

Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Irányítástechnika és Informatika Tanszék

Gaál Réka Dorottya

Szoftverfejlesztés .net platformon

Webfejlesztés .NET Blazor keretrendszeren MatBlazor külső komponens könyvtár segítségével

Konzulens:

Dr. Simon Balázs

BUDAPEST, 2022

Tartalomjegyzék

[Összefoglaló 5](#_Toc101888628)

[1 Frontend modulok 6](#_Toc101888629)

[1.1 Pages 6](#_Toc101888630)

[1.1.1 Cart 6](#_Toc101888631)

[1.1.2 Home 6](#_Toc101888632)

[1.1.3 IceCream 6](#_Toc101888633)

[1.1.4 Login 7](#_Toc101888634)

[1.1.5 Product 7](#_Toc101888635)

[1.1.6 Profile 7](#_Toc101888636)

[1.1.7 SignUp 7](#_Toc101888637)

[1.2 Shared 8](#_Toc101888638)

[1.2.1 MainLayout 8](#_Toc101888639)

[2 BackEnd modulok 9](#_Toc101888640)

[2.1 Pages 9](#_Toc101888641)

[2.1.1 Home.razor.cs – public partial class Home 9](#_Toc101888642)

[2.1.2 IceCream.razor.cs – public partial class IceCream 9](#_Toc101888643)

[2.1.3 Login.razor.cs – public partial class Login 10](#_Toc101888644)

[2.1.4 Product.razor.cs – public partial class Product 10](#_Toc101888645)

[2.1.5 SignUp.razor.cs – public partial class SignUp 10](#_Toc101888646)

[2.2 Shared 11](#_Toc101888647)

[2.2.1 MainLayout.razor.cs – public partial class MainLayout 11](#_Toc101888648)

[2.3 További modulok 11](#_Toc101888649)

[2.3.1 Allergen.cs – public class Allergen 11](#_Toc101888650)

[2.3.2 Card.cs – public class Card 12](#_Toc101888651)

[2.3.3 CartElement.cs – public class CartElement 12](#_Toc101888652)

[2.3.4 GenderDef.cs – public class GenderDef 13](#_Toc101888653)

[2.3.5 Global.cs – public static class Global 13](#_Toc101888654)

[3 Felhasznált MatBlazor komponensek 14](#_Toc101888655)

[3.1 Felugró értesítések 14](#_Toc101888656)

[3.1.1 SnackBar 14](#_Toc101888657)

[3.2 Form Control-ok 14](#_Toc101888658)

[3.2.1 Checkbox 14](#_Toc101888659)

[3.2.2 DatePicker 14](#_Toc101888660)

[3.2.3 NumericUpDownField 14](#_Toc101888661)

[3.2.4 Select 14](#_Toc101888662)

[3.2.5 Slider 14](#_Toc101888663)

[3.2.6 TextField 14](#_Toc101888664)

[3.2.7 Validation via EditContext 14](#_Toc101888665)

[3.3 Gombok 15](#_Toc101888666)

[3.3.1 Button 15](#_Toc101888667)

[3.3.2 Icon 15](#_Toc101888668)

[3.3.3 IconButton 15](#_Toc101888669)

[3.4 Layout-ok 15](#_Toc101888670)

[3.4.1 Card 15](#_Toc101888671)

[3.4.2 Divider 15](#_Toc101888672)

[3.4.3 Layout Grid 15](#_Toc101888673)

[3.4.4 List 15](#_Toc101888674)

[3.4.5 Paper 15](#_Toc101888675)

[3.4.6 Themes 15](#_Toc101888676)

[3.4.7 Typography 15](#_Toc101888677)

[3.5 Navigáció 16](#_Toc101888678)

[3.5.1 AppBar 16](#_Toc101888679)

[3.5.2 ButtonLink 16](#_Toc101888680)

[3.6 Táblázatok 16](#_Toc101888681)

[3.6.1 Table 16](#_Toc101888682)

[Cart 16](#_Toc101888683)

[IceCream 16](#_Toc101888684)

[Login 17](#_Toc101888685)

[Product 17](#_Toc101888686)

[Profile 17](#_Toc101888687)

[SignUp 17](#_Toc101888688)

[MainLayout 18](#_Toc101888689)

[4 Az elkészült webalkalmazás - képgaléria 19](#_Toc101888690)

Összefoglaló

A feladat egy .NET Blazor alapú weboldal elkészítése volt a MatBlazor külső komponens könyvtár minél széleskörűbb felhasználásával és tesztelésével. Ennek érdekében egy Webshophoz hasonló webes felületet valósítottam meg, ahol főleg a komponensek változatos felhasználására törekedtem a funkciók tényleges megvalósítása helyett.

A Webshop fagylaltokat árul, áll egy üdvözlő oldalból, főoldalból, ahol a fagylaltválaszték látható szűrhető módon, a fagylaltok önálló oldalaiból, profil oldalból bejelentkező és regisztrációs felülettel együtt, és a kosárból. Ezen oldalak egységes megjelenését egy állandó fej- és lábléc biztosítja.

Összességében úgy érzem, a feladatból rengeteget tudtam tanulni és fejleszteni magamat. Egy teljesen új technológiával ismerkedhettem meg, ami nem csak korszerű, de rengeteg lehetőséget is rejt magában a fejlesztők számára. Korábban nem fejlesztettem webes felületet .NET platformon, de a Blazor egy olyan keretrendszert biztosít ehhez, ami nagyban megkönnyítette és élvezhetővé tette a tanulást, a komponensekben mutatott hiányosságait pedig széles körben pótolják a hozzá elérhető külső komponenskönyvtárak.

Ezen pozitív tapasztalatok birtokában és a konzulensemtől, Dr. Simon Balázstól kapott segítségnek, iránymutatásnak és türelemnek köszönhetően egy tartalmas félévet tudhatok magaménak az Önálló laboratórium tárgy keretein belül és szívesen fogom bővíteni a későbbiekben akár önszorgalomból is az itt megalapozott tudásomat a témakörben.

# Frontend modulok

Az általam fejlesztett frontend modulok felsorolása és rövid ismertetésük. Az egyes modulokhoz tartozó .razor file-ok tartalmazzák a megjelenítésért felelős HTML kódot és a .css file-ok tartalmazzák a HTML kódban felhasznált saját stílusokat.

## Pages

### Cart

Ez az osztály valósítja meg a kosárban található elemek megjelenítését. Tartozik hozzá egy .razor és egy .css file.

**.razor:** Kosárelemenként áll egy sorból, ami tartalmazza az elem mennyiségét állítható módon, nevét, árát és egy eltávolítás gombot. Legalul jobb oldalt látható a teljes ár.

**.css:** Található benne egy formázás az árak megjelenítésére és a teljes összeg megjelenítésére.

### Home

Ez az osztály valósítja meg az üdvözlő oldal megjelenítését. Tartozik hozzá egy .razor, egy .css file.

**.razor:** Egy négyzetrácsos elrendezésben tartalmaz egy üdvözlő szöveget, alatta az oldalhoz tartozó leírást, legalul pedig a termékkategóriákat tartalmazó három kártyaként formázott div blokk.

**.css:** Tartalmaz egy stílust a az üdvözlő szövegre, az oldal leírását tartalmazó szövegre és itt található a három termékkategóriát tartalmazó blokk formázása is.

### IceCream

Ez az osztály valósítja meg a fagyik listájának megjelenítését egy szűrési felülettel együtt. Tartozik hozzá egy .razor és egy .css file.

**.razor:** Egy négyzetrácsos elrendezés bal oldalán találhatók a szűrési feltételek, jobb oldalt pedig a fagyikat tartalmazó kártyák. A szűrők közt található egy checkbox csak a készleten lévő elemek megjelenítésére. Alatta található két csuszka a minimum és a maximum ár beállítására dinamikusan. Legalul végezetül megtalálhatók a fagyik színére szűrő checkbox-ok. Bal oldalt a kártyákat egy foreach ciklus jeleníti meg, minden fagyira egy új cellát hoz létre és elhelyezi benne a fagyi adataival kitöltött kártyát.

**.css:** Tartalmaz egy stílust a szűrőket tartalmazó cellára, a kártyákra és egy stílust, amit akkor alkalmazunk egy kártyán, ha a szűrési feltételek alapján azt nem szeretnénk megjeleníteni.

### Login

A bejelentkező felületet megvalósító osztály. Egy .razor file-ból áll.

**.razor:** Az oldal címe alatt található a két beviteli mező a felhasználónévnek és a jelszónak. Ezalatt található a bejelentkezés gomb, ami alatt a regisztrációs oldalra irányító gomb, amennyiben a felhasználó még nem rendelkezik fiókkal. Ha a felhasználó rossz felhasználónévvel vagy jelszóval próbál bejelentkezni, egy snackbar üzenet figyelmezteti erre.

### Product

Az egyes fagyikat egyenként megjelenítő rendelési felület. Tartozik hozzá egy .razor és egy .css file.

**.razor:** A termék neve alatt két oszlopban jeleníti meg a termék információkat és a rendelési felületet. Bal oldalt látható a termék képe, alatta pedig a rendelési mennyiség és annak az ára. Jobb oldalt található az allergén táblázat és a kosárba tétel gomb. Amennyiben a termék nincs készleten, a gomb helyett a Sold out felirat jelenik meg.

**.css:** A kép szélességét megadó stílus.

### Profile

A felhasználói profilt megjelenító oldal. Egy .razor file-ból áll.

**.razor:** Az oldal címe alatt két oszlop található. Bal oldalt látható a profilkép , jobb oldalt pedig a felhasználó adatai egy listában vonallal elválasztva.

### SignUp

A regisztrációs oldalt megjelenítő modul. Egy .razor file-ból áll.

**.razor:** Az oldal címe alatt találhatók egymás alatt sorban a beviteli mezők: felhasználónév, jelszó, születési dátum kiválasztása, nem. A beviteli mezők alatt található két gomb. Az egyikkel regisztrálni lehet, a másikkal pedig kiüríteni a már kitöltött mezőket. Végezetül található a file alján egy validációs kód, ami vizsgálja, hogy minden mező ki van-e töltve, a jelszó legalább 8 karakteres-e és a születési dátum nem későbbi-e, mint a mai dátum. Ehhez tartozik egy SignUpModell osztály, ami a beviteli mezők változóit adja meg követelményekkel együtt. Ha sikeres a regisztráció, megjelenik az üzenet, hogy most már be lehet jelentkezni.

## Shared

### MainLayout

Az oldal keretét megvalósító modul. Ebben található a fejléc és a lábléc. Tartozik hozzá egy .razor és egy .css file is.

**.razor:** A header tag-ben található egy téma beállítás, ami az alkalmazás színeit Htározza meg. Ezután egy appbar biztosítja az oldal fejlécét. Ezen található a home gomb és az oldal neve balra igazítva, majd egy keresősáv, kosár és profil gomb jobbra igazítva. Ezután található a body, ahová a különböző oldalak layout-jai kerülnek navigációtól függően. Végezetül található még a keret alján egy lábléc a különböző elérhetőségekkel. A kód legalján található a sikertelen keresés esetén megjelenő snackbar.

**.css:** Található benne formázás a body-ra, a láblécre és a helykitöltő panelre.

# BackEnd modulok

Az általam írt backend modulok felsorolása és rövid ismertetése. Az oldal hátterét C# kód adja, az ezen a nyelven írt osztályokat, változóikat és függvényeiket mutatom be.

## Pages

### Home.razor.cs – public partial class Home

**Változók:**

* **szoveg1: string** – A leírást tartalmazó szöveg első fele.
* **szoveg2: string** – A leírást tartalmazó szöveg második fele.

**Függvények:**

* **onCategory1(): void** – Az első kategória kiválasztása esetén a fagyik oldalára irányít.
* **onCategory2(): void** – A második kategória kiválasztása esetén a fagyik oldalára irányít.
* **onCategory3(): void** – A harmadik kategória kiválasztása esetén a fagyik oldalára irányít.

### IceCream.razor.cs – public partial class IceCream

**Változók:**

* **onlyInStock: bool** – Be van-e pipálva a checkbox, hogy csak a készleten lévő elemeket jelenítsük meg.
* **setMax: int** – A beállított maximum ár.
* **setMin: int** – A beállítottminimum ár.
* **minPrice: int** – A legkisebb elérhető ár.
* **maxPrice: int** – A legnagyobb elérhető ár.
* **checkboxes: bool[]** – A színekhez tartozó checkboxok közül melyik van bepipálva.

**Függvények:**

* **IceCream()**– A konstruktorban állítjuk be a minimum és maximum árakat.

### Login.razor.cs – public partial class Login

**Változók:**

* **username: string** – A beírt felhasználónév.
* **password: string** – A beírt jelszó.
* **snackBarIsOpen: bool** – Látható-e a snackbar.

**Függvények:**

* **loginOnClick(): void** – A bejelentkezés gomb callback függvénye. Helyes felhasználónév és jelszó esetén a profil oldalra irányít, egyéb esetben láthatóvá teszi a snackbar-t, hogy a beírt bejelentkezési adatok valamelyike nem helyes.

### Product.razor.cs – public partial class Product

**Változók:**

* **amount: int** – A megadott mennyiség a termékből.
* **allergens: Allergen[]**  – A táblázatban megjelenítendő allergének.

**Függvények:**

* **Product()** – Konstruktor. A mennyiséget 0-ra állítja.
* **toPay()**: **int -**  visszatér a termék ára és a kiválasztott mennyiség szerinti fizetendő összeggel.
* **onAddToCart()**: **void –** A kosárba helyezés callback függvénye. Ha a termék még nincs a kosárban, akkor belerakja mennyiséggel együtt. Ha benne van, akkor növeli a mennyiségét.

### SignUp.razor.cs – public partial class SignUp

**Változók:**

* **myModel: SignUpModel –** A regisztrációs mezők validálását segítő modell.
* **GenderItems: string[] –** A választható opciók a „nem” legördülő listában.

**Függvények:**

* **Reset(): void –** A reset gomb callback függvénye. Visszaállítja az összes beviteli mező értékét.

## Shared

### MainLayout.razor.cs – public partial class MainLayout

**Változók:**

* **searchString: string –** A keresési szöveg.
* **theme: MatTheme –** A MatBlazor téma, amit az oldalra alkalmazunk.

**Függvények:**

* **HomeOnClick(): void –** A home gomb callback függvénye, ami a home oldalra navigál.
* **profileOnClick(): void –** A profil ikon gomb callback függvénye, ami ha a felhasználó be van már jelentkezve, akkor a profiljára irányítja őt, ha nincs, akkor a bejelentkező felületre.
* **cartOnClick(): void –** A kosár ikon gomb callback függvénye, ami a kosárra navigálja a felhasználót.
* **searchOnClick(): void –** A keresés gomb callback függvénye, ami a beírt kulcsszó alapján a termék oldalára navigál vagy jelzi, hogy nem találja a terméket

## További modulok

### Allergen.cs – public class Allergen

Az allergéneket megvalósító osztály az allergén táblázat kitöltéséhez.

**Változók:**

* **name: string –** Az allergén neve, tartozik hozzá property.
* **contains: bool –** Tartalmazza-e a fagyi az adott allergént. Mivel egy konstans táblázatot jelenítek meg minden termék mellett terméktől függetlenül, így egyszerűbb volt itt elmenteni ennek a változónak az értékét. Valós működés mellett ez a megoldás nem működne. Tartozik hozzá property.

**Függvények:**

* **Allergen(name: string, contains: bool) –** Konstruktor, ami beállítja a paraméterként kapott értékre a név és a tartalmazás értékét.

### Card.cs – public class Card

A fagyik kártyáira kerülő információkat tároló objektum.

**Változók:**

* **name: string –** A fagyi neve, tartozik hozzá property.
* **price: int –** A fagyi ára, tartozik hozzá property.
* **description: string –** A fagyi leírása, tartozik hozzá property.
* **imageUrl: string –** A fagyi képének URL-je, tartozik hozzá property.
* **color: int –** A fagyi színének sorszáma, tartozik hozzá property.
* **inStock: bool –** A fagyi van-e éppen raktáron, tartozik hozzá property.

**Függvények:**

* **Card(name: string, price: int, description: string, imageUrl: string, color: int, inStock: bool) –** Konstruktor, ami beállítja a változók értékeit a paraméterben kapottakra.
* **onSelected(): void –** Ez a függvény fut le, amikor a felhasználó kiválaszt egy fagyit.

### CartElement.cs – public class CartElement

A kosárban lévő elemek.

**Változók:**

* **product: Card –** A kosárban lévő termék, tartozik hozzá property.
* **amount: int –** A kosárban lévő termék mennyisége, tartozik hozzá property.

**Függvények:**

* **CartElement(card: Card, amount: int) –** Konstruktor, beállítja a változók értékeit a paraméterként kapottakra.
* **remove():void –** Törli a kosárból az adott elemet.

### GenderDef.cs – public class GenderDef

A legördülő listából választható nemek.

**Változók:**

* **name: string –** A nem neve, tartozik hozzá property.

**Függvények:**

* **GenderDef(name: string) –** Konstruktor, beállítja a name változó értékét a paraméterként kapottra..

### Global.cs – public static class Global

Statikus osztály a globális változók tárolására.

**Változók:**

* **loggedIn: bool –** A felhasználó be van-e éppen jelentkezve.
* **cards: List<Card> -** A termékek listája.
* **names:** **string[] –** A fagyik neveinek listája.
* **prices:** **int[] –** A fagyik árainak listája.
* **descriptions:** **string[] –** A fagyik leírásainak listája.
* **urls:** **string[] –** A fagyik képeinek url-jeinek listája.
* **colors:** **int[] –** A fagyik színeinek sorszámának listája.
* **stocks:** **bool[] –** A fagyik közül melyek vannak készleten.
* **theChosenOne:** **Card –** Az aktuálisan kiválasztott termék, ha a felhasználó kiválaszt egyet, hogy megtekintse.
* **products:** **List<CartElement> -** A kosárban található elemek listája.
* **FullPrice: int –** A kosárelemekért összesen fizetendő összeg.

**Függvények:**

* **createCards(): void –** Létrehozza a fagyikhoz tartozó objektumokat.

# Felhasznált MatBlazor komponensek

Ebben a fejezetben ismertetem a MatBlazor külső komponens könyvtárból felhasznált elemeket. A .razor modulokon végigmenve mutatom be az azokban felhasznált komponenseket, a felhasználásuk módját, bennük rejlő lehetőségeket és nehézségeiket.

## Felugró értesítések

### SnackBar

A Snackbar általában az oldalon felugró rövid szöveges üzenet esetleg egy gombbal kiegészítve. Snackbar-t használtam a Login és a MainLayout felületen.

A felhasználáshoz először el kell helyezni egy MatSnackBar komponenst, amit egy abban elhelyezett MatSnackbarContent komponens segítségével tudunk feltölteni tartalommal.

<**MatSnackbar** @bind-IsOpen="változó\_neve">

<**MatSnackbarContent**>Üzenet</**MatSnackbarContent**>

</**MatSnackbar**>

A Snackbar lehelyezése után a C# kódban

## Form Control-ok

### Checkbox

### DatePicker

### NumericUpDownField

### Select

### Slider

### TextField

### Validation via EditContext

## Gombok

### Button

Gombokat használtam a Login, a Product és a SignUp oldalon.

### Icon

### IconButton

## Layout-ok

### Card

### Divider

### Layout Grid

### List

### Paper

### Themes

### Typography

## Navigáció

### AppBar

### ButtonLink

## Táblázatok

### Table

## Cart

MatList - MatListItem

MatPaper

MatNumericUpDownField

MatH5

## IceCream

Grid Layout classes - mat-layout-grid, mat-layout-grid-inner, mat-layout-grid-cell, mat-layout-grid-cell-span-„n”

MatH1

MatH5

MatDivider

MatCheckbox

MatBody1

MatBody2

MatSlider

MatCard – MatCardContent, MatCardMedia, MatHeadline6, MatSubtitle2, MatCardActions, MatCardActionIcons

MatButtonLink

## Login

MatH1

MatTextField

MatButton

MatBody2

MatButtonLink

MatSnackBar - MatSnackBarContent

## Product

MatH2

MatDivider

MatBody1

MatNumericUpDownField

MatTable – MatTableHeader, MatMatTableRow

MatButton

## Profile

MatH1

MatH4

MatList – MatListItem, MatListItemPrimaryText, MatListItemSecondaryText, MatListDivider

## SignUp

Validation via EditContext - ValidationMessage

MatTextField

MatH1

MatDatePicker

MatSelect – MatOptionString

MatButton

## MainLayout

MatThemeProvider

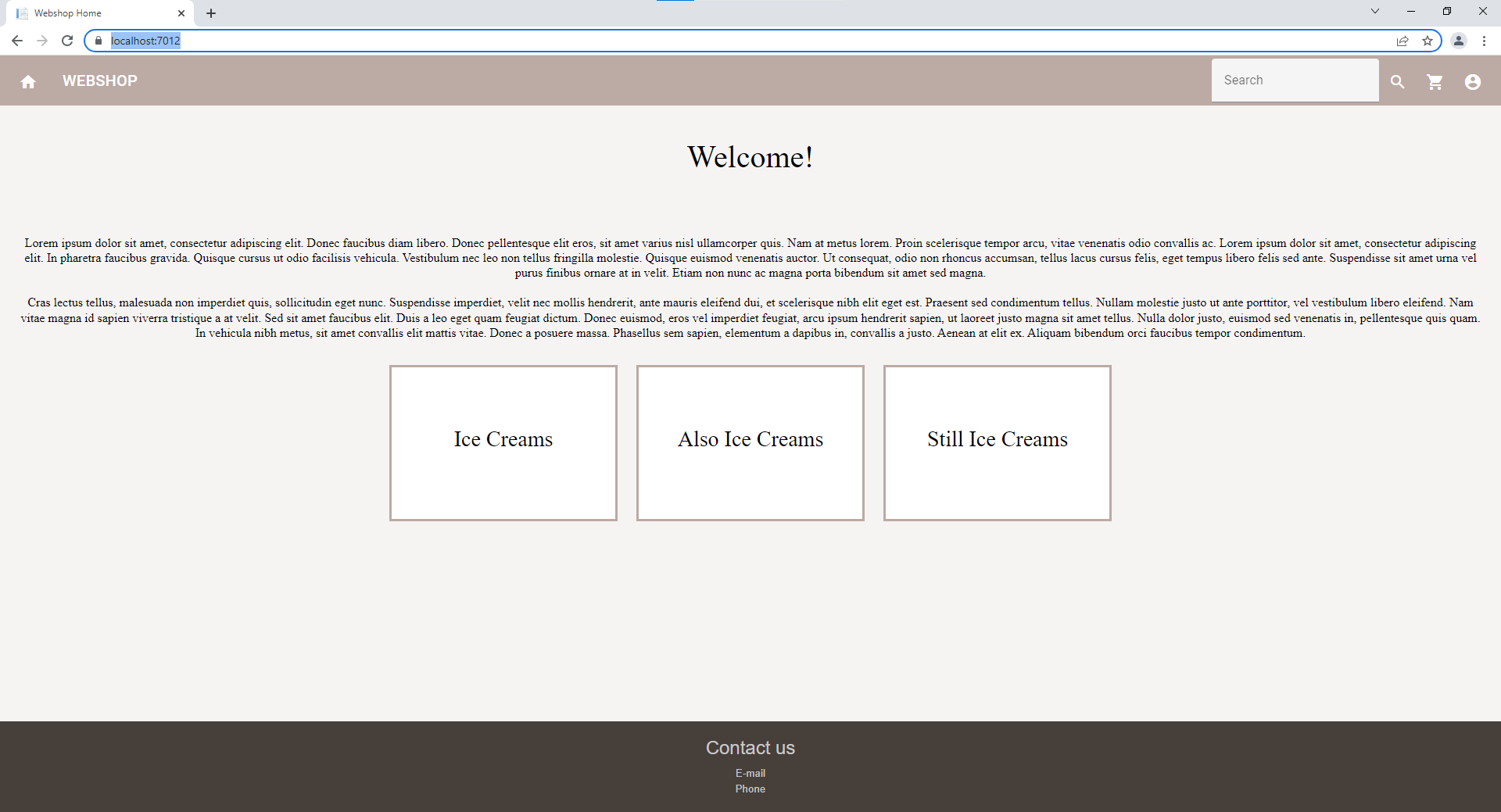
MatAppbarContainer – MatAppBar, MatAppbarRow, MatAppBarSection, MatAppbarTitle, MatAppBarContent

MatIconButton

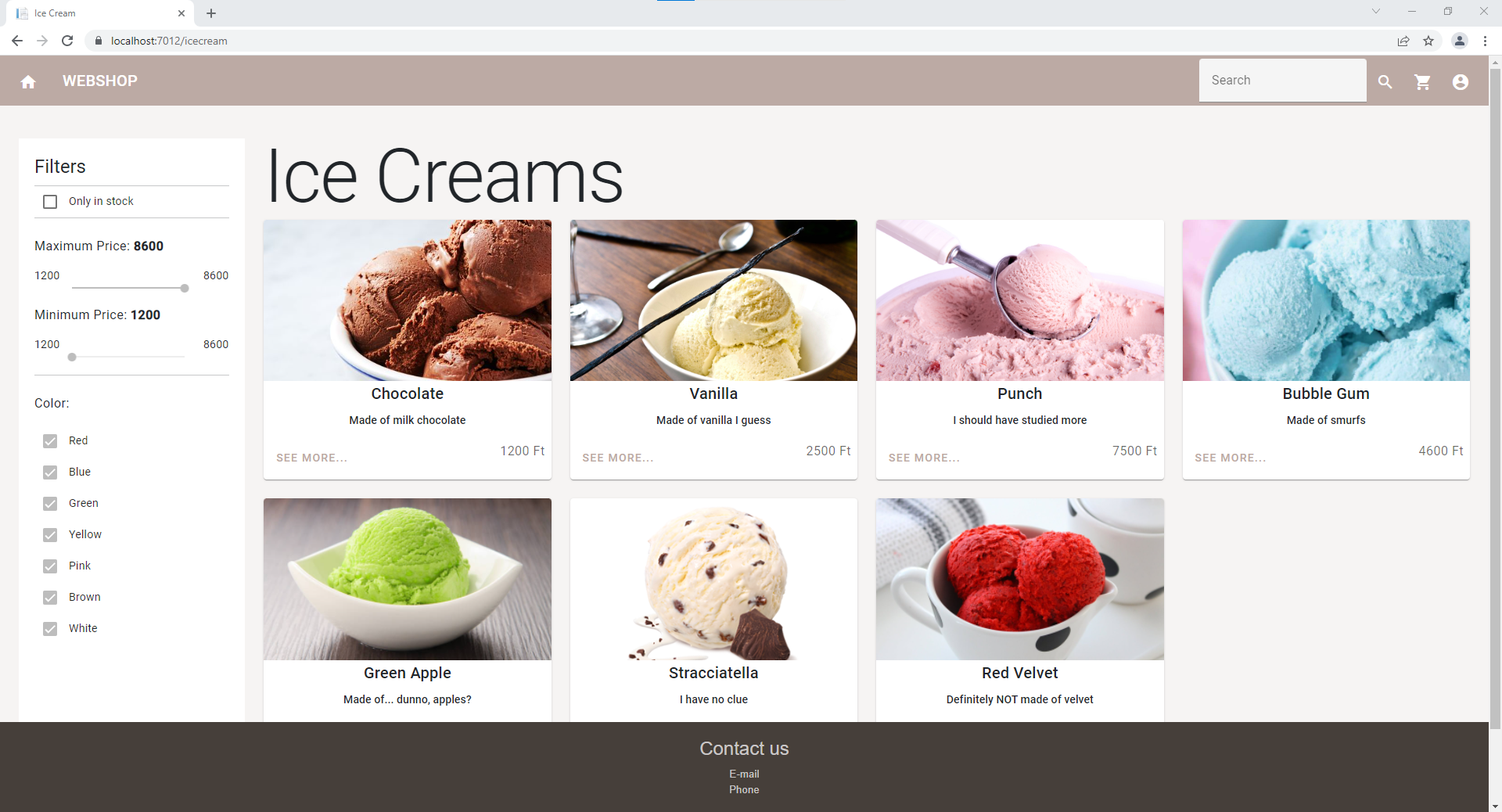
MatTextField

MatSnackBar - MatSnackbarContent

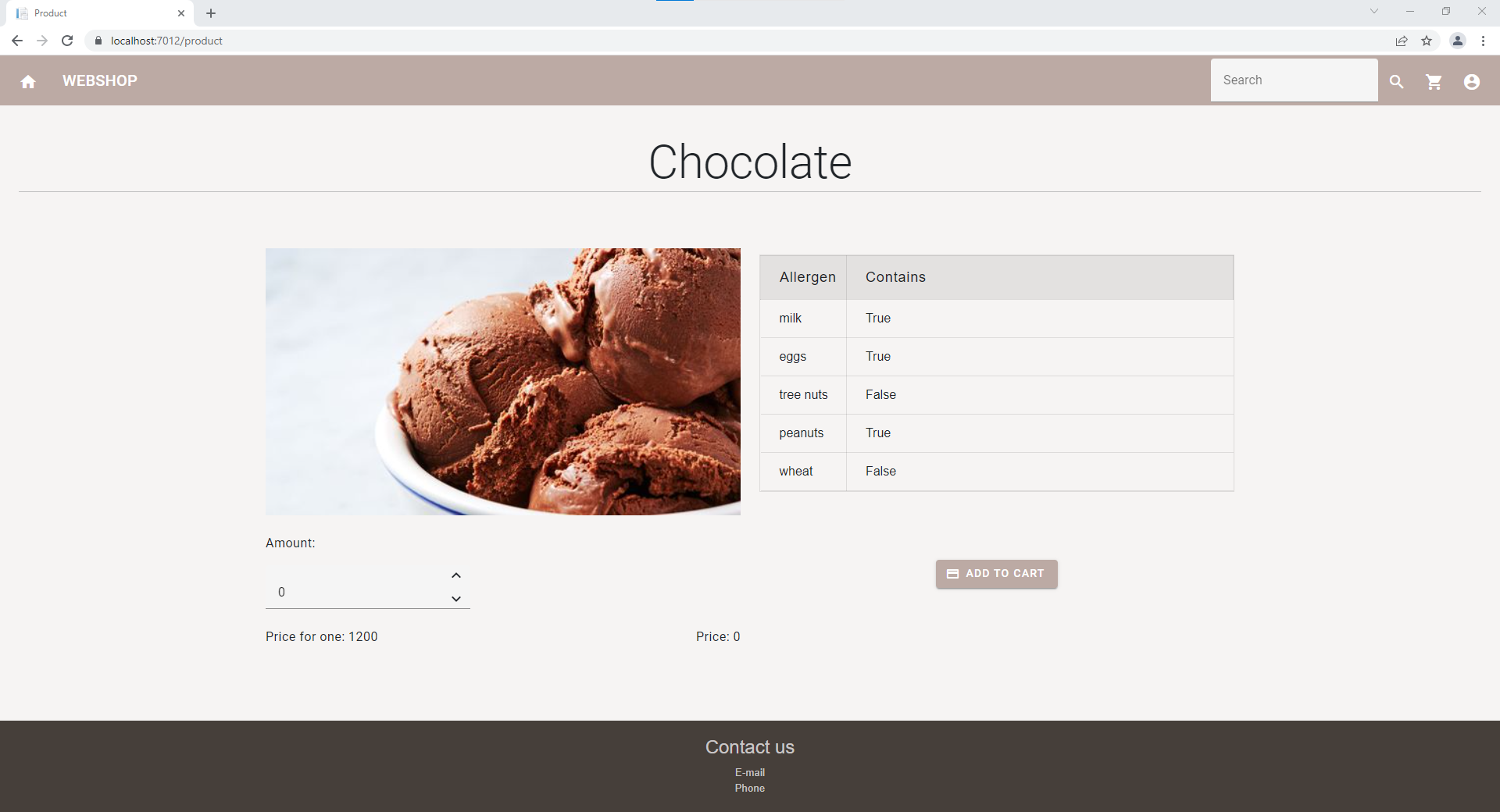
# Az elkészült webalkalmazás - képgaléria



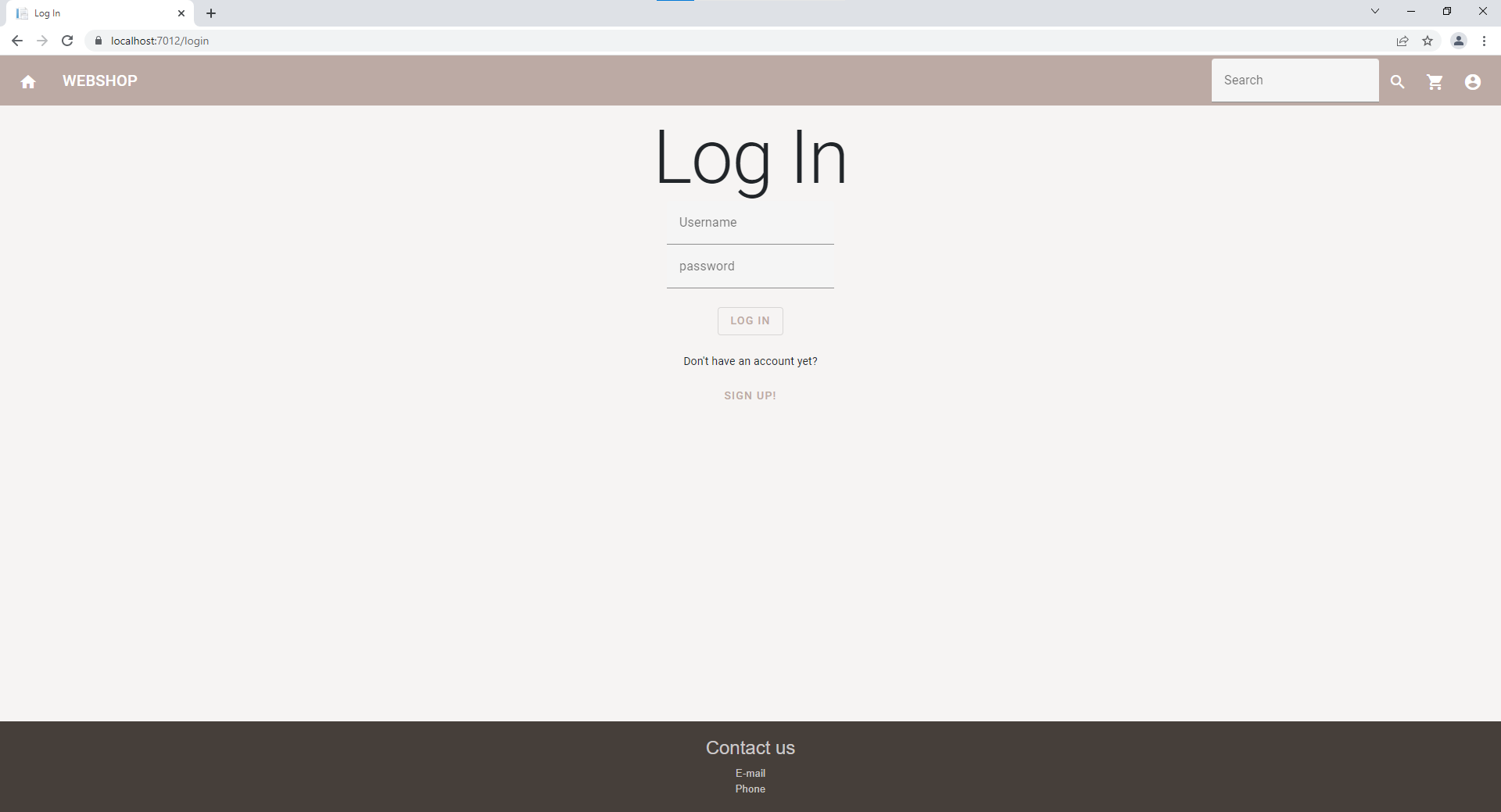
. ábra: Home oldal



2. ábra: Fagyik oldala



3. ábra: Termékek oldala

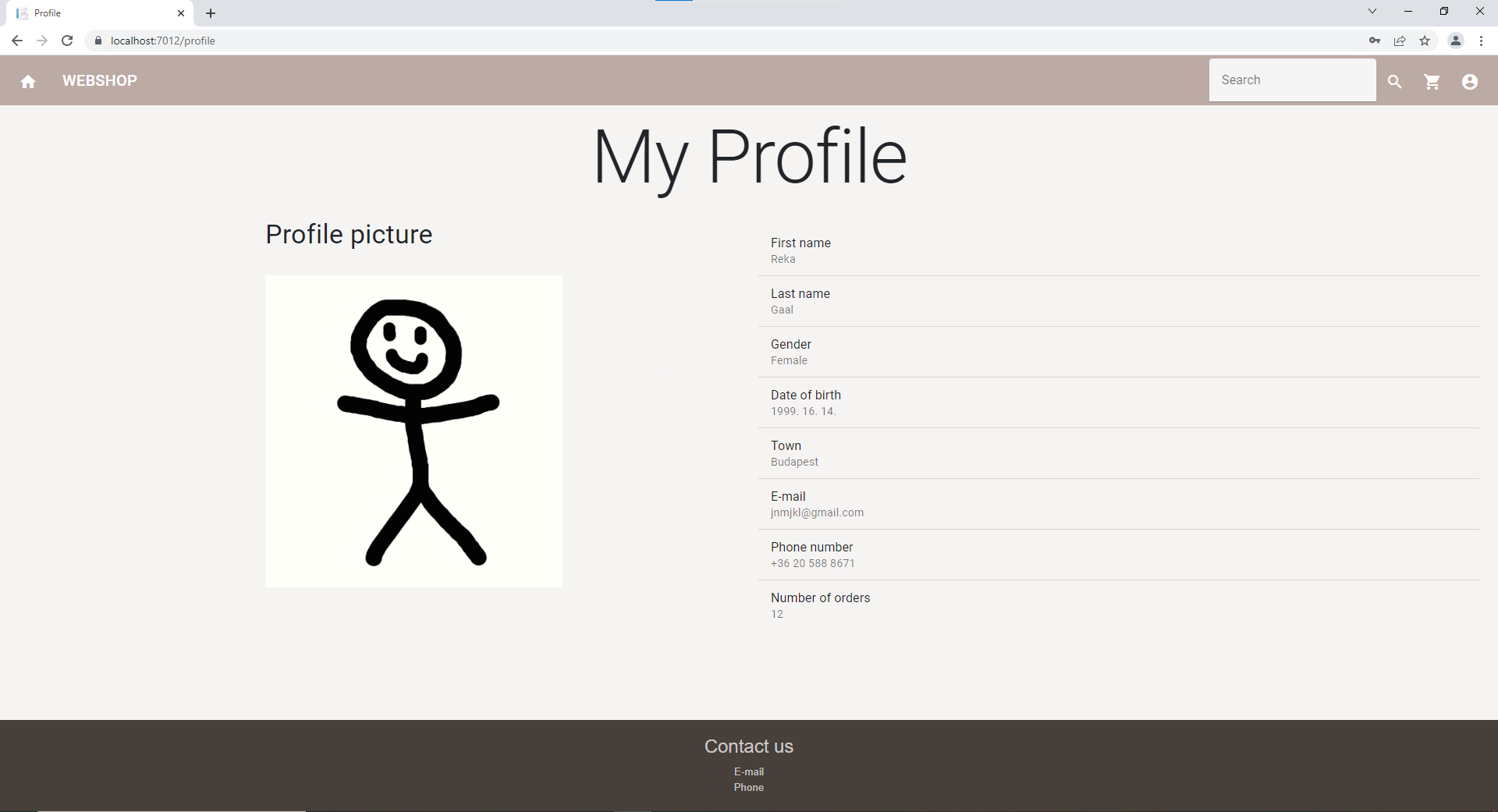


4. ábra: Bejelentkezés

A képen szöveg látható

Automatikusan generált leírás

5. ábra: Regisztráció



6. ábra: Profil oldal

A képen szöveg látható

Automatikusan generált leírás

7. ábra: Kosár