A black background with grey leaves

AI-generated content may be incorrect.

Vizsgaremek

PapírPalota webshop

A grey logo on a black background

AI-generated content may be incorrect.

2025. április 29.

Galambos Bence

Pál Barna

Varga Bálint

**Vizsgaremek adatlap**

A Vizsgaremek készítői:

Neve: Galambos Bence

E-mail címe: [galambos.bence@blathy.info](mailto:galambos.bence@blathy.info)

Neve: Pál Barna

E-mail címe: [pal.barna@blathy.info](mailto:pal.barna@blathy.info)

Neve: Varga Bálint

E-mail címe: [varga.balint@blathy.info](mailto:varga.balint@blathy.info)

A Vizsgaremek témája:

Többrétegű, REST API alapú webshop és adminisztrációs rendszer papír és irodaszerek értékesítésére

A Vizsgaremek címe:

PapírPalota – Többrétegű kereskedelmi platform Laravel + React + WPF

Kelt: Budapest, 2024. április 29.

........................................ ........................................

Galambos Bence Pál Barna

........................................

Varga Bálint

A Vizsgaremek készítőinek aláírása

**Eredetiség nyilatkozat**

Alulírott tanuló kijelentem, hogy a vizsgaremek saját és csapattársa(i)m munkájának

eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült

vizsgaremekrészét képező anyagokat az intézmény archiválhatja és felhasználhatja.

Kelt: Budapest, 2024. április.29.

............................................................

Galambos Bence

Tanuló aláírása

............................................................

Pál Barna

Tanuló aláírása

............................................................

Varga Bálint

Tanuló aláírása

Tartalom

[**1. Projekt ismertetése** 4](#_Toc196856786)

[**2. Felhasználói dokumentáció** 4](#_Toc196856787)

[**2.a Felületek ismertetése** 4](#_Toc196856788)

[**2.b Funkciók ismertetése** 8](#_Toc196856789)

[**2.c Használati instrukciók** 8](#_Toc196856790)

[**3. Fejlesztői dokumentáció** 9](#_Toc196856791)

[**3.a Munkamegosztás** 9](#_Toc196856792)

[**3.b Telepítési dokumentáció** 9](#_Toc196856793)

[**3.c Adatbázis dokumentáció** 10](#_Toc196856794)

[**3.d Komponens/struktúra dokumentáció** 11](#_Toc196856795)

[**3.e API dokumentáció** 12](#_Toc196856796)

[**3.f Teszt dokumentáció** 13](#_Toc196856797)

[**4. Továbbfejlesztési lehetőségek** 13](#_Toc196856798)

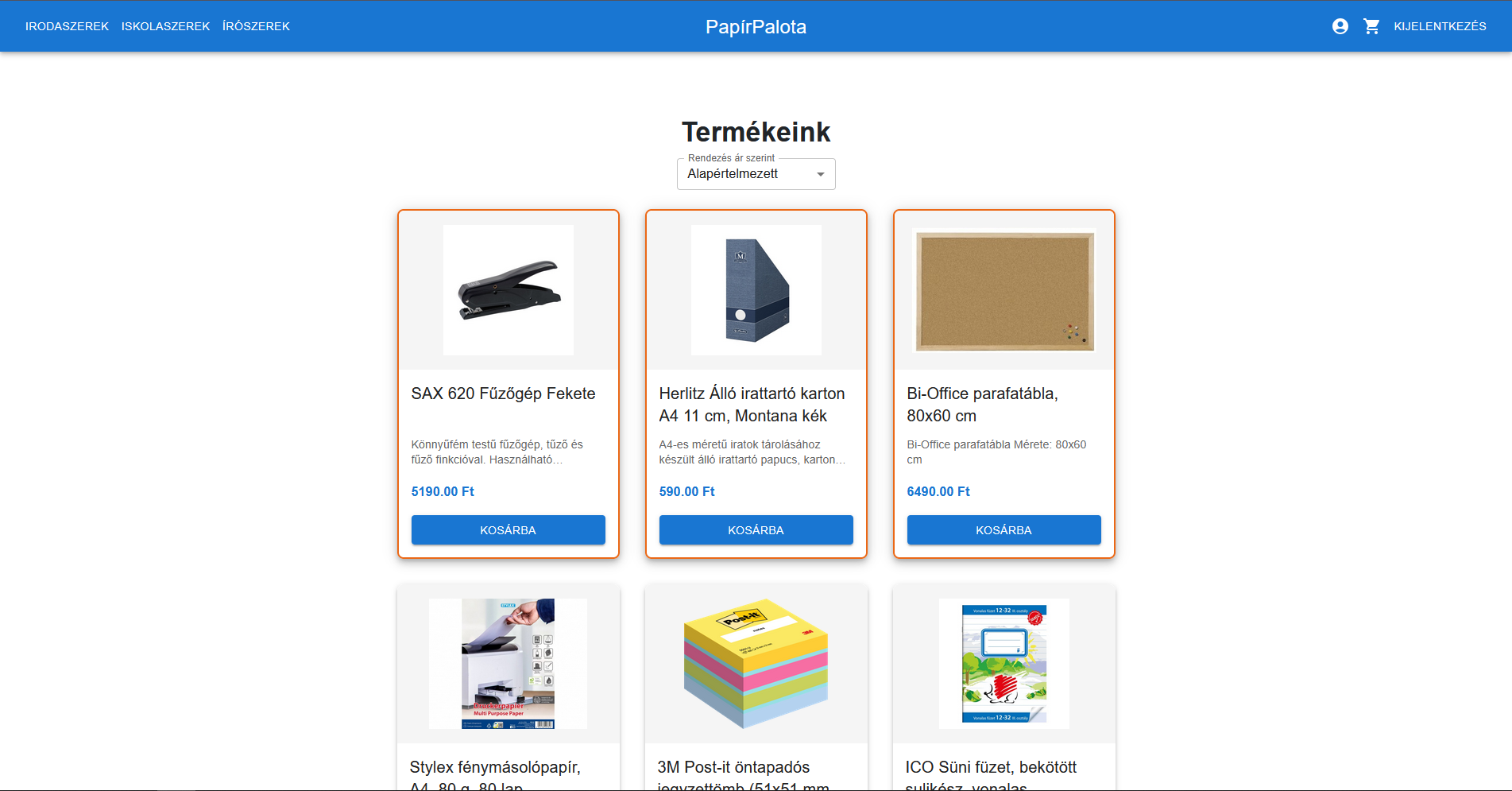
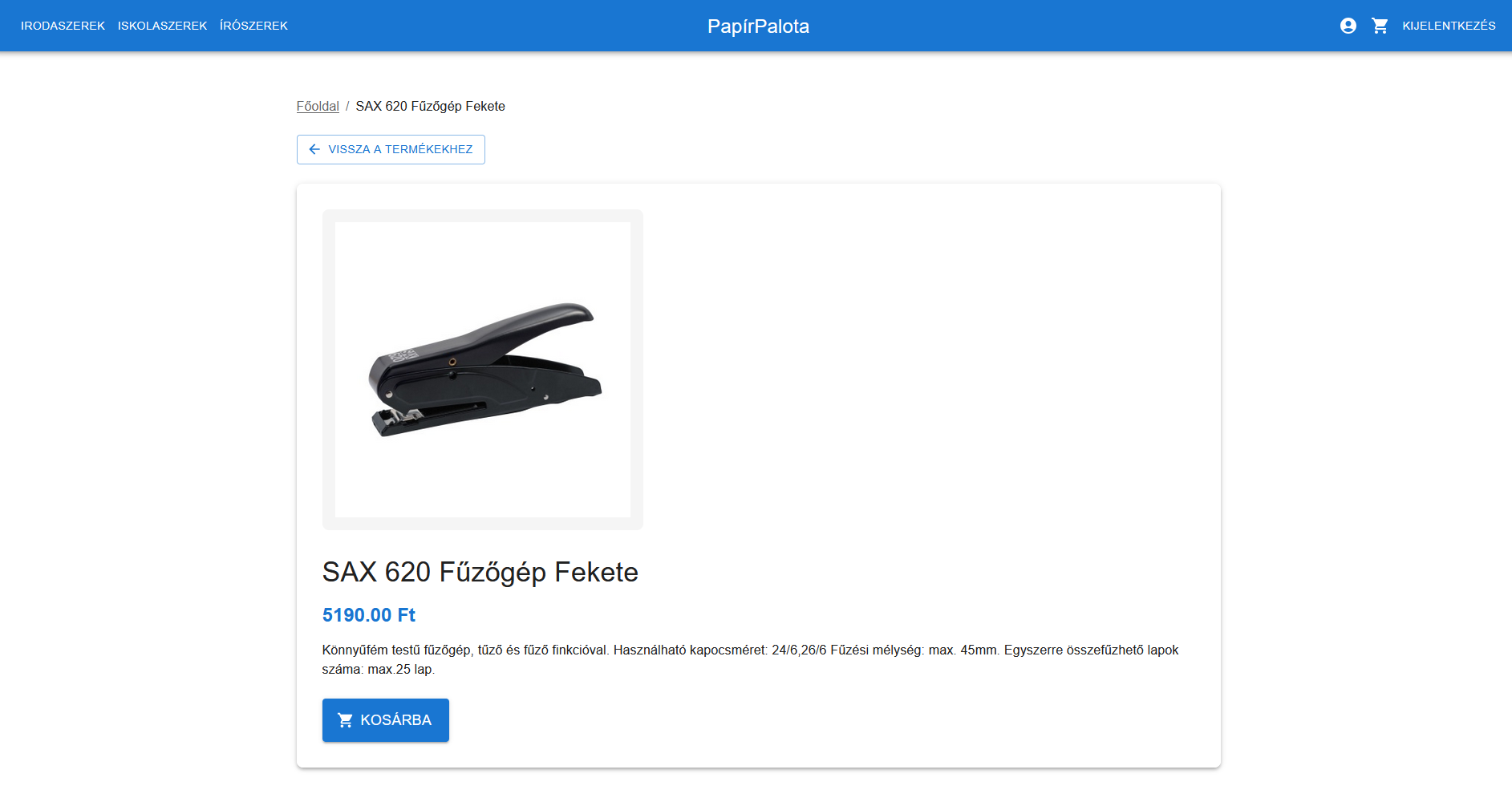
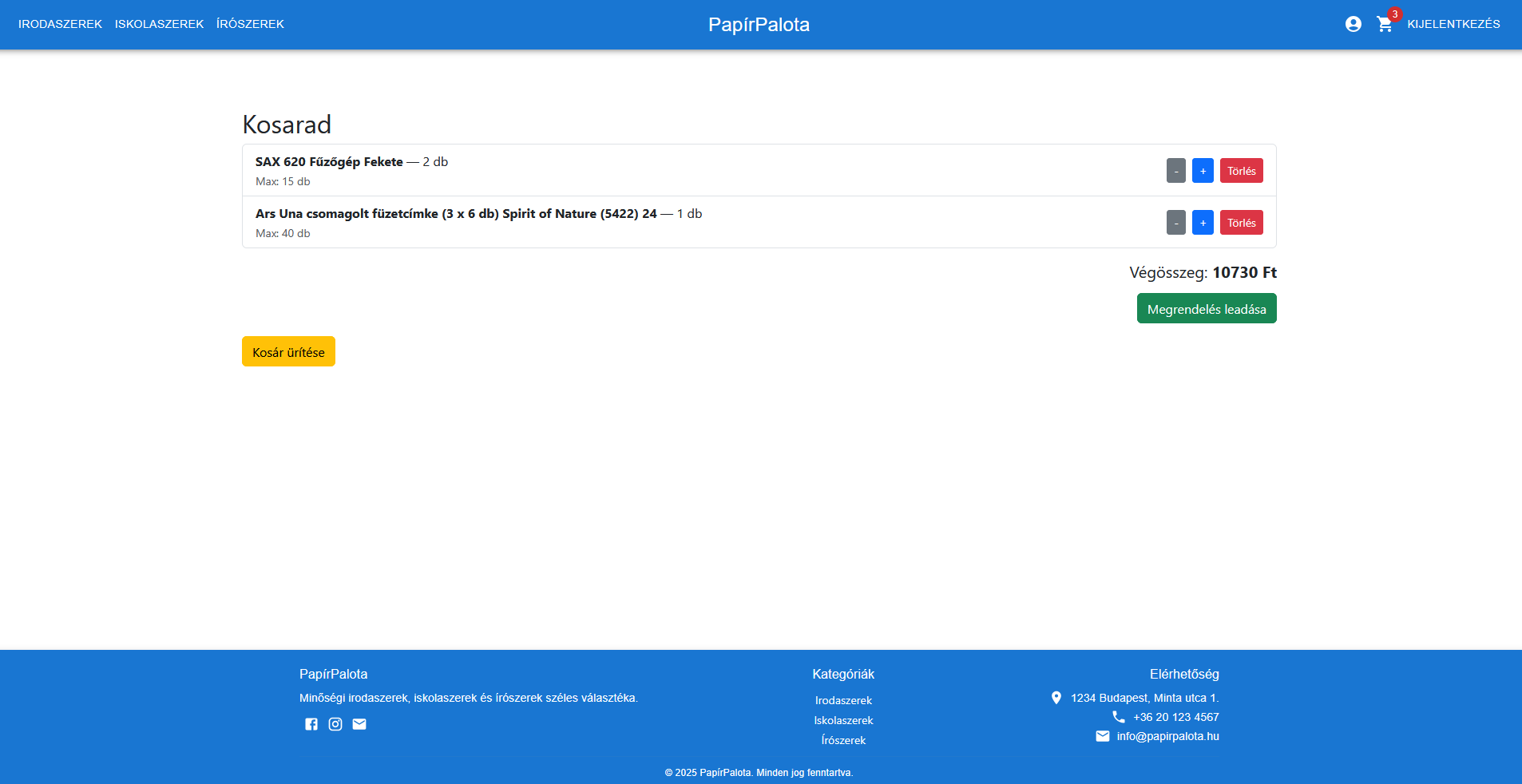
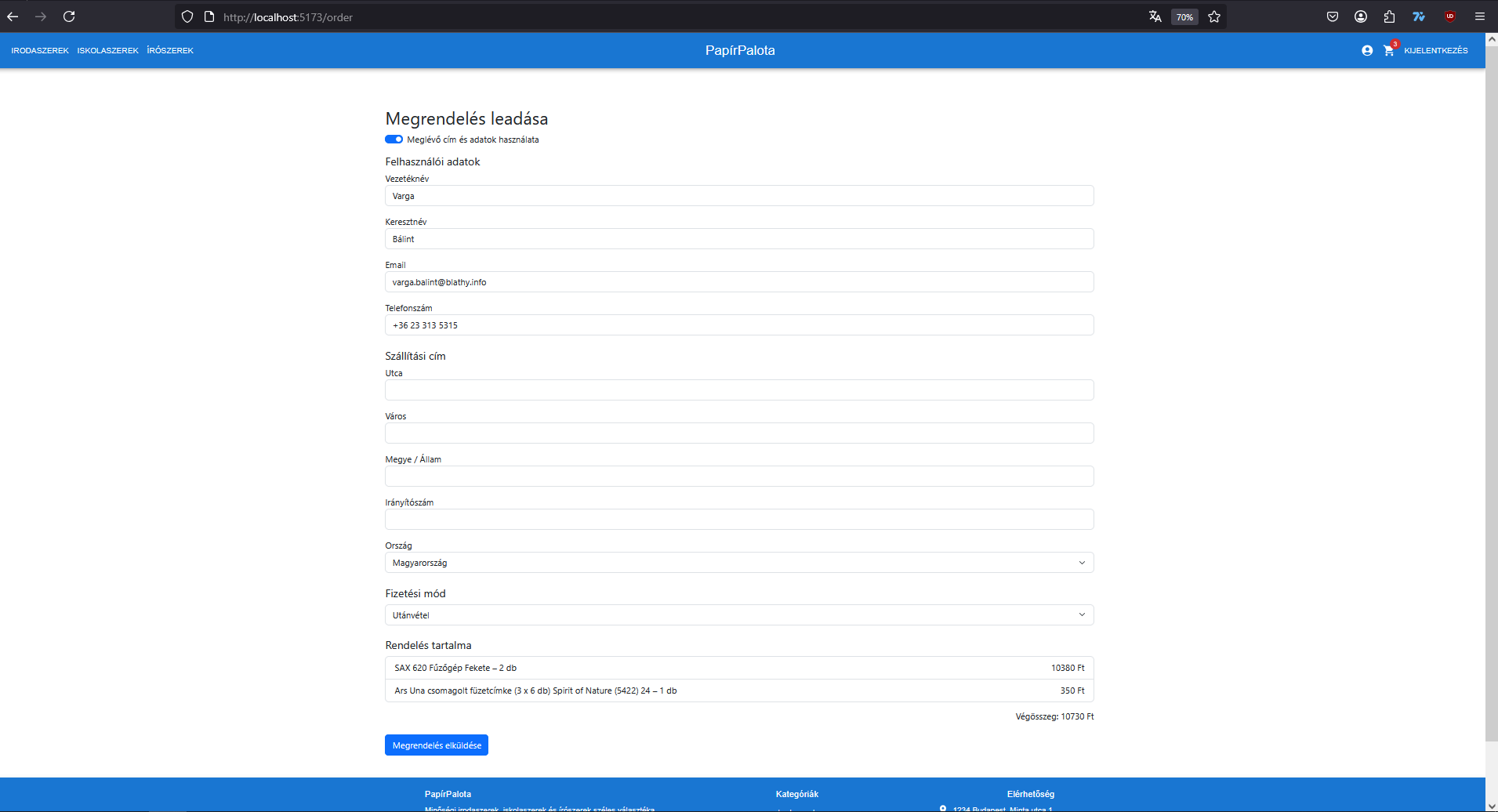
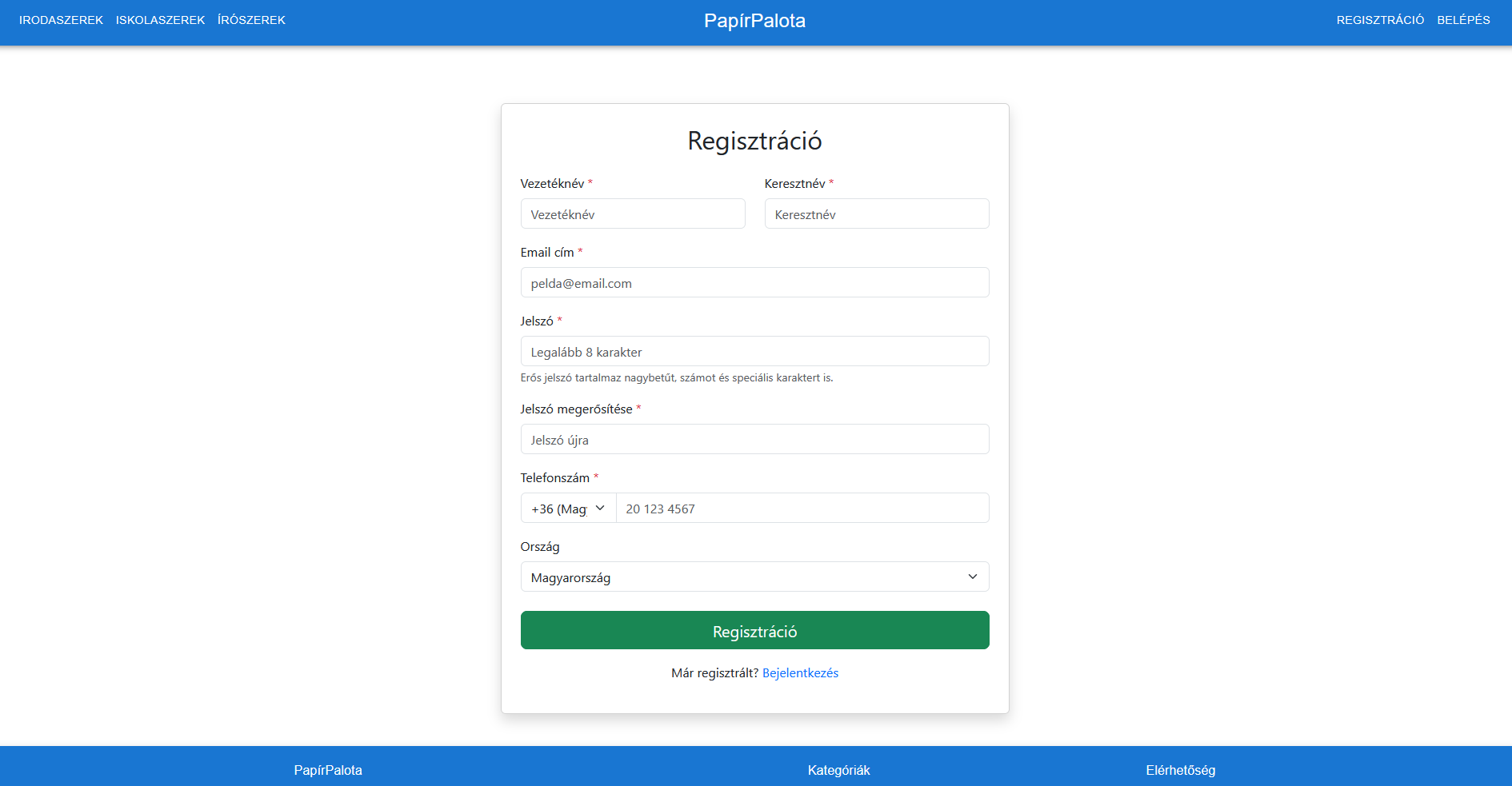
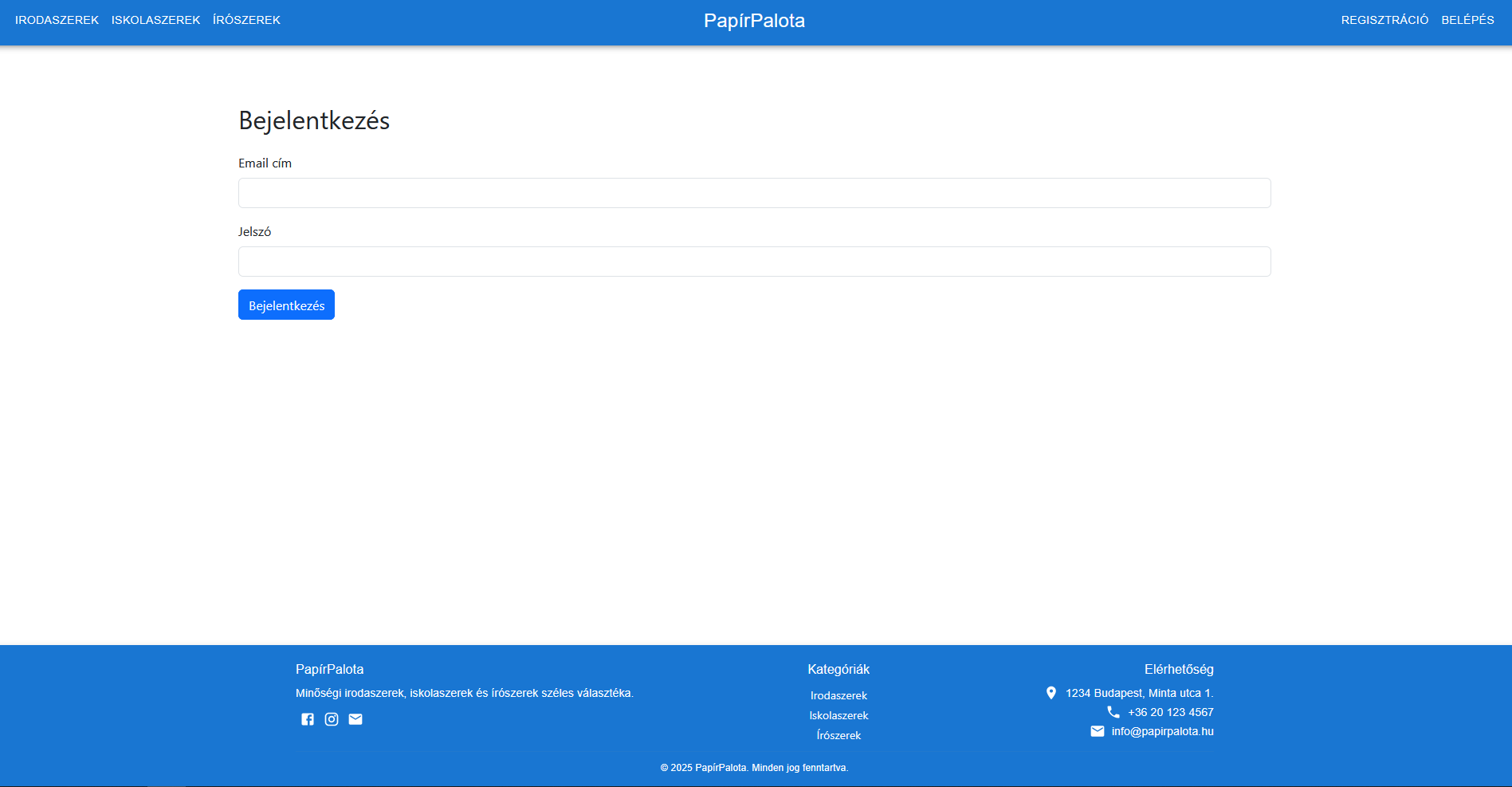
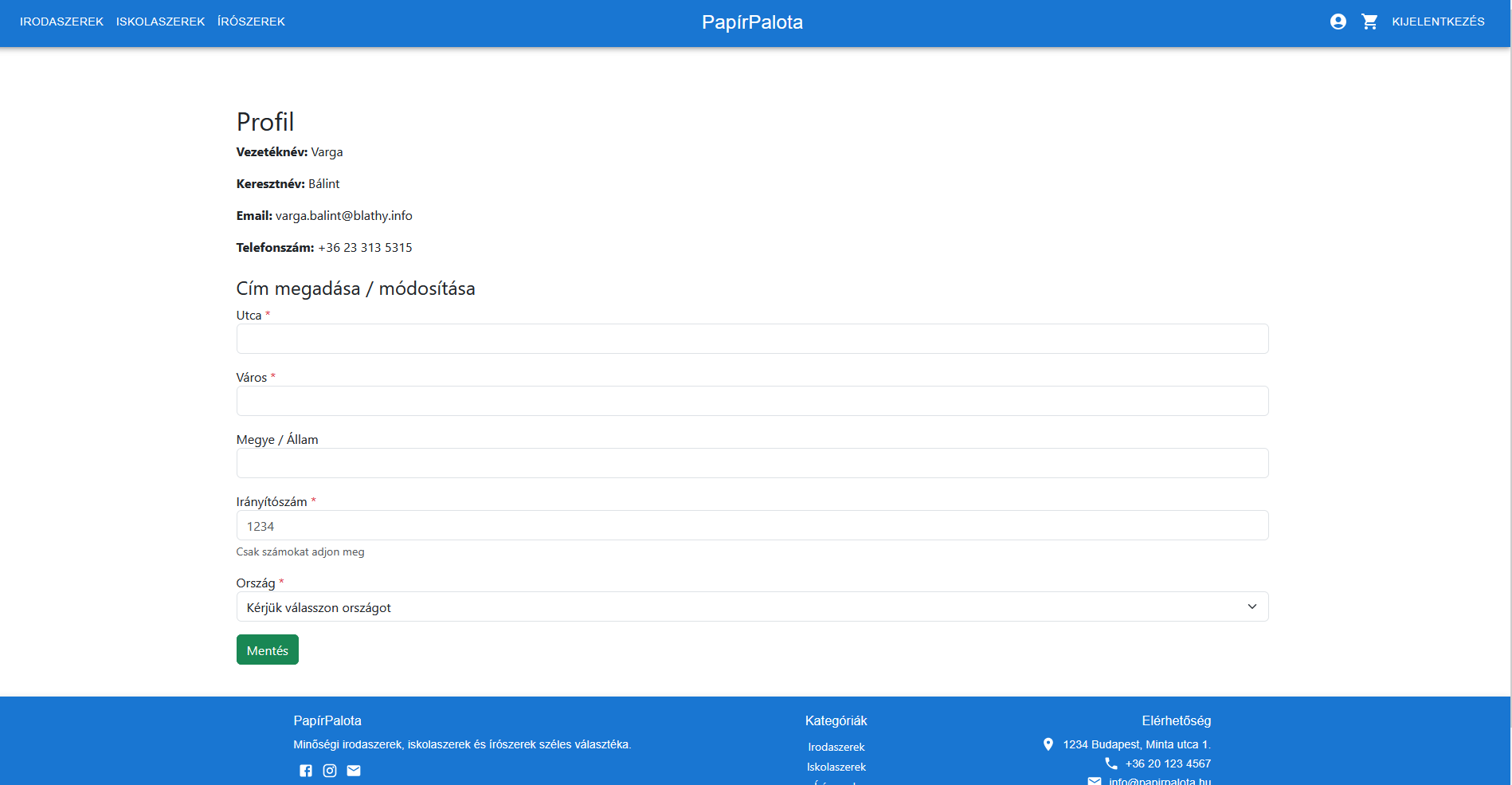
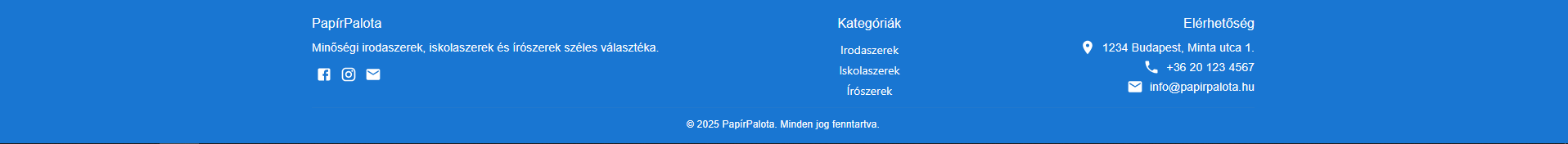
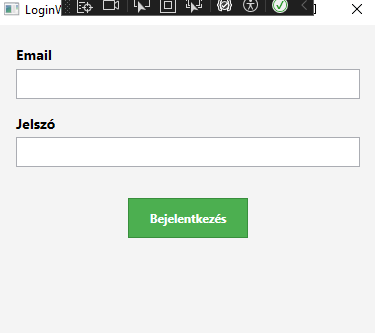
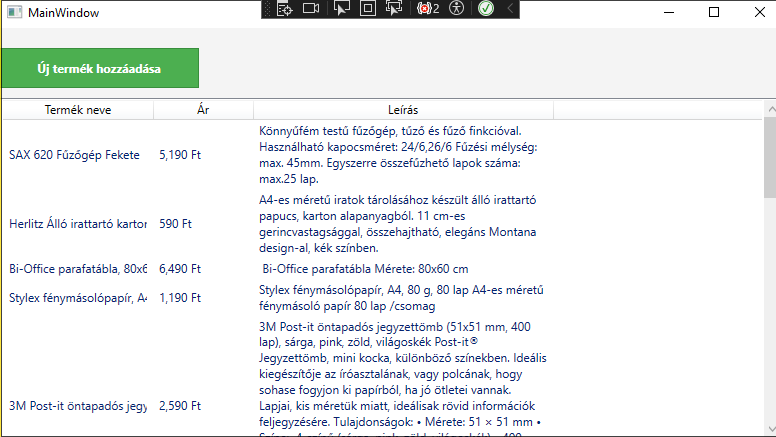
[**5. Összegzés** 13](#_Toc196856799)

**1. Projekt ismertetése**

A PapírPalota („OfficeWebshop”) egy teljes körű, többrétegű webes és asztali kereskedelmi rendszer. A backendet Laravel (PHP) keretrendszerrel valósítottuk meg, a frontend webes rész React + Vite és Material-UI alapokon fut, az adminisztrációs asztali klienst pedig WPF (C#) alatt építettük MVVM mintával. A rendszer fő célja, hogy az adminisztrátorok könnyen listázhassák, létrehozhassák, módosíthassák és törölhessék a termékeket, miközben a felhasználói felület gyors, aszinkron adatlekéréseken alapul. A rendszer lehetővé teszi a termékek kezelését, kosárkezelést, rendelés leadását, felhasználói profil és címkezelést.

**2. Felhasználói dokumentáció**

**2.a Felületek ismertetése**

* **Webes felület (React App):**
* **Home**: terméklista (GET /api/products), kategória szűrés (?category\_id=), ár szerinti rendezés 
* **ProductDetail**: termék részletek (GET /api/products/{id}), kosárba helyezés
* **CartPage**: kosár listázás (GET /api/cart), mennyiség módosítás (PUT), törlés (DELETE), kosár ürítés (POST /api/cart/clear), végösszeg
* **OrderPage**: rendelés leadása (POST /api/order), meglévő vagy új cím és adatok használata, fizetési módok (utánvétel, bankkártya) 
* **RegisterForm / LoginForm**: regisztráció (POST /api/register), bejelentkezés (POST /api/login) 
* 
* **ProfilePage**: profiladatok (GET /api/profile), címkezelés (GET/POST /api/profile/address) 
* **Layout**: navigáció, kategória linkek, kosár és profil menü, footer információk
* 
* **Asztali felület (WPF):**
  + **LoginWindow** – modal bejelentkezés e-mail/jelszó párossal
  + **MainWindow** – terméklista, vezérlőgombokkal CRUD-műveletek indításához
  + **ProductAddWindow / ProductEditWindow** – külön ablakban új termék hozzáadása vagy szerkesztéseA screenshot of a computer

    AI-generated content may be incorrect.A screenshot of a computer

    AI-generated content may be incorrect.

**2.b Funkciók ismertetése**

* **Autentikáció**: Laravel Sanctum token alapú (register, login, védett végpontok auth:sanctum middleware-rel)
* **Termékek CRUD**: teljes termékkezelés a ProductController által (weben csak listázás + részletezés + kosárba helyezés)
* **Kategóriák listázása**: GET /api/categories
* **Felhasználók listázása**: GET /api/users (admin)
* **Kosár műveletek**: CartController – tétel hozzáadás, frissítés, törlés, ürítés
* **Rendelés**: OrderController – kosár tételek rendelésként mentése, készletellenőrzés, rendelési tétel és fizetés mentése
* **Fizetés lekérdezés**: PaymentController – GET /api/order/{order}/payment
* **Címkezelés**: AddressController – profil cím lekérése és mentése

**2.c Használati instrukciók**

1. **Webes kliens indítása**
   * **Backend futtatása:**
   * **cd webshop-backend**
   * **composer install**
   * **cp .env.example .env && php artisan key:generate**
   * **php artisan migrate:fresh --seed**
   * **php artisan serve**
   * **Frontend futtatása**
   * cd webshop-frontend
   * npm install
   * npm run dev
2. **Asztali alkalmazás indítása**
   * Visual Studio-ban nyisd meg a WPF projektet
   * Futtatás előtt ellenőrizd, hogy a AuthService.Token üresen indul
   * Indítsd el a megadott OnStartup logika szerint
   * **vagy**
   * Nyissa meg az OfficeWebshopAdminPanelApp mappán belül az OfficeWebshopAdminPanel.exe-t

**3. Fejlesztői dokumentáció**

**3.a Munkamegosztás**

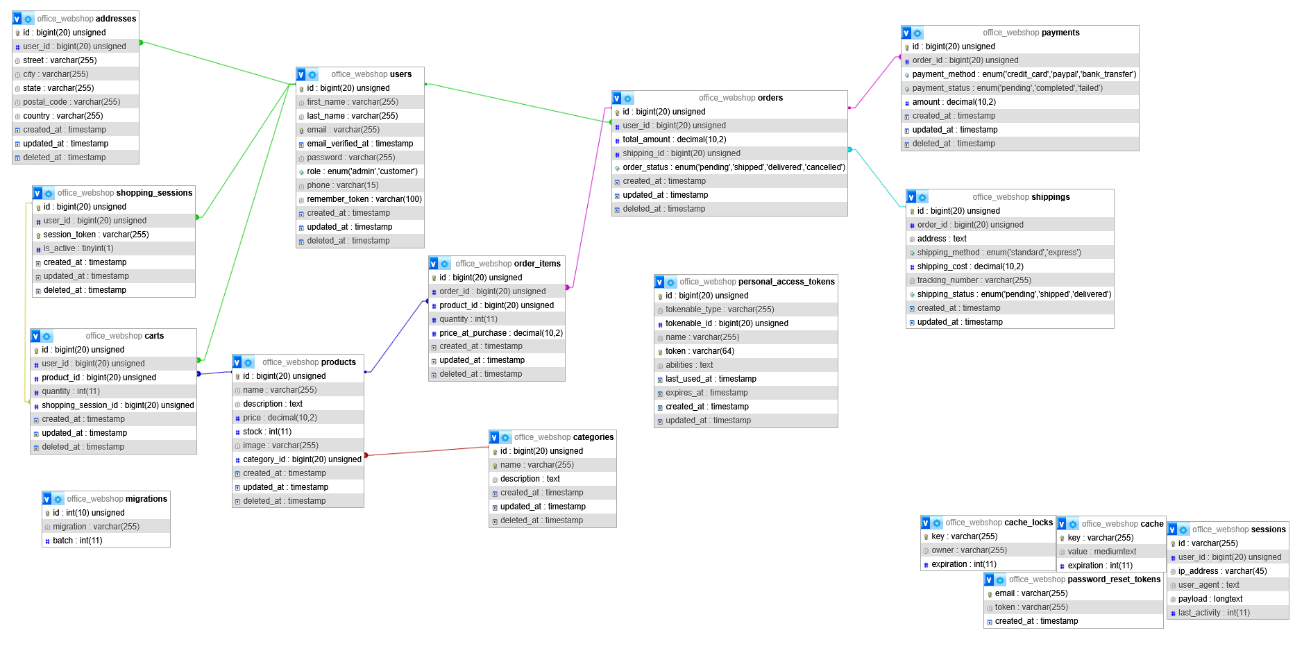
Backend: Galambos Bence, Pál Barna, Varga Bálint  
Frontend: Galambos Bence, Varga Bálint  
WPF: Pál Barna, Varga Bálint

**3.b Telepítési dokumentáció**

* **Backend (Laravel):**
  1. composer create-project --prefer-dist laravel/laravel webshop-backend
  2. .env fájl beállítása (DB\_CONNECTION, DB\_HOST, DB\_PORT, DB\_DATABASE, DB\_USERNAME, DB\_PASSWORD)
  3. php artisan migrate:fresh --seed
  4. php artisan serve
* **Frontend (React):**
  1. npm create vite@latest webshop-frontend --template react
  2. npm install
  3. npm run dev
* **Asztali alkalmazás (WPF):**
  1. Megnyitás VS 2022-ben, .NET 6+
  2. Futtatás OnStartup logika szerint

**3.c Adatbázis dokumentáció**

**i. Diagram**

****

**ii. Mezők részletes ismertetése**

* **users**: first\_name, last\_name, email|unique, password, role{admin,customer}, phone?, softDeletes
* **categories**: name|unique, description?, softDeletes
* **products**: name, description?, price (decimal), stock (int), image, category\_id FK cascade, softDeletes
* **orders**: user\_id, total\_amount (decimal), shipping\_id?, order\_status{pending,shipped,delivered,cancelled}, softDeletes
* **order\_items**: order\_id, product\_id, quantity, price\_at\_purchase, softDeletes
* **payments**: order\_id, payment\_method{credit\_card,paypal,bank\_transfer}, payment\_status{pending,completed,failed}, amount, softDeletes
* **shippings**: order\_id?, address (text), shipping\_method{standard,express}, shipping\_cost, tracking\_number?, shipping\_status, timestamps
* **addresses**: user\_id, street, city, state?, postal\_code, country, softDeletes
* **carts**: user\_id, product\_id, quantity, shopping\_session\_id?, softDeletes
* **personal\_access\_tokens**: morphs tokenable, name, token|unique, abilities?, expires\_at?

**iii. Kapcsolatok indoklása**

* User – Address: 1:1
* Category – Product: 1:N
* Product – Cart: 1:N
* User – Order: 1:N
* Order – OrderItem: 1:N; Order – Payment: 1:1; Order – Shipping: 1:1

**3.d Komponens/struktúra dokumentáció**

**i. Projekt elemei**

* **Backend**: Models (User, Category, Product, Order, OrderItem, Payment, Shipping, Address, Cart), Controllers (Auth, Product, Category, Order, Cart, Payment, Address), Requests, Seeders, Migrations
* **Frontend**: Components (Home, ProductDetail, CartPage, OrderPage, RegisterForm, LoginForm, ProfilePage, Layout)
* **WPF**: Modellek (ProductModel), ViewModel-ek (BaseViewModel, LoginViewModel, ProductAddViewModel, ProductEditViewModel, ProductViewModel), Views(Nézetek) (LoginWindow, ProductAddWindow, ProductEditWindow), AuthService és HTTP kliens

1. **Osztályok / fájlok**
   * Backend: Models, Controllers, Requests, Seeders, Migrations
   * Frontend: React komponensek (App.jsx), Axios konfiguráció, state-hookok
   * WPF: Modellek, ViewModel-ek, Nézetek (XAML)
2. **Kapcsolatok**
   * React ↔ Laravel API (Axios)
   * WPF ↔ Laravel API (HttpClient + token)

**ii. Főbb elemek részletes ismertetése**

* **Controller / BusinessLogic (ProductController.php):**
  + index() – összes termék JSON listázása
  + store() – új termék mentése validált kérelemmel
  + update() – meglévő termék frissítése
  + destroy() – termék törlése (soft delete)
* **AuthController**: register(), login() – jelszó erősség, validáció
* **CartController**: kosár CRUD, mennyiség update, clear
* **OrderController**: rendelés létrehozás, készletellenőrzés, tétel- és fizetésmentés
* **AddressController**: profilcím lekérés/mentés updateOrCreate
* **Service Layer (AuthService.cs – WPF):**
  + Statikus osztály a bejelentkezéskor kapott token tárolására és visszatöltésére

**3.e API dokumentáció**

**i. Végpontok**

1. **Útvonalak**

|  |  |  |
| --- | --- | --- |
| Módszer | Útvonal | Leírás |
| GET | /api/user | Authentikált felhasználó |
| GET | /api/users | Felhasználók listázása (admin) |
| GET | /api/categories | Kategóriák listázása |
| Resource | /api/products | Termékek CRUD |
| POST | /api/register | Regisztráció |
| POST | /api/login | Bejelentkezés |
| GET | /api/profile | Profiladat |
| GET | /api/profile/address | Profilcím lekérdezése |
| POST | /api/profile/address | Profilcím mentése/updateOrCreate |
| GET | /api/cart | Kosár tartalma |
| POST | /api/cart | Kosárba helyezés |
| PUT | /api/cart/{cart} | Kosár tétel mennyiség frissítése |
| DELETE | /api/cart/{cart} | Kosár tétel törlése |
| POST | /api/cart/clear | Kosár ürítése |
| POST | /api/order | Rendelés leadása |
| GET | /api/orders | Felhasználói rendelések listázása |
| GET | /api/order/{order}/payment | Fizetés lekérdezése rendeléshez |

1. **Paraméteres kérések lehetőségei**

* **register**: first\_name, last\_name, email|email:dns|unique, password|min8, A-Z,a-z,0-9, phone?
* **login**: email, password
* **Store/UpdateProductRequest**: name|string, description|string?, price|numeric, stock|integer, image|file|string, category\_id|exists:categories
* **Cart store/update**: product\_id|exists:products, quantity|integer|min1|max=stock
* **Order store**: user adatok, címadatok, payment\_method|in:cod,credit\_card
* **Address store**: cím mezők required, state?

1. **Válasz**

* JSON objektum adattal és/vagy msg mezővel

**3.f Teszt dokumentáció**

Laravel Feature tesztek különböző féle cselekményekre: AuthTest, CartTest, ProductTest, OrderTest

A screenshot of a computer

AI-generated content may be incorrect.

**4. Továbbfejlesztési lehetőségek**

* **Lapozás pagináció a webes és WPF listákban** – API válasz beburkolása links.next\_page\_url-lal, ciklikus oldalletöltés
* **Kapcsolatok bővítése** – márkák, rendeléskezelés, fizetés modul
* Felhasználói rendelés előzmények, értesítések, wishlist

**5. Összegzés**

A PapírPalota(webshop) (OfficeWebshop) projekt egy moduláris felépítésű, könnyen karbantartható kereskedelmi rendszer. A Laravel backend és React frontend architektúra jól szétválasztja az üzleti logikát és a felhasználói felületet, míg az asztali WPF kliens kiegészíti az adminisztrációs igényeket. A jövőben tervezett fejlesztések tovább erősítik a rendszer funkcionalitását és használhatóságát, miközben a dokumentáció mind felhasználói, mind fejlesztői szempontból átláthatóvá teszi a kód bővítését és üzemeltetését.