**Záródolgozat**

Képzésben részt vevő személy neve: Lőrincz Gergely Tamás

Képzés: felnőttek szakmai képzése

Szak: Webfejlesztő

**A záródolgozat címe:**

**Power Chord Vinyls dokumentáció**

Konzulens: Sándor László

Beadási határidő: 2024. 06. 21.

Győr, 2024. 05. 30

**Módos Gábor**igazgató

**Tulajdonosi nyilatkozat**

Ez a dolgozat a saját munkám eredménye. Dolgozatom azon részeit, melyeket más szerzők munkájából vettem át, egyértelműen megjelöltem.

Ha kiderülne, hogy ez a nyilatkozat valótlan, tudomásul veszem, hogy a szakmai vizsgabizottság a szakmai vizsgáról kizár és szakmai vizsgát csak új záródolgozat készítése után tehetek.

Győr, 2024. június 21.

képzésben részt vevő személy aláírása

Power Chord Vinyls dokumentáció

Tartalomjegyzék

Tartalom

[Témamegjelölés 5](#_Toc169887499)

[Alkalmazás célja: 5](#_Toc169887500)

[Az alkalmazás főbb funkciói: 6](#_Toc169887501)

[Felhasznált technológiák 8](#_Toc169887502)

[Backend 8](#_Toc169887503)

[Adatbázis-kezelő rendszer 8](#_Toc169887504)

[Előnyei: 8](#_Toc169887505)

[Hátrányai: 8](#_Toc169887506)

[Frontend 9](#_Toc169887507)

[Fejlesztés menete 10](#_Toc169887508)

[Az adatbázis megtervezése és feltöltése 10](#_Toc169887509)

[A backend végpontok tervezése és megvalósítása 10](#_Toc169887510)

[A frontend tervezése és megvalósítása 10](#_Toc169887511)

[Az adatbázis felépítése: Egy a többhöz kapcsolat 10](#_Toc169887512)

[Validálás, hibaüzenetek 12](#_Toc169887513)

[Fő oldal 13](#_Toc169887514)

[A regisztrációs oldal 14](#_Toc169887515)

[A bejelentkezés oldal 15](#_Toc169887516)

[A kosár 16](#_Toc169887517)

[A rólunk oldal 17](#_Toc169887518)

# Témamegjelölés

## Alkalmazás célja:

Manapság sérülésmentes bakelit lemezt vásárolni nem a legegyszerűbb dolog. Nem éppen aktuális technológia, ezért akik továbbra is szeretnek ezzel a módszerrel zenét hallgatni, vagy netán csak gyűjtői értéke miatt próbálják beszerezni, azoknak nehéz dolguk van. A Power Chord Vinyls eredeti régi lemezeket, valamint utángyártott, frissen készült lemezeket is gyárt, felhasználóknak és gyűjtőknek egyaránt, megcélozva a jó állapotban lévő bakelit lemezek piaci rését.

Az alkalmazás használói elsősorban ahogy említettem a független gyűjtők, és a technológia fennálló használói, de dekoratív célból studiókban ugyanúgy fellelhető általában egy jó pár bakelit lemez. Az alkalmazás arra lett tervezve, hogy nekik eladja a mai napokban már egyre kevesebb helyen megvásárolható terméket.  
  
Az elkészülés pillanatában az alkalmazás képes lesz:

* Számok listázására különböző előadoktól
* Lebonyolítsa a felkeresett számok eladását
* Új előadók és hozzájuk fűzött számok képpel történő feltöltésére (admin felhasználóknak)
* Elmenteni lemezeket egy kosárba
* A kosár kiürítése vásárlás után
* Felhasználók regisztrációját és belépését kiszolgálni

## Az alkalmazás főbb funkciói:

Előadók számainak kilistázása

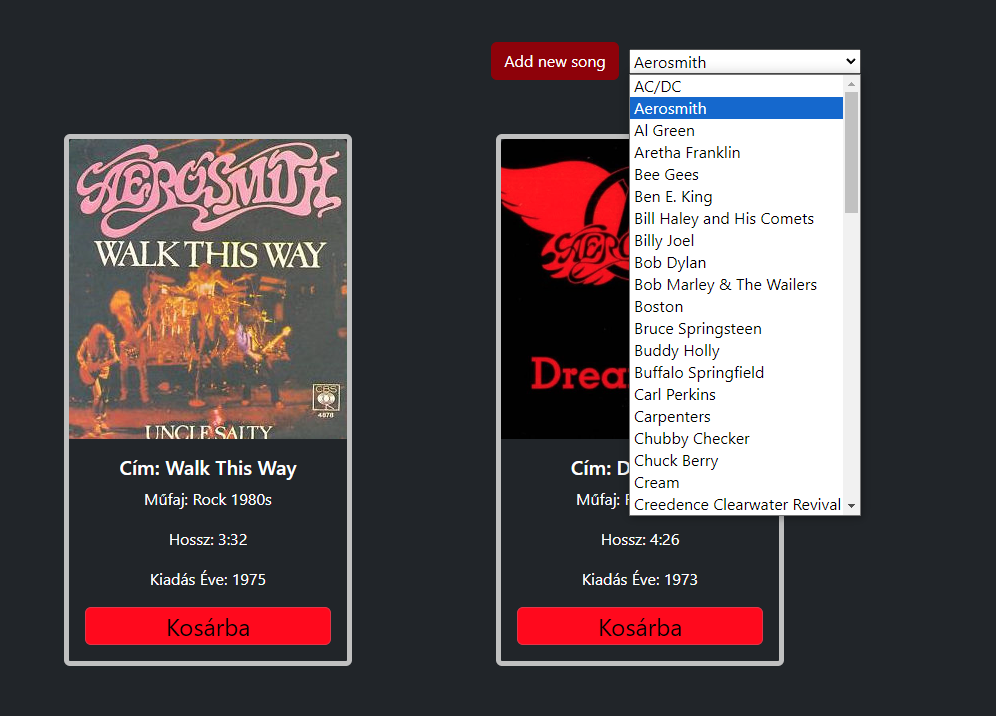
Az előadó kiválasztása legördülő menüből lehetséges.

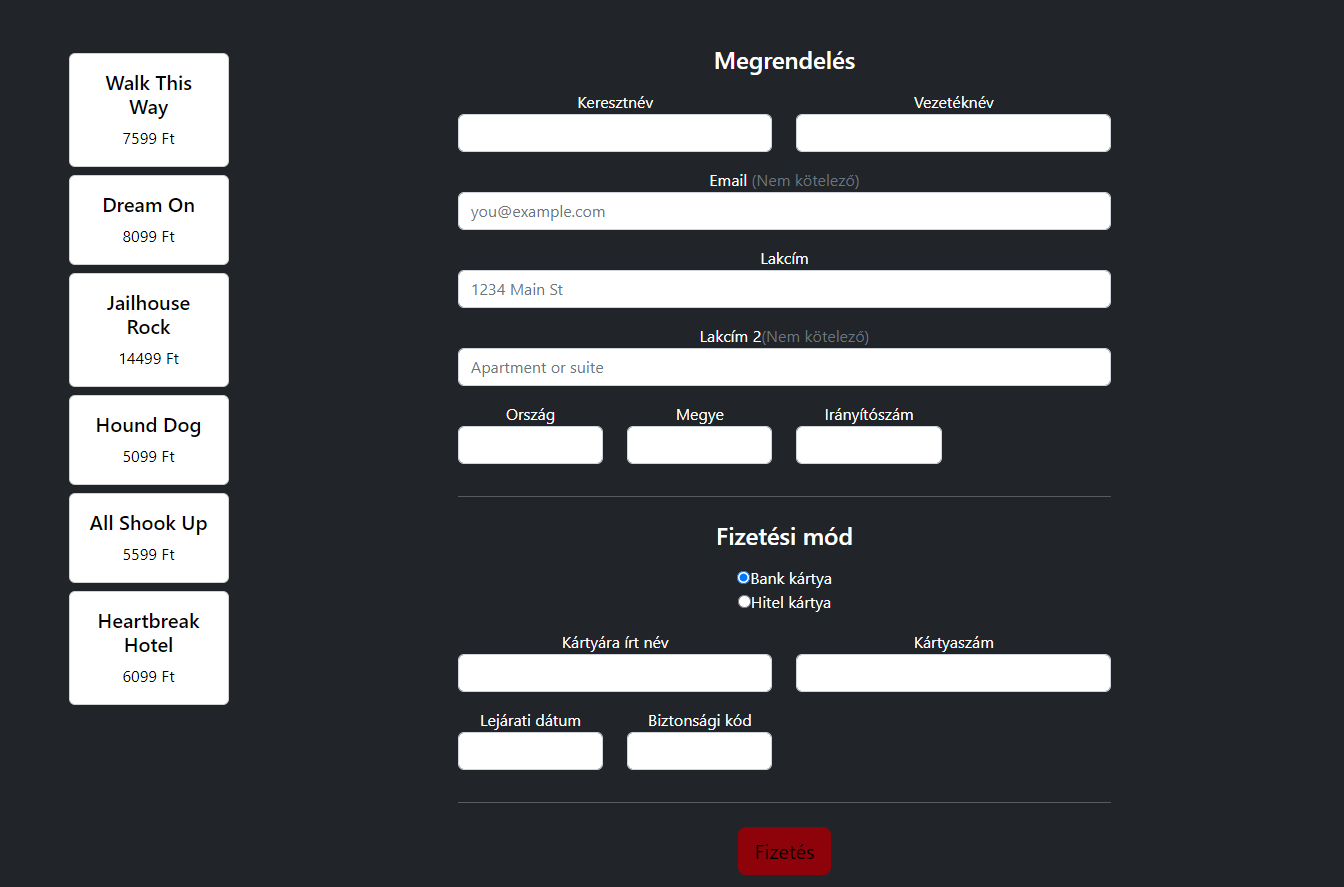
Új előadó illetve új zene felvitele.

Zene és/vagy előadó törlése.

Zene és/vagy előadó módosítása.

Zeneszám megvásárlása bakelit lemezen





## Felhasznált technológiák

### Backend

PHP programozási nyelv alapokon Laravel keretrendszer segítségével történt a backend programozása.

### Adatbázis-kezelő rendszer

A backend alkalmazásunk adatbázis motorjának MySQL adatbázis kezelő rendszert választottam. Mivel az egyik, talán maga a legelterjedtebb rendszer, könnyen található hozzá segítség, ha valamit nem tudnék megvalósítani, valamint általában stabilan is működik.

### Előnyei:

Skálázhatóság:

A relációs adatbázisok jól skálázhatóak vertikálisan (nagyobb teljesítményű szerverek használatával) és bizonyos mértékig horizontálisan is.

Adatlekérdezés

Az SQL (Structured Query Language) használata egyszerűvé és hatékonnyá teszi az adatok lekérdezését, manipulálását és kezelését.

Adatbiztonság:

A relációs adatbázisok erős biztonsági mechanizmusokat kínálnak, beleértve a felhasználói jogosultságokat és a hozzáférés-szabályozást.

Kapcsolatok kezelése:

A relációs adatbázisok jól kezelik a különböző adatok közötti kapcsolatokat, mivel táblák és kulcsok segítségével hatékonyan tudják összekapcsolni az adatokat.

### Hátrányai:

Komplexitás:

A relációs adatbázisok bonyolultak lehetnek, különösen nagy és összetett adatstruktúrák kezelése esetén.

Merevstruktúra**:**

A relációs adatbázisok struktúrája merev, ami nehézzé teszi az adatbázis sémájának változtatását vagy kiterjesztését a jövőbeni igények szerint.

Teljesítmény:

Nagy mennyiségű adat és komplex lekérdezések esetén a relációs adatbázisok teljesítménye csökkenhet, ami lassabb válaszidőkhöz vezethet.

Költség:

A relációs adatbázisok, különösen a nagy vállalati megoldások, drágák lehetnek mind licencelés, mind karbantartás szempontjából.

## Frontend

A VueJs könyvtárak sok más keretrendszerhez képest kis méretűek, így gyorsabban tölt be kliens oldalon a weboldal. Valamint a keretrendszer megkönnyíti a programozás részét is a fejlesztésnek, ugyanis míg sok más esetben külön fájlban kell megírnod a kódot, VueJs-ben az adott view ablakában elhelyezhető, és sokkal kevesebb időt igényel így a fejlesztés ennek a része.

A keretrendszer a javascript és css integrálását is nagyon megkönnyítette, az előző oldalon említett okok miatt. Az oldal javarészt bootstrapes design sablonokkal működik, itt-ott pár változtatással, vagy néhány esetben kompletten átalakított, vagy nulláról felépített design részlettel, hogy ne legyen teljes mértékben egy bootstrap modulok másolatainak a gyüjteménye.  
  
A backend adatait a betöltés után javascriptet dolgozza fel az oldal, valamint használja és végez vele műveleteket ott, ahol épp szükséges.

# Fejlesztés menete

A fejlesztés végig a GitHub-on történt, mivel fontosnak tartom az alkalmazás verziókövetését.

## Az adatbázis megtervezése és feltöltése

Az adatbázis tartalma folyamatosan frissült, és egyes helyzetekben, kiegészült új oszlopokkal is, amikor épp szükség volt rá. A songs és artists táblák léteztek először, és később kiegészült egy users, végül meg a kosár táblájával. Egy előre nem látott elnevezési hiba miatt laravelben, a shoppingcart table-t át kellett neveznem shopping\_carts-ra, hogy elérjék a kiküldött requestek. Meg lehetett volna oldani másképpen is a problémát, de ez volt a legkönnyebb, legkézenfekvőbb megoldás.

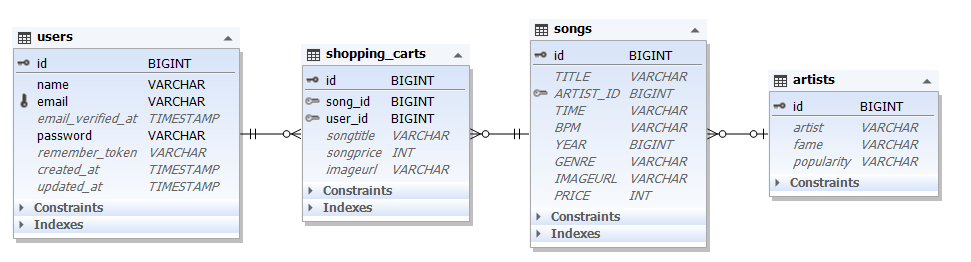
## A backend végpontok tervezése és megvalósítása

A tervezés fázisban minden végponthoz apiResource-ot használtam, de amint elértem az oldalnak a végleges állapotát, a szükségtelen route-októl megszabadultam. Először mindig a route-okat, majd utána a hozzá szükséges, illetve a velük elért funkciókat valósítottam meg. Ez alól csak pár dolog képez kivételt, amikor frontend fejlesztés közben jutott eszembe, hogy még szükség lenne valamire. Legelső sorban a songs és artists route-ját hoztam létre, valamint ahol szükséges, korlátoztam vendég, bejelentkezett user, és admin user-re a részeit. A shoppingcart elérése volt a következő, és legvégül miután leteszteltem hogy maga a funkció működik, elkezdtem lekorlátozni megjelenítést úgy, hogy a gombot csak bejelentkezett user lássa, valamint nem admin esetében, csak a saját id-jéhez tartozó kosár elemeket. Így létrejött a register, login, és logout route, valamint annak csatolt elemei.

## A frontend tervezése és megvalósítása

A frontend nagyrészt a backend haladását követte kisebb-nagyobb lemaradásokkal, de volt egy pár ritka eset is, amikor az járt előrébb. Ez legtöbbször olyankor fordult elő, amikor hirtelen eszembe jutott valami ami még kellhet az oldalra, és így először a felületet csináltam meg hozzá, és csak azután valósítottam meg a hozzá tartozó backend részt. Miután lefejlesztettem az átlag felhasználók által elérhető oldalt, jöhettek az admin funkciók, mint pl. az új zene hozzáadása.

# Az adatbázis felépítése: Egy a többhöz kapcsolat



A mezők típusai és megszorításai:

songs.TITLE: varchar(1024)

songs.ARTIST\_ID: big int(20)

songs.TIME: varchar(1024)

songs.BPM: varchar(1024)

songs:YEAR: big int(20)

songs.GENRE: varchar(1024)

songs.IMAGEURL: varchar(255)

songs.PRICE: int(11)

artists.artist: varchar(1024)

artists.fame: varchar(1024)

artists.populairty: varchar(1024)

shopping\_carts.song\_id: big int(20), kötelező

shopping\_carts.user\_id: big int (20), kötelező

shopping\_carts.songprice: int (11)

shopping\_carts.songtitle: varchar (255)

shopping\_carts.imageurl: varchar (255) (Technikai okok miatt végül nem használt adat)

users.name: varchar (255), kötelező

users.email: varchar (255), kötelező

users.password: varchar (255), kötelező

personal\_access\_tokens.tokenable\_type: varchar (255) , kötelező

personal\_access\_tokens.tokenable\_id: big int (20) , kötelező, unsigned

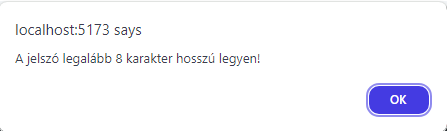
personal\_access\_tokens.name: varchar (255) , kötelező

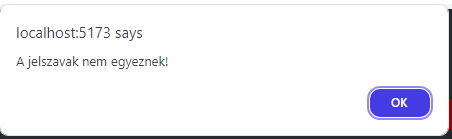
personal\_access\_tokens.token: varchar (64) , kötelező

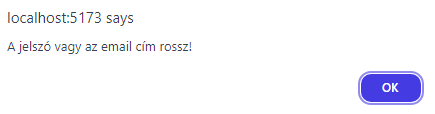
# Validálás, hibaüzenetek

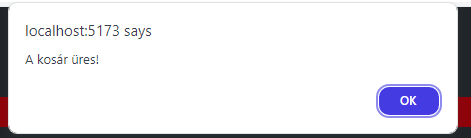
A backend oldalon requestekben megfogalmazott megkötések ellenőrzik, hogy elfogadható adatot akar-e valaki bevinni az adott mezőbe. A protected $fillabe-ben lévő adatok csak, amik szerkeszhetőek, illetve megadhatóak.

A jelszó hosszának és egyezésének meglétét a frontend és a backend is ellenőrzi.



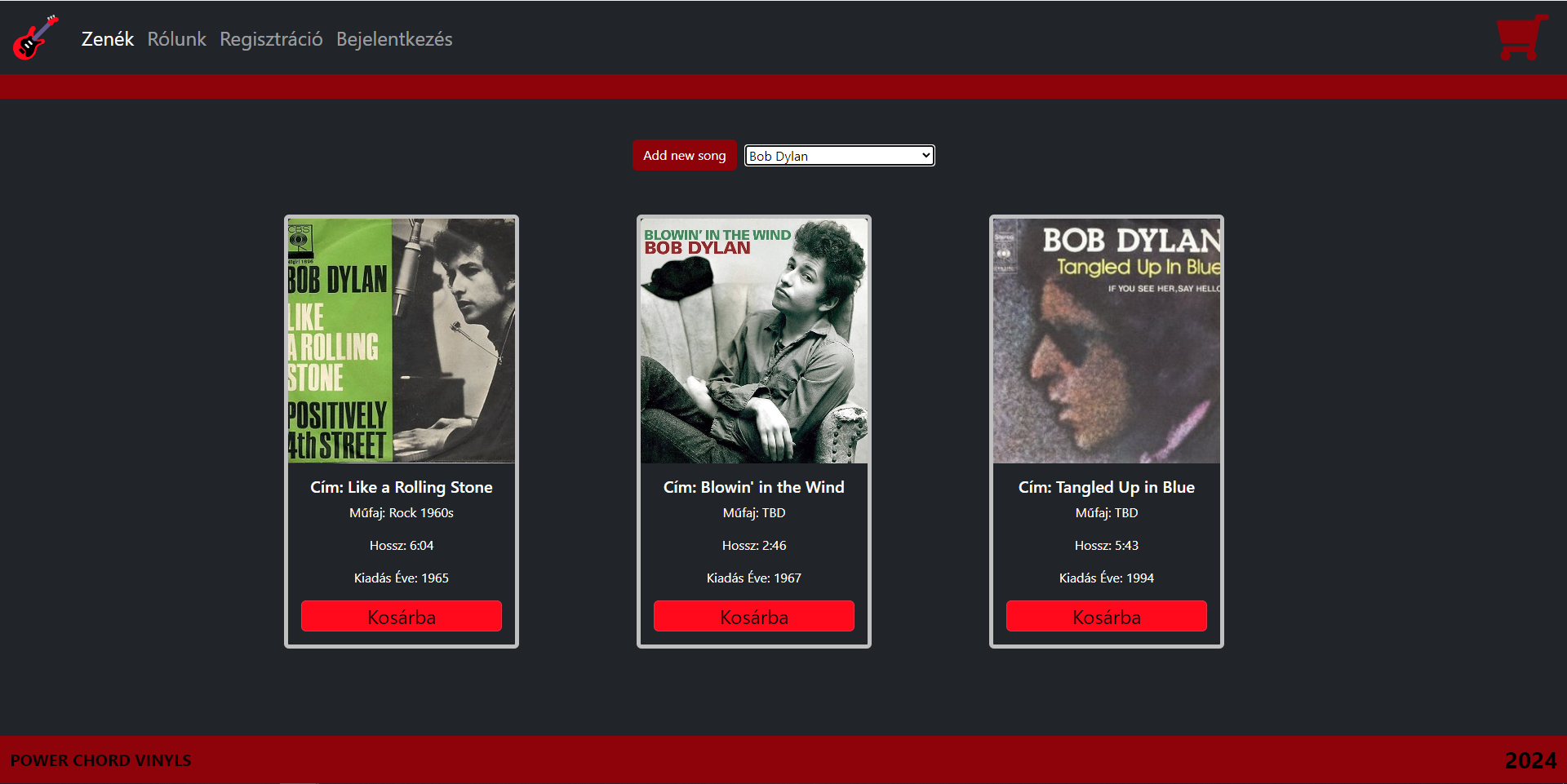






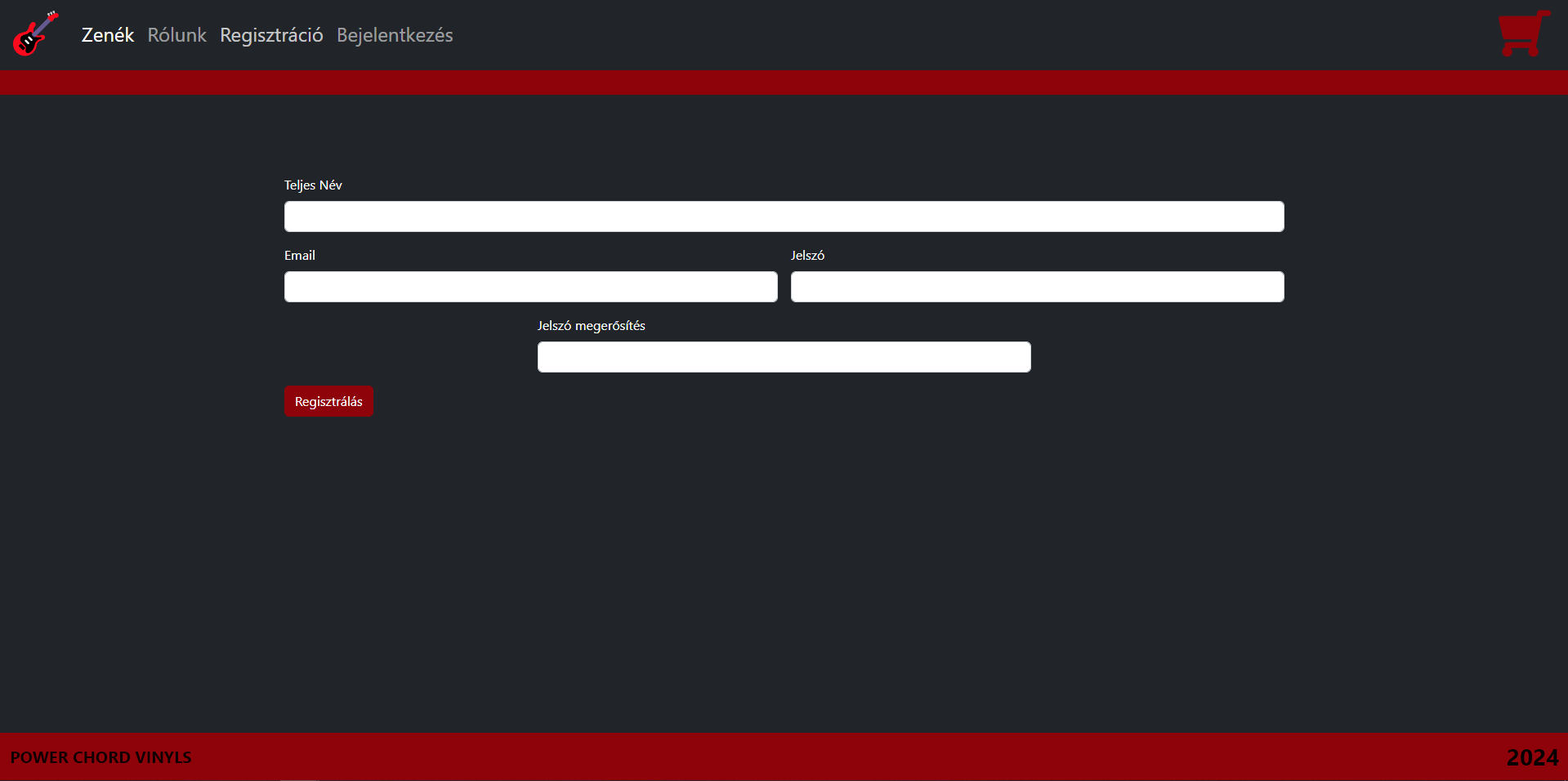
# Fő oldal

Egy képernyőkép és a működés leírása



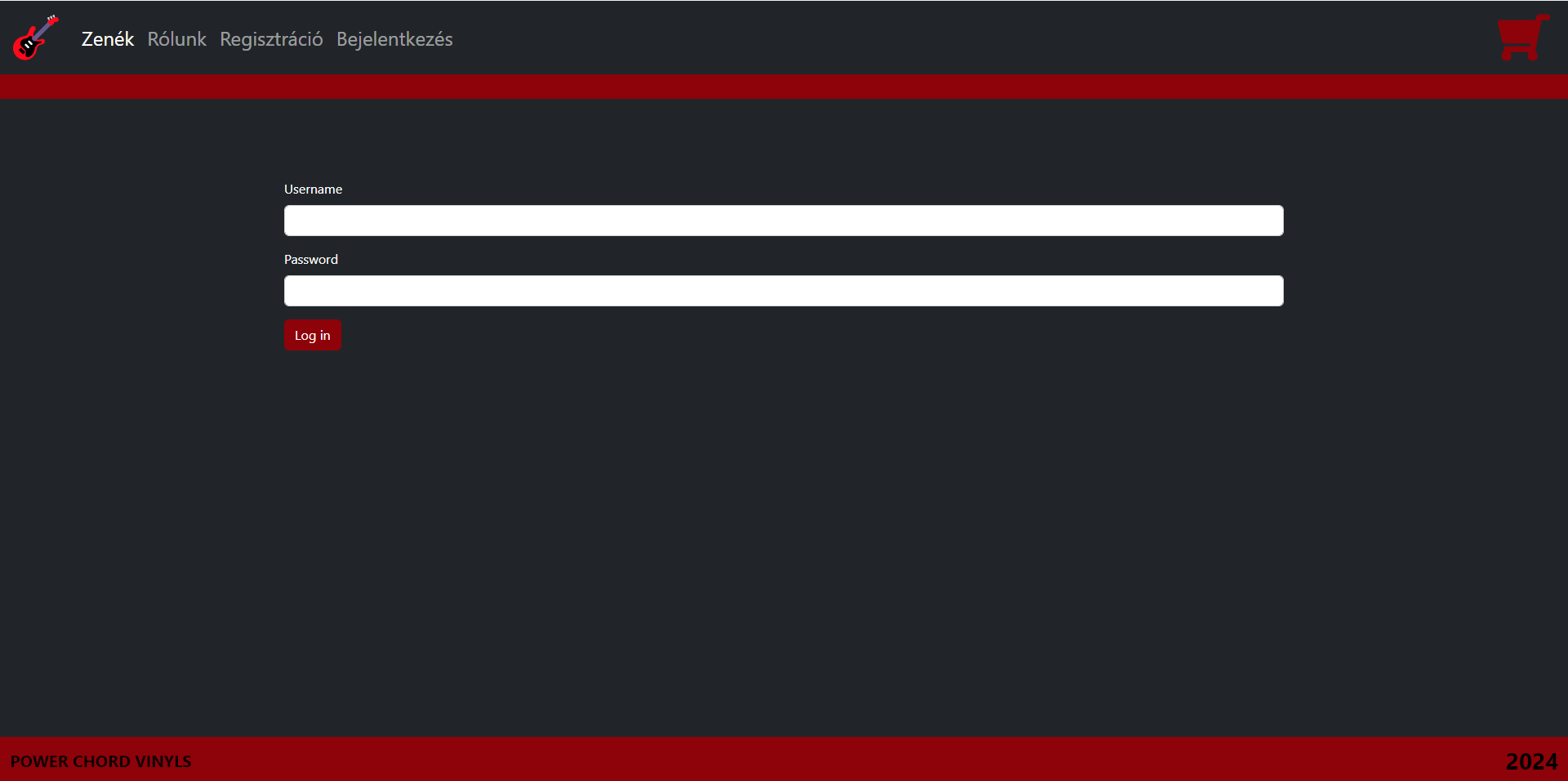
1. A zenék menüpont a fő oldalra visz
2. A kosár ikon a kosár oldalra visz, ahol az oda betett termékek el vannak tárolva, user\_id kapcsolja őket különböző felhasználókhoz. Minden user csak a saját elemeit látja.
3. A regisztráció tartalmazza az új felhasználó létrehozás felületét.
4. A bejelentkezés már regisztrált felhasználóknak van, belépés után érhetik el az oldal további funkcióit.
5. A Rólunk menüpontra kattintva beleolvashat a cég létrejöttének céljába, és egy rövid ismertetőt is elolvashat.

# A regisztrációs oldal



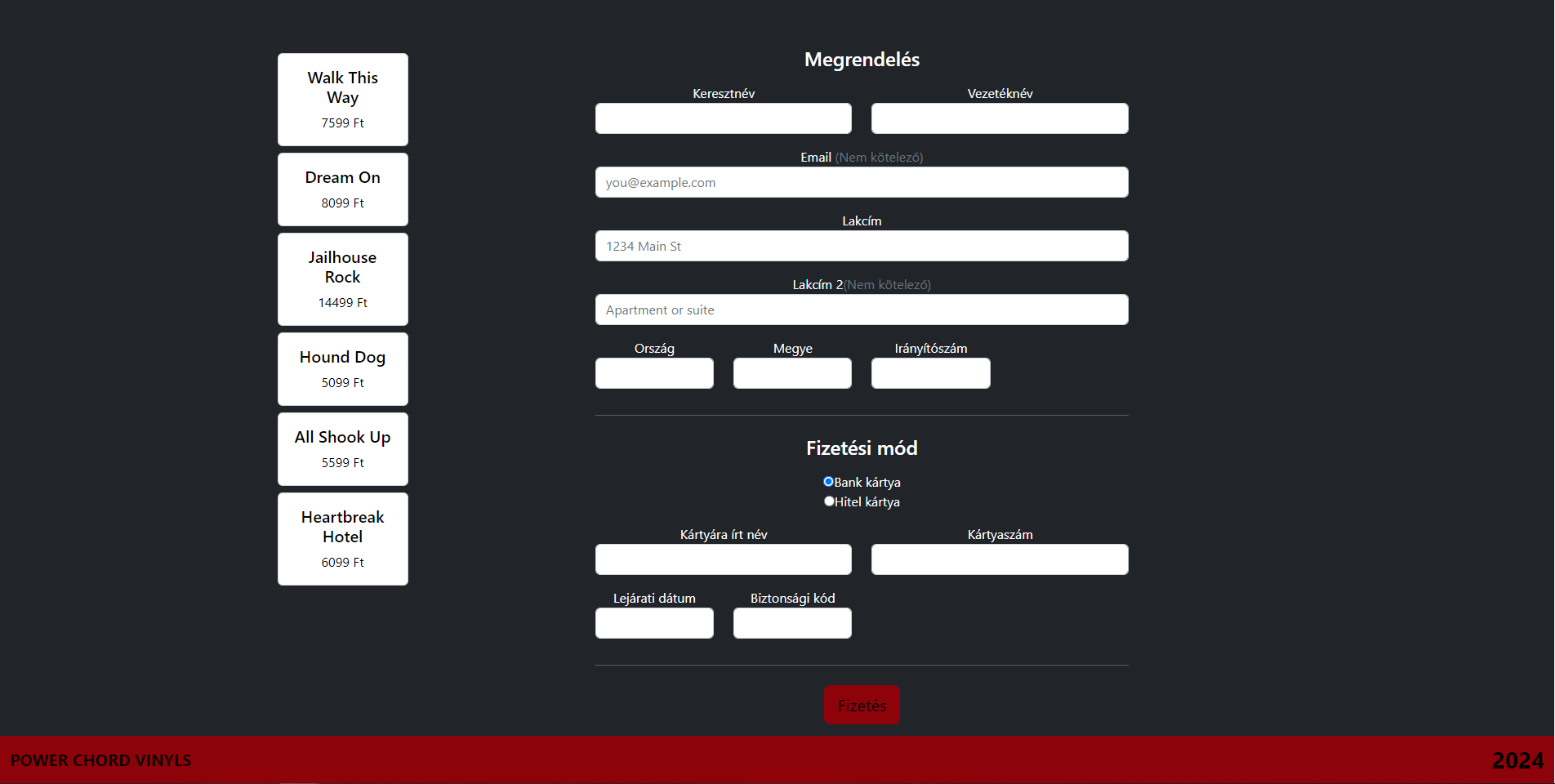
A regisztrációs oldal, ahol új felhasználót lehet létrehozni. A jelszónak legalább 8 karakterből kell állnia, és egyeznie kell a megerősítő mezőben megadott stringel. Egy email címet csak egy regisztrációnál lehet használni, a következőnél már nem fogja engedni.

# A bejelentkezés oldal



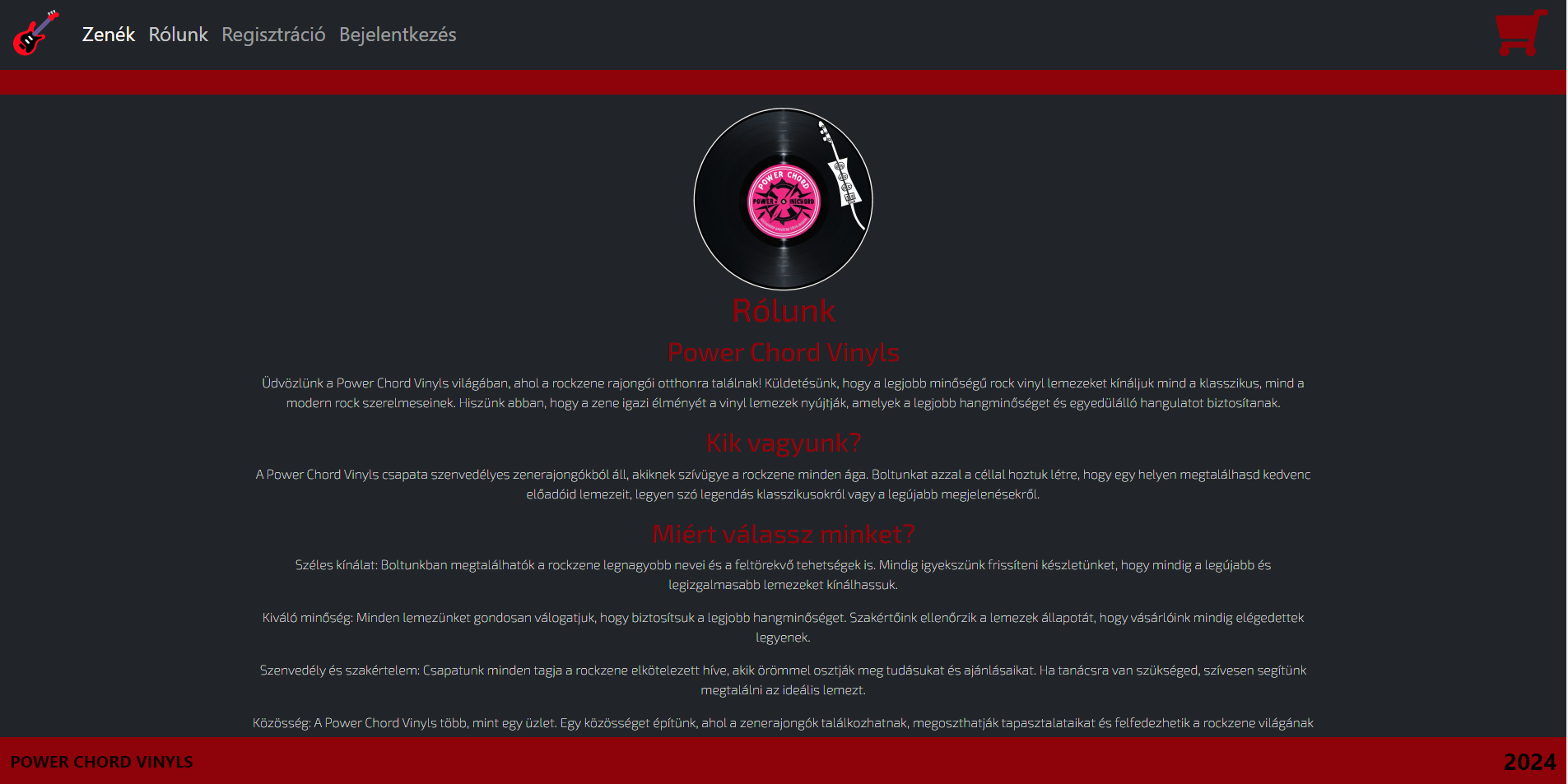
Ezen az oldalon jelentkezhet be a felhasználó, és ezután érheti el a saját kosarát, valamint tehet bele új elemeket. Admin felhasználó esetében az új zene hozzáadása a kínálathoz is megnyílik.

# A kosár



Ez a kosár oldal, ahova az ide betett elemek vásárlását lehet véglegesíteni. Egy vásárlás lebonyolítása és utalás megerősítése után a kosárba tett elemek automatikusan törlődnek.

# A rólunk oldal



Pár információt tartalmaz az oldalról, valamint a vásárló itt tájékozódhat, hogy mit kínál maga a bolt.