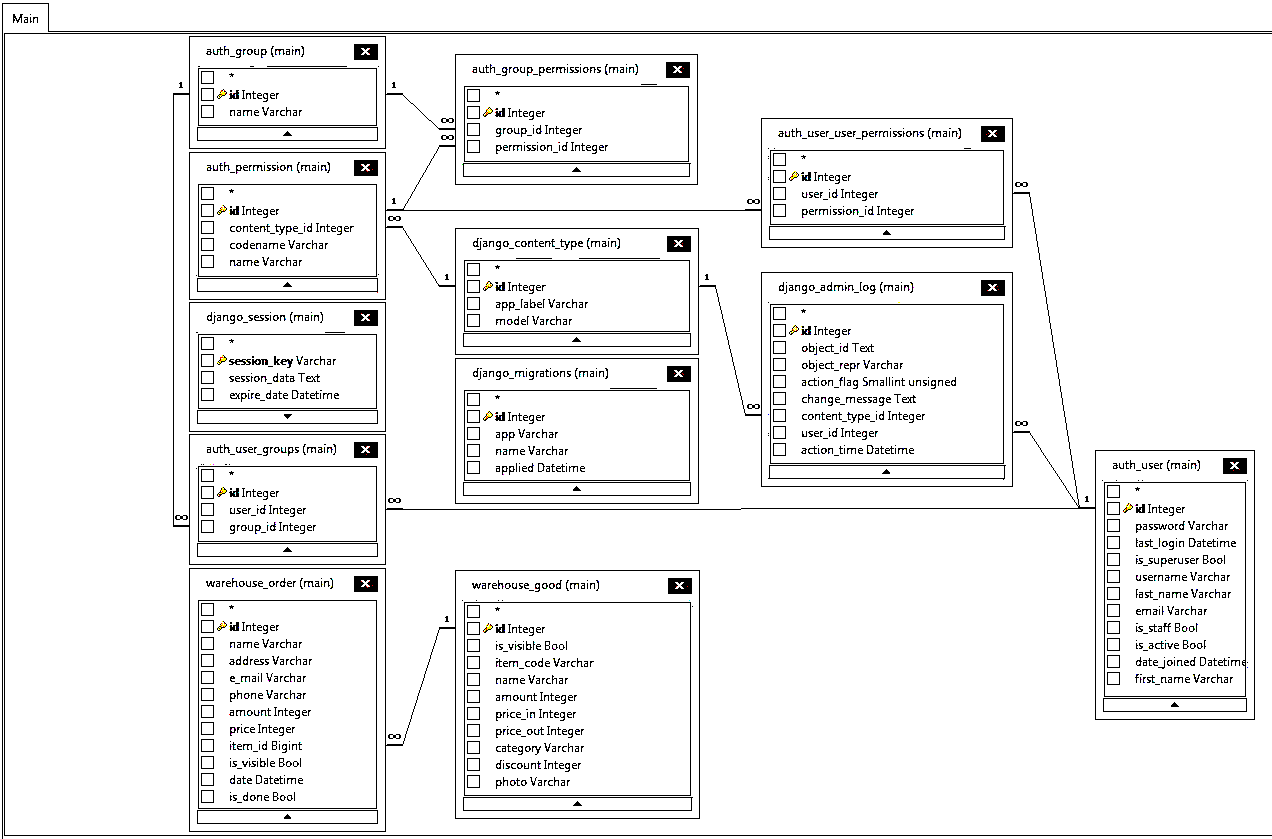
A jelen projektfeladatban szereplő raktárkezelő weboldal célja egy elektronikai cikkeket árusító webáruház termékeinek nyilvántartása. A nyilvántartásba beletartozik a termékportfólióban bekövetkezett esetleges változásának, a termékek mennyiségeinek változásának (beszerzés, rendelés, rendelés visszamondás esetei), valamint az egyes termékek különböző adatainak változásának naprakészen tartása.

1. Fejlesztői dokumentáció
   1. A fejlesztőkörnyezet

A Raktárkezelő projektfeladat elkészítése során Visual Studio Code fejlesztőkörnyezetet használtunk Django web keretrendszerrel, melyek beépített funkcióival jelentősen elősegítik a weblapfejlesztés folyamatát. A változások nyomonkövetésére GitHub változáskövető rendszert alkalmaztunk.

* 1. A kialakított adatszerkezet

A kialakított adatszerkezet az 1. ábrán látható módon épül fel. Az adatszerkezet vizualizációjára SQLite Expert Professional szoftvert alkalmaztunk.



A Django web keretrendszer tábláin túl további két tábla került kialakításra:  
A „warehouse\_good”, mely a raktárban található árucikkeket tartalmazza azok neveivel, azonosítóival, mennyiségeivel, áraival, kategóriáival, kedvezményeivel és fényképeivel.   
A „warehouse\_order” - a rendelés adatait tartalmazó tábla - olyan adatokkal, mint az illető neve, címe, e-mail címe, telefonszáma, a megrendelt árucikk azonosítója, mennyisége, ára és a rendelés dátuma. A „warehouse\_order” idegen kulcsa a „warehouse\_good” „item\_id” elsődleges kulcsa.

Az adatbázis sematikája alább olvasható.

*Database: [db]*

*Connection: read-write*

*File name: D:\02\_Szoftverfejleszto\_es\_tesztelo\4\_Felev\Projekt\_feladat\vizsga\config\db.sqlite3*

*File size: 151552 bytes*

*Page size: 4096*

*Encoding: UTF-8*

*Auto vacuum: 0*

*Tables: 14*

*Views: 0*

*Virtual Tables: 0*

*------------------------------------------------------------*

*Table [auth\_group]*

*Fields: 2*

*[id]: integer*

*NOT NULL*

*[name]: varchar(150)*

*UNIQUE*

*NOT NULL*

*Indexes: 0*

*Triggers: 0*

*Table constraints:*

*Primary Key:*

*Fields: [id]*

*On Conflict:*

*Foreign Keys: 0*

*Unique constraints: 0*

*Check constraints: 0*

*Table [auth\_group] end*

*------------------------------------------------------------*

*Table [auth\_group\_permissions]*

*Fields: 3*

*[id]: integer*

*NOT NULL*

*[group\_id]: integer*

*NOT NULL*

*[permission\_id]: integer*

*NOT NULL*

*Indexes: 3*

*[auth\_group\_permissions\_permission\_id\_84c5c92e]*

*[permission\_id]*

*[auth\_group\_permissions\_group\_id\_b120cbf9]*

*[group\_id]*

*[auth\_group\_permissions\_group\_id\_permission\_id\_0cd325b0\_uniq] UNIQUE*

*[group\_id]*

*[permission\_id]*

*Triggers: 0*

*Table constraints:*

*Primary Key:*

*Fields: [id]*

*On Conflict:*

*Foreign Keys: 2*

*[] ([group\_id]) REFERENCES [auth\_group]([id]) DEFERRABLE DEFERRED*

*[] ([permission\_id]) REFERENCES [auth\_permission]([id]) DEFERRABLE DEFERRED*

*Unique constraints: 0*

*Check constraints: 0*

*Table [auth\_group\_permissions] end*

*------------------------------------------------------------*

*Table [auth\_permission]*

*Fields: 4*

*[id]: integer*

*NOT NULL*

*[content\_type\_id]: integer*

*NOT NULL*

*[codename]: varchar(100)*

*NOT NULL*

*[name]: varchar(255)*

*NOT NULL*

*Indexes: 2*

*[auth\_permission\_content\_type\_id\_codename\_01ab375a\_uniq] UNIQUE*

*[content\_type\_id]*

*[codename]*

*[auth\_permission\_content\_type\_id\_2f476e4b]*

*[content\_type\_id]*

*Triggers: 0*

*Table constraints:*

*Primary Key:*

*Fields: [id]*

*On Conflict:*

*Foreign Keys: 1*

*[] ([content\_type\_id]) REFERENCES [django\_content\_type]([id]) DEFERRABLE DEFERRED*

*Unique constraints: 0*

*Check constraints: 0*

*Table [auth\_permission] end*

*------------------------------------------------------------*

*Table [auth\_user]*

*Fields: 11*

*[id]: integer*

*NOT NULL*

*[password]: varchar(128)*

*NOT NULL*

*[last\_login]: datetime*

*[is\_superuser]: bool*

*NOT NULL*

*[username]: varchar(150)*

*UNIQUE*

*NOT NULL*

*[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*

*Indexes: 0*

*Triggers: 0*

*Table constraints:*

*Primary Key:*

*Fields: [id]*

*On Conflict:*

*Foreign Keys: 0*

*Unique constraints: 0*

*Check constraints: 0*

*Table [auth\_user] end*

*------------------------------------------------------------*

*Table [auth\_user\_groups]*

*Fields: 3*

*[id]: integer*

*NOT NULL*

*[user\_id]: integer*

*NOT NULL*

*[group\_id]: integer*

*NOT NULL*

*Indexes: 3*

*[auth\_user\_groups\_group\_id\_97559544]*

*[group\_id]*

*[auth\_user\_groups\_user\_id\_6a12ed8b]*

*[user\_id]*

*[auth\_user\_groups\_user\_id\_group\_id\_94350c0c\_uniq] UNIQUE*

*[user\_id]*

*[group\_id]*

*Triggers: 0*

*Table constraints:*

*Primary Key:*

*Fields: [id]*

*On Conflict:*

*Foreign Keys: 2*

*[] ([user\_id]) REFERENCES [auth\_user]([id]) DEFERRABLE DEFERRED*

*[] ([group\_id]) REFERENCES [auth\_group]([id]) DEFERRABLE DEFERRED*

*Unique constraints: 0*

*Check constraints: 0*

*Table [auth\_user\_groups] end*

*------------------------------------------------------------*

*Table [auth\_user\_user\_permissions]*

*Fields: 3*

*[id]: integer*

*NOT NULL*

*[user\_id]: integer*

*NOT NULL*

*[permission\_id]: integer*

*NOT NULL*

*Indexes: 3*

*[auth\_user\_user\_permissions\_permission\_id\_1fbb5f2c]*

*[permission\_id]*

*[auth\_user\_user\_permissions\_user\_id\_a95ead1b]*

*[user\_id]*

*[auth\_user\_user\_permissions\_user\_id\_permission\_id\_14a6b632\_uniq] UNIQUE*

*[user\_id]*

*[permission\_id]*

*Triggers: 0*

*Table constraints:*

*Primary Key:*

*Fields: [id]*

*On Conflict:*

*Foreign Keys: 2*

*[] ([user\_id]) REFERENCES [auth\_user]([id]) DEFERRABLE DEFERRED*

*[] ([permission\_id]) REFERENCES [auth\_permission]([id]) DEFERRABLE DEFERRED*

*Unique constraints: 0*

*Check constraints: 0*

*Table [auth\_user\_user\_permissions] end*

*------------------------------------------------------------*

*Table [django\_admin\_log]*

*Fields: 8*

*[id]: integer*

*NOT NULL*

*[object\_id]: text*

*[object\_repr]: varchar(200)*

*NOT NULL*

*[action\_flag]: smallint unsigned*

*NOT NULL*

*[change\_message]: text*

*NOT NULL*

*[content\_type\_id]: integer*

*[user\_id]: integer*

*NOT NULL*

*[action\_time]: datetime*

*NOT NULL*

*Indexes: 2*

*[django\_admin\_log\_content\_type\_id\_c4bce8eb]*

*[content\_type\_id]*

*[django\_admin\_log\_user\_id\_c564eba6]*

*[user\_id]*

*Triggers: 0*

*Table constraints:*

*Primary Key:*

*Fields: [id]*

*On Conflict:*

*Foreign Keys: 2*

*[] ([content\_type\_id]) REFERENCES [django\_content\_type]([id]) DEFERRABLE DEFERRED*

*[] ([user\_id]) REFERENCES [auth\_user]([id]) DEFERRABLE DEFERRED*

*Unique constraints: 0*

*Check constraints: 1*

*[] Expression: '"action\_flag" >= 0'*

*Table [django\_admin\_log] end*

*------------------------------------------------------------*

*Table [django\_content\_type]*

*Fields: 3*

*[id]: integer*

*NOT NULL*

*[app\_label]: varchar(100)*

*NOT NULL*

*[model]: varchar(100)*

*NOT NULL*

*Indexes: 1*

*[django\_content\_type\_app\_label\_model\_76bd3d3b\_uniq] UNIQUE*

*[app\_label]*

*[model]*

*Triggers: 0*

*Table constraints:*

*Primary Key:*

*Fields: [id]*

*On Conflict:*

*Foreign Keys: 0*

*Unique constraints: 0*

*Check constraints: 0*

*Table [django\_content\_type] end*

*------------------------------------------------------------*

*Table [django\_migrations]*

*Fields: 4*

*[id]: integer*

*NOT NULL*

*[app]: varchar(255)*

*NOT NULL*

*[name]: varchar(255)*

*NOT NULL*

*[applied]: datetime*

*NOT NULL*

*Indexes: 0*

*Triggers: 0*

*Table constraints:*

*Primary Key:*

*Fields: [id]*

*On Conflict:*

*Foreign Keys: 0*

*Unique constraints: 0*

*Check constraints: 0*

*Table [django\_migrations] end*

*------------------------------------------------------------*

*Table [django\_session]*

*Fields: 3*

*[session\_key]: varchar(40)*

*NOT NULL*

*[session\_data]: text*

*NOT NULL*

*[expire\_date]: datetime*

*NOT NULL*

*Indexes: 1*

*[django\_session\_expire\_date\_a5c62663]*

*[expire\_date]*

*Triggers: 0*

*Table constraints:*

*Primary Key:*

*Fields: [session\_key]*

*On Conflict:*

*Foreign Keys: 0*

*Unique constraints: 0*

*Check constraints: 0*

*Table [django\_session] end*

*------------------------------------------------------------*

*Table [sqlite\_master]*

*Fields: 5*

*[type]: TEXT*

*[name]: TEXT*

*[tbl\_name]: TEXT*

*[rootpage]: INTEGER*

*[sql]: TEXT*

*Indexes: 0*

*Triggers: 0*

*Table constraints:*

*Primary Key:*

*Fields:*

*On Conflict:*

*Foreign Keys: 0*

*Unique constraints: 0*

*Check constraints: 0*

*Table [sqlite\_master] end*

*------------------------------------------------------------*

*Table [sqlite\_sequence]*

*Fields: 2*

*[name]:*

*[seq]:*

*Indexes: 0*

*Triggers: 0*

*Table constraints:*

*Primary Key:*

*Fields:*

*On Conflict:*

*Foreign Keys: 0*

*Unique constraints: 0*

*Check constraints: 0*

*Table [sqlite\_sequence] end*

*------------------------------------------------------------*

*Table [warehouse\_good]*

*Fields: 10*

*[id]: integer*

*NOT NULL*

*[is\_visible]: bool*

*NOT NULL*

*[item\_code]: varchar(255)*

*NOT NULL*

*[name]: varchar(255)*

*NOT NULL*

*[amount]: integer*

*NOT NULL*

*[price\_in]: integer*

*NOT NULL*

*[price\_out]: integer*

*NOT NULL*

*[category]: varchar(50)*

*NOT NULL*

*[discount]: integer*

*NOT NULL*

*[photo]: varchar(100)*

*NOT NULL*

*Indexes: 0*

*Triggers: 0*

*Table constraints:*

*Primary Key:*

*Fields: [id]*

*On Conflict:*

*Foreign Keys: 0*

*Unique constraints: 0*

*Check constraints: 0*

*Table [warehouse\_good] end*

*------------------------------------------------------------*

*Table [warehouse\_order]*

*Fields: 11*

*[id]: integer*

*NOT NULL*

*[name]: varchar(255)*

*NOT NULL*

*[address]: varchar(255)*

*NOT NULL*

*[e\_mail]: varchar(255)*

*NOT NULL*

*[phone]: varchar(20)*

*NOT NULL*

*[amount]: integer*

*NOT NULL*

*[price]: integer*

*NOT NULL*

*[item\_id]: bigint*

*NOT NULL*

*[is\_visible]: bool*

*NOT NULL*

*[date]: datetime*

*NOT NULL*

*[is\_done]: bool*

*NOT NULL*

*Indexes: 1*

*[warehouse\_order\_item\_id\_4ef4a399]*

*[item\_id]*

*Triggers: 0*

*Table constraints:*

*Primary Key:*

*Fields: [id]*

*On Conflict:*

*Foreign Keys: 1*

*[] ([item\_id]) REFERENCES [warehouse\_good]([id]) DEFERRABLE DEFERRED*

*Unique constraints: 0*

*Check constraints: 0*

*Table [warehouse\_order] end*

------------------------------------------------------------

* 1. A program érdekesebb algoritmusai, kódrészletei

1. **Tekintsük az alábbi kódrészletet a views.py állományból:**

@api\_view(['GET'])

def item\_view(request,pk):

    try:

        good = Good.objects.get(id=pk)

    except:

        return Response({'message':f'Good ID {pk} is not exsist!'})

    serialized = GoodSerializer(good, many=False)

    choices = Good.CATEGORY\_CHOICES

    return render(request,'item.html',{'item': serialized.data,'choices':choices})

A kódrészlet lehetővé teszi, hogy az adatbázisból az adatok a termék oldalára fetch művelet segítsége nélkül kerüljenek. A „good” nevű változó a „Good” osztályból kap meg minden szükséges adatot. A termék „item.html” oldalán az adatok az alábbi kód segítségével kerülnek átadásra.

<input type="text" style="border-radius: 10px;" name="item\_code" value="{{item.item\_code}}">

Új termék felvétele esetén különböző kategóriákba sorolhatjuk azt a későbbi könnyebb keresés érdekében (1. ábra). A „choices” változó értéke a „Good” osztály „CATEGORY CHOICES” elemeit veszi fel, melyek a „models.py” fájlban vannak definiálva az alábbiak szerint:

class Good(models.Model):

    CATEGORY\_CHOICES = (

    ("1","Szórakoztató elektronika"),

    ("2","Háztartási gép"),

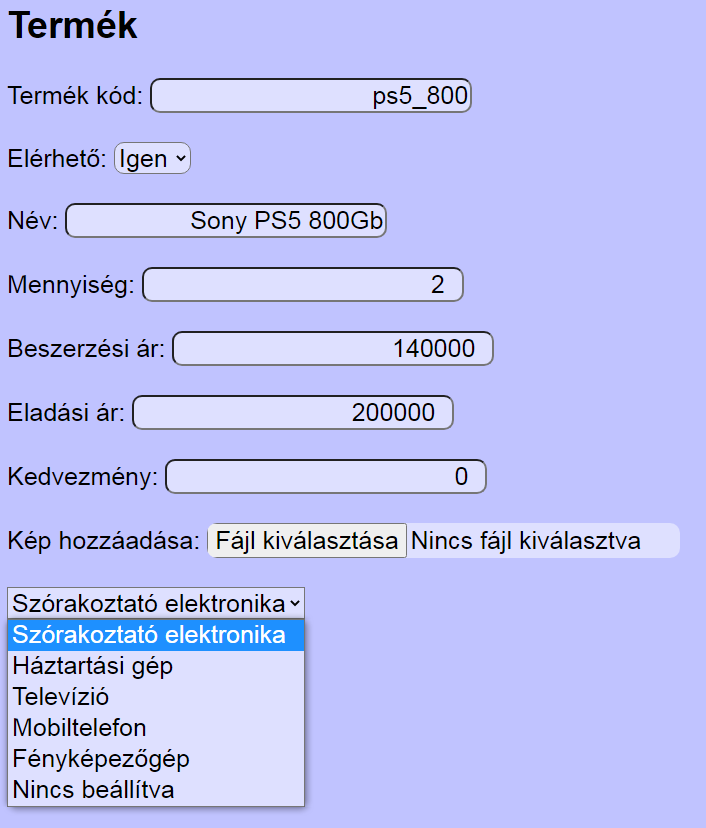
    ("3","Televízió"),

    ("4","Mobiltelefon"),

    ("5","Fényképezőgép"),

    ("6","Nincs beállítva")

Ezek segítségével a .html állomány már a kész adatokat kapja meg, a renderelés a szerver oldalon történik.



1. ábra: A termékkategóriák kiválasztása

A „serialized = GoodSerializer(good, many=False)” sorban a GoodSerializer osztály egy példányát hozzuk létre, amelynek a feladata a Good modell objektumainak szerializálása. A GoodSerializer példányosítása fontos, mert a HTTP válaszokat általában adatok formájában kell visszaadni a kliensnek. Ez a folyamat magában foglalja az adatok átalakítását olyan formátumba, amely könnyen kezelhető a kliens oldalon. A szerializáció egy ilyen átalakítási folyamat, amely lehetővé teszi az objektumok átalakítását JSON formátumba.

1. **Tekintsük az alábbi kódrészletet az „add\_good.html” állományból:**

 <form action="{%url 'good' %}" method="post">

        {% csrf\_token %}

A {% csrf\_token %} a Cross-Site Request Forgery (CSRF) védelem része. A CSRF egy olyan támadási módszer, amikor egy támadó valamilyen módon manipulálja a felhasználó böngészőjét, hogy olyan kérelmeket küldjön egy webalkalmazás szerverének, amelyek az adott felhasználó nevében futnak, anélkül hogy ő tudna róla. A {% csrf\_token %} parancs beilleszti a Jinja sablonba a CSRF token-t, ami egy véletlenszerűen generált érték. Ez segít megvédeni az alkalmazást az ilyen típusú támadások ellen, mivel a szerver csak azokat a kérelmeket fogadja el, amelyek tartalmazzák a helyes CSRF token-t.

* 1. Tesztelés

A program tesztelésének célja a raktárkezelő weboldal helyes működéséről megbizonyosodni különböző böngészők használatával.

A tesztelés során a raktárkészleten lévő termékek és a megrendelt termékek mennyiségének és árának összehasonlításánál tapasztaltunk rendellenes működést. Az összehasonlításért az alábbi kódrészlet felel:

order.amount = int(request.data['amount'])

            order.price = int(request.data['price'])

            amount = abs(order\_original\_amount - order.amount)

            if order\_original\_amount > order.amount:

                item\_id.amount += amount

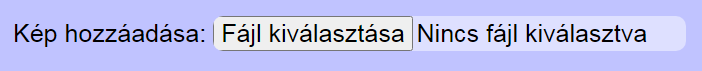
            elif order\_original\_amount < order.amount:

                item\_id.amount -= amount

Mivel az „amount” mennyiség és a „price” ár adatokat string-ként kapjuk meg, az összehasonlítás érdekében egész szám formátumúra (integer) kell alakítani. Ez az átalakítás kezdetben nem történt meg, ezért nem működött megfelelően a weboldal. A kódrészlet javításra került.

* 1. Fejlesztési lehetőségek

Fejlesztésre mindig van lehetőég. Jelen Raktárkezelő weblap fejlesztési lehetőségei közé soroljuk az egyes, nyilvántartott termékekhez tartozó képek hozzáadását, melynek a feltöltő gombja már elhelyezésre került az „item.html” oldalon (2. ábra).



2. ábra: A termék képének hozzáadására szolgáló felület

További fejlesztési lehetőség egy, a Raktárkezelő rendszerhez tartozó és azzal összhangban működő internetes áruház oldal kialakítása, mely a vásárlók számára biztosítana közvetlen hozzáférést a nem admin felületekhez.

1. Felhasználói dokumentáció
   1. A program célja, lényegesebb funkciói

A raktárkezelő weblap lehetővé teszi egy vállalat számára, hogy nagy mennyiségű terméket tartson nyilván, azok minden adatával. Továbbá a weblap segítségével felvezethetjük a vevők megrendeléseit és naprakész információkhoz juthatunk az összes termékről.

* 1. Szükséges hardvereszközök és szoftver

A Raktárkezelő weblap használatához egy számítógépre / laptopra / okostelefonra, egy billentyűzetre, egy képi megjelenítést lehetővé tévő eszközre (pl. monitorra), a kurzor mozgatását biztosító egérre és internetkapcsolatra van szükség. Szükséges továbbá egy webböngésző szoftver (pl. Internet Explorer, Google Chrome, Mozilla Firefox) megléte az adott eszközön

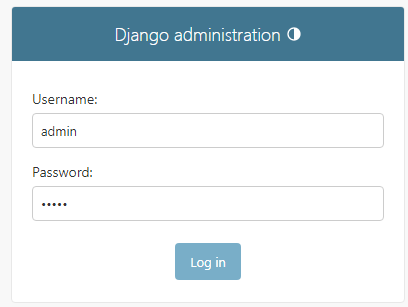
* 1. Telepítés és indítás

A Raktárkezelő weboldal telepítést nem igényel, megnyitása és elérése webböngészőn keresztül történik.

* 1. A program részletes bemutatása

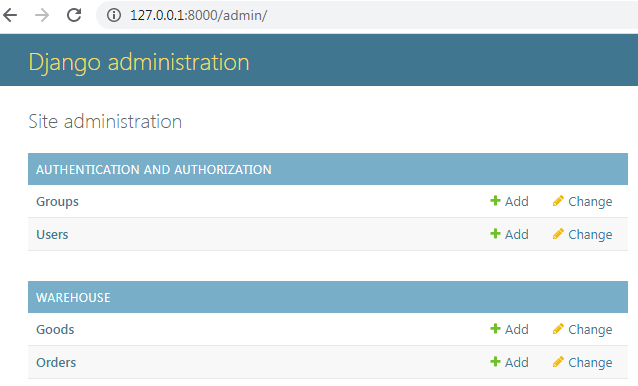
A Raktárkezelő program adatbázisa jelenleg az alábbi címen érhető el:

<http://127.0.0.1:8000/admin/login/?next=/admin/>



. ábra: Az admin felhasználó bejelentkezési felülete

Az „admin” felhasználónév és az „admin” jelszó megadását követően az alábbi felületre érkezünk (4. ábra).

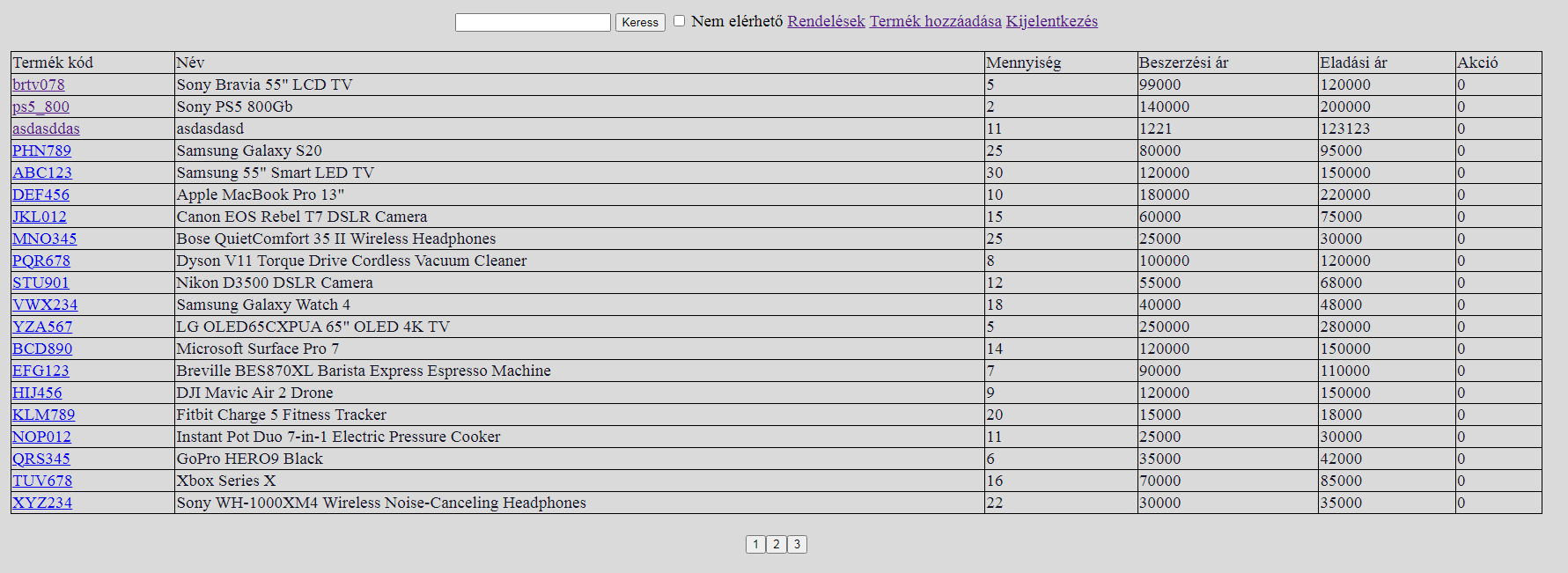


4. ábra: Az admin felhasználói felület

A felületen szerkeszthetjük az adatbázist, új termékeket vihetünk fel és törölhetünk.

Maga a Raktárkezelő program admin felhasználó számára elérhető felülete (5. ábra) az alábbi címen érhető el:

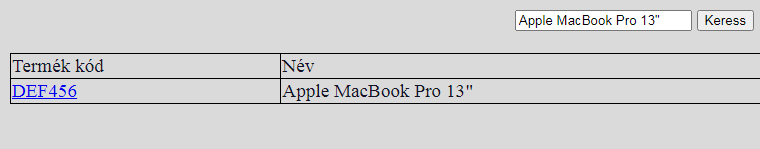
<http://127.0.0.1:8000/w_admin/>



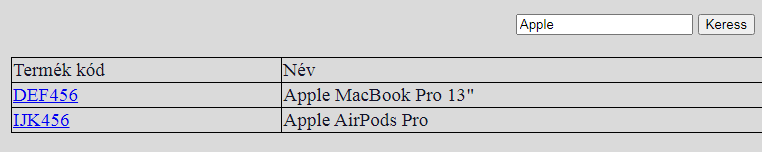
5. ábra: A raktárban nyilvántartott termékek listája

Mint látható a felület 3 oldalból áll jelenleg, de ez a termékek mennyiségétől függ. Az egyes oldalakra az oldalszámokra történő kattintással juthatunk el. Az oldal tartalmaz egy keresőmezőt, ahol közvetlenül a termék nevének vagy egyéb adatának megadásával szűrhetjük a raktárban nyilvántartott árucikkeket. Nem szükséges a termék nevének pontos megadása, elegendő, ha a keresett szöveg csupán része a termék nevének. A keresőmezőbe írást követően a „Keress” feliratú gomb megnyomásával küldhetjük el a kérést.

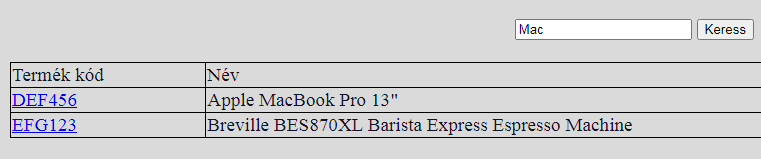
A keresőmező különböző felhasználási módját mutatják be a következő ábrák.



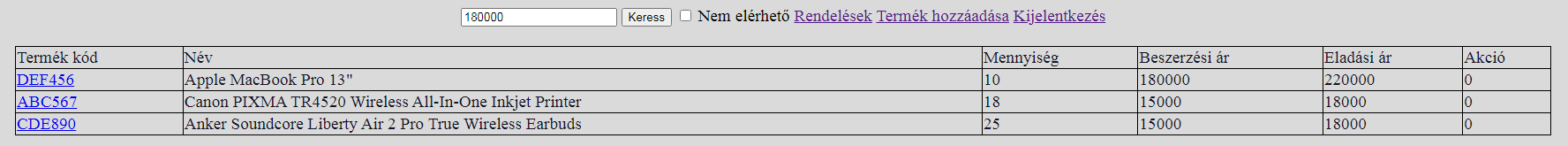
. ábra: Keresés a termék teljes nevének megadásával



. ábra: Keresés a kezdőszó megadásával

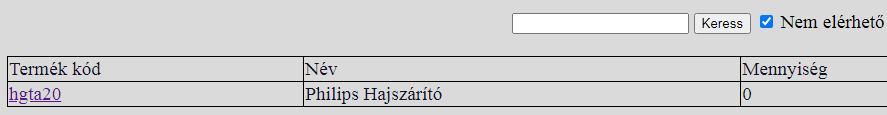


8. ábra: Keresés egy szövegrészlet megadásával



. ábra: Keresés a termék ára szerint

Az oldal tartalmaz egy jelölőnégyzetet, úgynevezett „checkbox-ot” is, melynek segítségével a nem elérhető termékeket listázhatjuk (10. ábra). A jelölőnégyzet a belekattintást követően azonnal szűri a termékeket. Amennyiben kivesszük a pipát, ismét a teljes raktárkészletet láthatjuk.

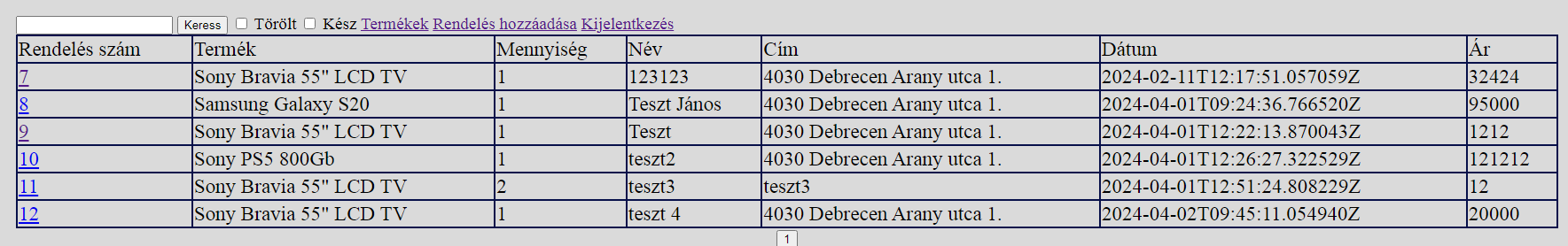


10. ábra: A jelölőnégyzet működése

A felület tartalmaz továbbá egy „Rendelések” nevű gombot, mellyel a rendeléskezelő felületre jutunk el; egy „Termék hozzáadása” gombot, mellyel az adatbázisba vihetünk fel további termékeket, ezzel növelve a raktárkészletet; és egy „Kijelentkezés” gombot, mellyel automatikusan a webshop oldalra jutunk el.

* + 1. Rendelések nyilvántartása

A „Rendelések” gombra kattintva az alábbi ábrán látható felület jelenik meg. (11. ábra).



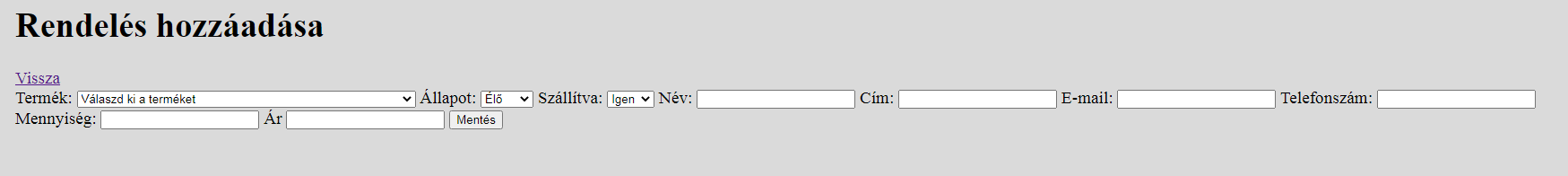
11. ábra: A rendelések nyilvántartó felülete

A felület jelenleg 1db oldalt tartalmaz, de a korábbihoz hasonlóan ez is a táblázatban szereplő termékek számától függ. Ez az oldal is tartalmaz egy keresőmezőt és egy „Keress” feliratú gombot, mellyel szűrhetünk a megrendelt termékek között.

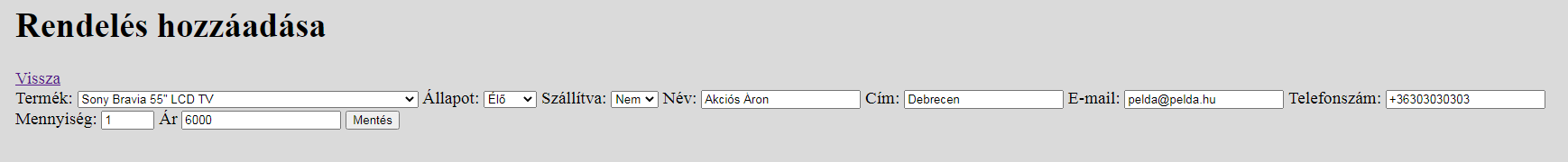
A „Törölt” jelölőnégyzet bepipálásával megtekinthetjük a törölt rendeléseket, a „Kész” jelölőnégyzettel pedig a már elkészült, teljesített rendeléseket listázhatjuk.

A „Rendelés hozzáadása” gombra kattintva az alábbi címre jutunk:

<http://127.0.0.1:8000/add_order/>

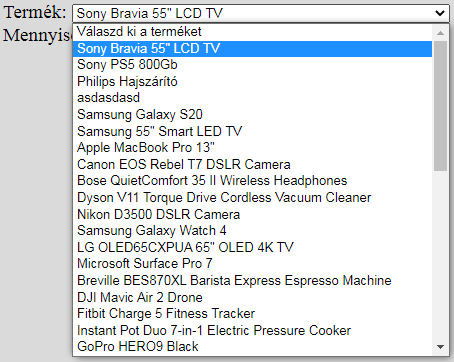


. ábra: Rendelés hozzáadására szolgáló felület



. ábra: Példa rendelés hozzáadására

A rendelés hozzáadása a következők szerint történik. A „Termék:” felirat mellett szereplő lenyíló listából kiválasztjuk az adatbázisban szereplő egyik terméket (14. ábra).



14. ábra: A termékek listája a rendelés felületen

Az „Állapot:” feliratnál lenyíló listából kiválasztjuk a rendelés jellegét, mely lehet „Élő”, vagyis tényleges rendelés, és „Törölt”, vagyis lemondott rendelés.



. ábra: A rendelés állapota

A „Szállítva” feliratnál található lenyíló listából az „Igen” vagy a „Nem” lehetőségek megadásával kiválaszthatjuk, hogy az adott rendelés kiszállításra került-e (16. ábra).



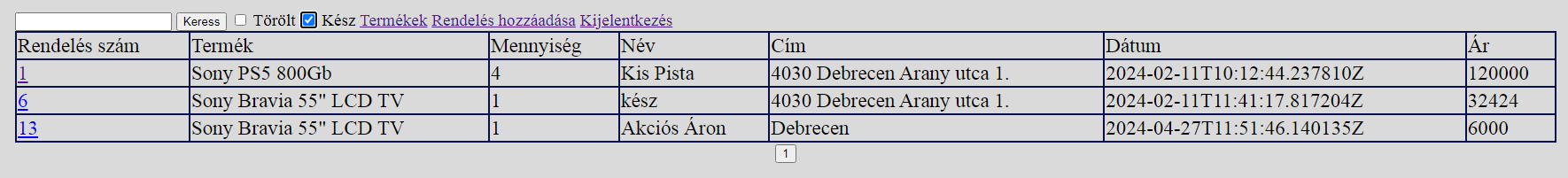
16. ábra: A Szállítva lenyíló lista választható lehetőségei

A többi mező (név, cím, e-mail, telefonszám, mennyiség, ár) kitöltésével a rendelést elkészíthetjük, ezt követően a „Mentés” gombra kattintva véglegesíthetjük. Ezzel a művelettel a következő címre jutunk: <http://127.0.0.1:8000/api/orders/> (17. ábra).



17. ábra: A rendelés véglegesítésére szolgáló felület

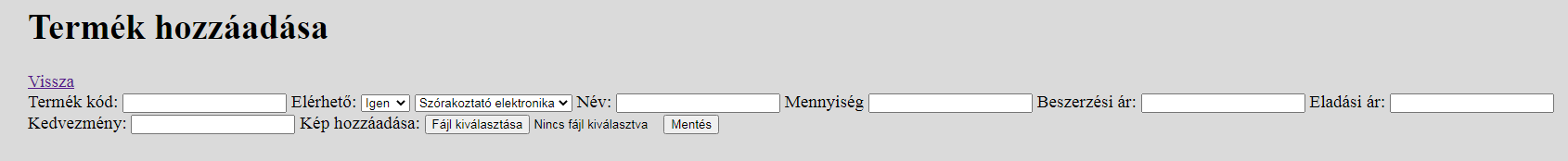
Ismét a „Mentés” gombra kattintva visszajutunk a rendelések nyilvántartó felületére (11. ábra), melynél a „Kész” jelölőnégyzet bepipálását követően az utolsó sorban láthatóvá válik számunkra az előbbiekben felvezetett rendelés (18. ábra).



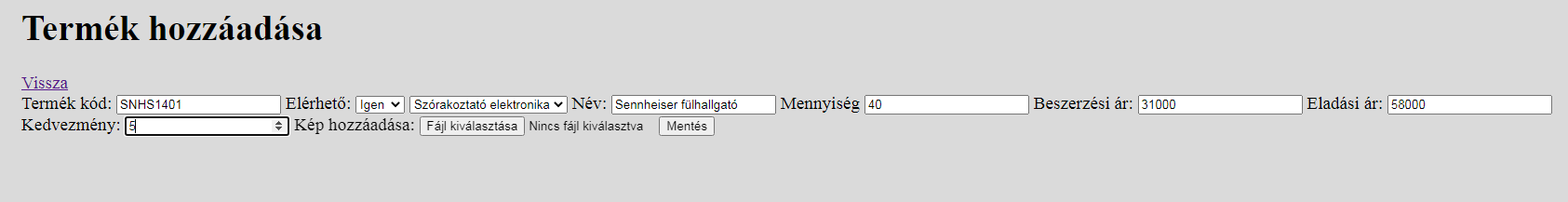
18. ábra: A felvezetett rendelés az utolsó sorban

* + 1. Termékek hozzáadása a raktárkészlethez

Amennyiben a raktárban nyilvántartott termékek listájánál (5. ábra) a „Termék hozzáadása” gombra kattintunk a következő felhasználói felületre érkezünk (19. ábra).

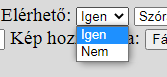


19. ábra: Az adatbázisban szereplő termékek hozzáadására szolgáló felhasználói felület



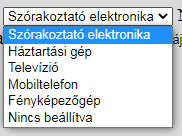
20. ábra: Példa termék hozzáadására

A felületen lehetőségünk van a termék kódjának, nevének, a raktárban szereplő mennyiségének, beszerzési- és eladási árának megadására. Lehetőség van kedvezmény megadására is, mely esetben a program kiszámolja a kedvezményes árat. Az „Elérhető:” feliratnál megadható, hogy a termék adott pillanatban elérhető-e a raktárban vagy nem.



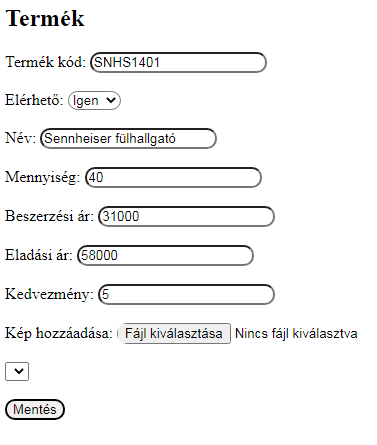
. ábra: A termék elérhetősége a raktárban

Továbbá megadható a termék csoportja is, mely a példában szereplő termék esetén (20. ábra) „Szórakoztató elektronika”. Az elérhető csoportok listáját a következő ábrán figyelhetjük meg (22. ábra).



22. ábra: A raktárban szereplő termék kategorizálása

A felület tartalmaz továbbá egy „Vissza” gombot, mellyel a raktárban szereplő termékek listájához juthatunk vissza (5. ábra) és egy „Mentés” gombot, mellyel a megadott adatok véglegesítésére szolgáló felhasználói felületre érkezünk.



. ábra: A felvezetett termékadatok véglegesítésére szolgáló felület

Az adatok módosítására itt is van lehetőség. Amennyiben az adatokat megfelelőnek találjuk, úgy a „Mentés” gombbal véglegesíthetjük azt.

* 1. Hibajelzések

A Raktárkezelő weboldal képes lekezelni a helytelen használatból származó hibákat, ilyenkor egy üzenetet küld a felhasználónak. Erre mutat egy példát az alábbi kódrészlet.

@api\_view(['GET'])

def item\_view(request,pk):

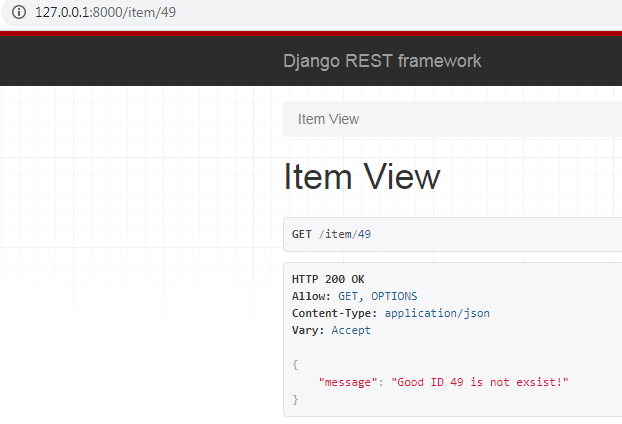
    try:

        good = Good.objects.get(id=pk)

    except:

        return Response({'message':f'Good ID {pk} is not exsist!'})

A fenti kódrészletben látható, hogy a „Good ID … is not exist!” üzenet jelenik meg amennyiben a webböngésző URL címébe a termék azonosítóját kézzel vinnénk fel és az adott termék nem létezik még az adatbázisban.



. ábra: Hibaüzenet, amennyiben egy termék nem szerepel az adatbázisban

* 1. Az admin felhasználó lehetőségei

A Raktárkezelő program admin felhasználója a teljes adatbázishoz hozzáfér. Lehetősége van az adatbázishoz adatot hozzáadni, abban adatot törölni, módosítani, valamint rendelést leadni és törölni.

1. Információkérés lehetősége

A Raktárkezelő program működésével kapcsolatos észrevételek, megjegyzések, problémák, valamint az oldal működésével kapcsolatban felmerülő kérdések esetén kérjük, hogy keressenek minket az alábbi elérhetőségek valamelyikén.

Kovács Tamás: kovacs.tamas8912@gmail.com

Barcza Máté: barcza.mate.korny@gmail.com

Fehér Ádám: fadam1026@gmail.com

1. Összefoglalás, köszönetnyilvánítás

Összefoglalásképpen elmondható, hogy a Raktárkezelő program valóban megkönnyíti a nagy mennyiségű-, különböző kategóriájú termékek nyilvántartását, naprakészen tartását. Különösen hasznos szerepet kaphat a mai globalizált világban.

Öröm volt számunka hárman egy csapatban dolgozni. A csapatmunka által jobban megismerhettük egymást és a felmerülő problémákon is könnyebben juthattunk túl. A telefonon és az interneten történő kapcsolattartás mellett különösen fontos szerepet tulajdonítottunk a személyes találkozóknak is, melyeken megbeszéltük a projektfeladat jelenlegi állapotát, a felmerülő akadályokat és a lehetséges megoldásaikat.

A Raktárkezelő program nem jöhetett volna létre a Debreceni SZC Mechwart András Gépipari és Informatikai Technikum oktatóinak segítsége nélkül, akik közül néhányan közvetlen, néhányan pedig közvetett formában támogattak minket a projektfeladat elkészítése során. Így ezúton is szeretnénk megköszönni Nemes Tamásnak, Jánosi Jánosnak, Magi Bélának, Fekete Balázsnak, Kiss Albertnek és Szegedi Imrének, hogy odaadó munkájukkal és rendelkezésre állásukkal megismertették velünk a Szoftverfejlesztő és -tesztelő szakmát és elősegítették a jelen dokumentációban szereplő Raktárkezelő program létrejöttét.

1. Irodalomjegyzék
2. Django keretrendszer: <https://docs.djangoproject.com/en/5.0/>
3. Program szintaktika, kialakítás: <https://www.w3schools.com/>