

Záródolgozat

Meggyesi József Márk

Pupa Coffee Roasting Co.

2023

TARTALOM

| | |
|---|-----------|
| BEVEZETÉS | 3 |
| TÉMAVÁLASZTÁS INDOKLÁSA..... | 3 |
| FEJLESZTŐI DOKUMENTÁCIÓ | 4 |
| FEJLESZTŐI KÖRNYEZET | 4 |
| WEBOLDAL TELEPÍTÉSE ÉS ELINDÍTÁSA..... | 4 |
| <i>A weblapon használt adatszerkezetek:</i> | <i>5</i> |
| FONTOSABB FORRÁSKÓD RÉSZLETEK | 12 |
| FELHASZNÁLÓI DOKUMENTÁCIÓ | 15 |
| <i>A Weboldal célja</i> | <i>15</i> |
| <i>Weboldal rövid bemutatása</i> | <i>15</i> |
| FEJLESZTÉSI LEHETŐSÉGEK | 23 |
| TESZT DOKUMENTÁCIÓ | 25 |
| IRODALOMJEGYZÉK | 26 |

BEVEZETÉS

Témaválasztás indoklása

Az életemben mindig is kiemelkedő szerepet játszott a kávézás, és számomra fontos, hogy minden napot egy kiváló kávéval indítsak. Azért hoztam létre weboldalamat, hogy lehetőséget teremtsék az egyedi pörkölésű kávék vásárlására, melyeknek úgy gondolom, hogy Magyarországon még nem élveznek széles körű elterjedtséget.

Ezen különleges pörkölési eljárásnak köszönhetően kávéink egyedülálló ízvilágot kínálnak, amelyekkel minden egyes reggel különlegessé tehető. Szívügyemnek tekintem, hogy minőségi kávéval örvendeztessem meg az embereket, és weboldalamon különleges választékot találhatsz azok számára, akik megoszthatják ezt a szenvedélyt velem.

FEJLESZTŐI DOKUMENTÁCIÓ

Fejlesztői Környezet

A weboldal fejlesztése során korszerű és hatékony eszközöket alkalmaztam a legjobb eredmény eléréséhez.

Kódszerkesztés: A weboldal kódjának írásához a Visual Studio Code-ot (VS Code) választottam. A VS Code egy könnyen kezelhető, kiválóan testreszabható kódszerkesztő, amely segít a hatékony fejlesztésben és hibakeresésben.

Adatbázis kezelés: Az adatbázis részéhez XAMPP-ot használtam. A XAMPP egy ingyenes és nyílt forráskódú szoftvercsomag, amely lehetővé teszi az adatbázisok létrehozását és kezelését. Ezzel az eszközzel könnyen és biztonságosan működtetheted az adatbázisodat.

Weboldal telepítése és elindítása

- Visual Studio Code telepítése
- XAMPP telepítése és Apache / MySQL indítása
- <http://localhost/phpmyadmin/> oldalon hozzon létre egy új táblát coffee_shop néven
- coffee_shop.sql adatbázist importálja be az újonnan létrehozott coffee_shop táblába.
- GitHub – ról letöltött Vizsgaprojekt fájl átnevezése Webpage-main – ról Webpage -re
- Helyezze a Webpage mappát a XAMPP/htdocs mappába

Mindezek után a weboldal elérhetővé válik a <http://localhost/webpage/> címen

Admin felület tesztelés céljából előre létrehozott admin felhasználó bejelentkezési adata:

Felhasználónév : Admin

Jelszó : Admin

Átlag user teszt felhasználó adatai:

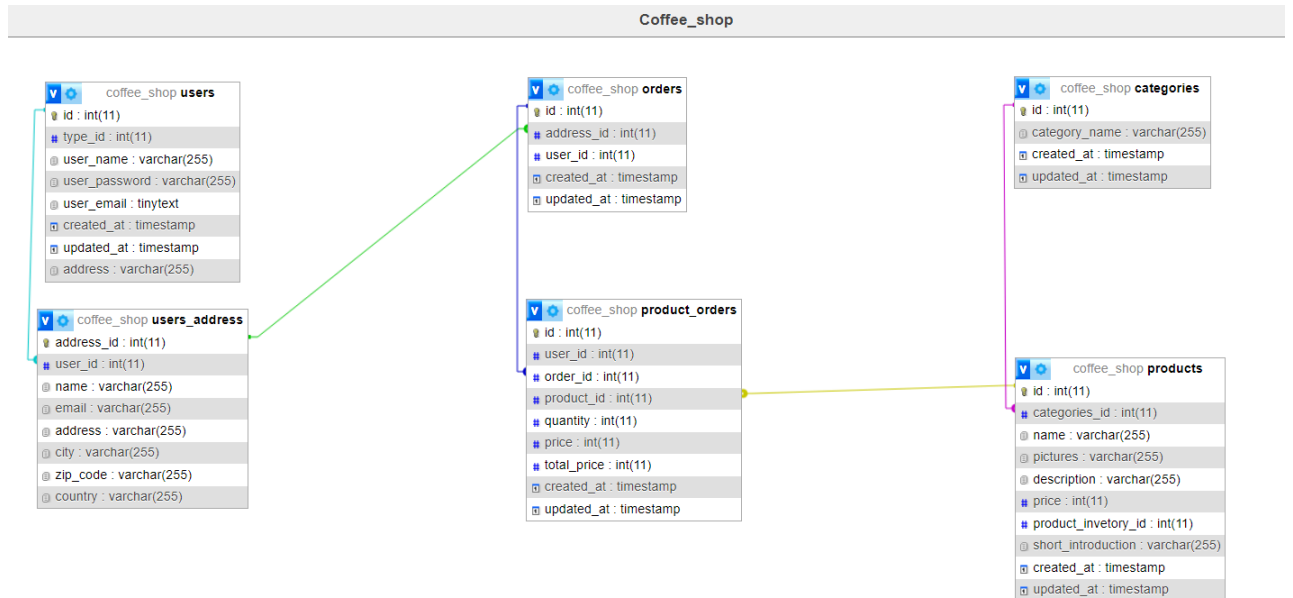
Felhasználónév: Test

Jelszó: Test

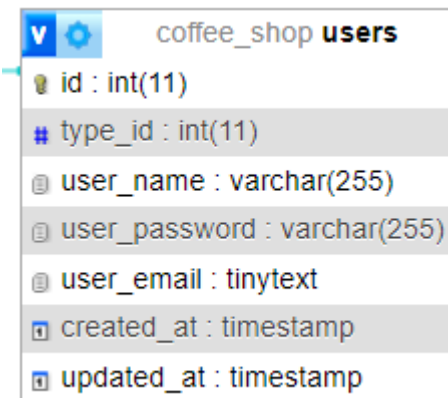
A weblapon használt adatszerkezetek:

A következő képeken látható lesz az adatszerkezet és a táblák különböző szerepének a részletes lebontása.

Az adatbázis felépítése



A users tábla



| coffee_shop users | |
|-------------------|--------------|
| id | int(11) |
| type_id | int(11) |
| user_name | varchar(255) |
| user_password | varchar(255) |
| user_email | tinytext |
| created_at | timestamp |
| updated_at | timestamp |

A users tábla a felhasználók alapvető adatainak tárolására szolgál. Ebben a táblában rögzítjük azokat az információkat, amelyek nélkülözhetetlenek a felhasználók bejelentkezéséhez a weboldalon. Az alábbi főbb adatok találhatók ebben a táblában:

user_name (Felhasználónév): A felhasználó által választott név, amely az azonosítás alapjául szolgál a weboldalon történő bejelentkezéskor.

user_password (Jelszó): A felhasználó által választott jelszó, amelyet biztonságosan tárolunk a password_hash segítségével.

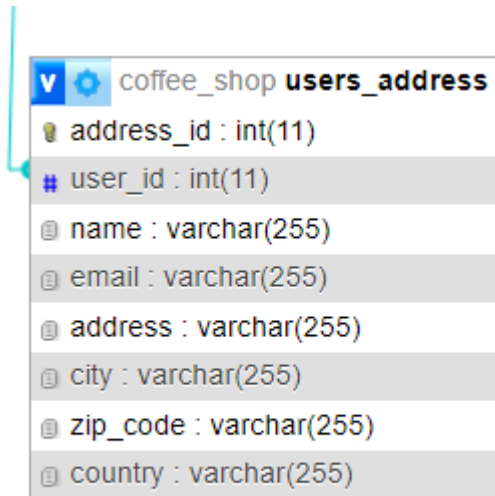
user_email (E-mail cím): A felhasználó által megadott e-mail cím, amely a kommunikációhoz és azonosításhoz szükséges.

Ezen felül a táblában két további fontos mező is található:

created_at (Létrehozás ideje): Ez az időpont rögzíti, mikor hozták létre a felhasználói rekordot a táblában. Segít nyomon követni, mikor regisztráltak az egyes felhasználók.

updated_at (Frissítés ideje): Ez az időpont rögzíti, mikor utoljára frissítették a felhasználói rekordot. Ennek a mezőnek köszönhetően könnyen nyomon követhető, mikor történtek változások a felhasználói adatokban.

User_address Tábla:

A screenshot of a database management interface showing the structure of a table named 'coffee_shop users_address'. The table has eight columns: 'address_id' (int(11)), 'user_id' (int(11)), 'name' (varchar(255)), 'email' (varchar(255)), 'address' (varchar(255)), 'city' (varchar(255)), 'zip_code' (varchar(255)), and 'country' (varchar(255)). The 'address_id' and 'user_id' columns are highlighted with a blue background, indicating they are primary keys. A green checkmark is visible next to the 'address_id' column.

| coffee_shop users_address | |
|---------------------------|--------------|
| address_id | int(11) |
| user_id | int(11) |
| name | varchar(255) |
| email | varchar(255) |
| address | varchar(255) |
| city | varchar(255) |
| zip_code | varchar(255) |
| country | varchar(255) |

A **users_address** tábla a felhasználók lakcíméhez kapcsolódó adatok tárolására szolgál. Az alábbi mezők találhatók ebben a táblában:

address_id (Lakcím azonosító): Egy egyedi azonosító minden lakcímhez.

user_id (Felhasználó azonosító): Ez a mező kapcsolódik a users tábla id mezőjéhez, így meghatározható, melyik felhasználóhoz tartozik a lakcím.

name (Név): A felhasználó neve, amely összekapcsolható a lakcímével.

email (E-mail cím): A felhasználó e-mail címe, amely egyéb kommunikációhoz vagy azonosításhoz szükséges lehet.

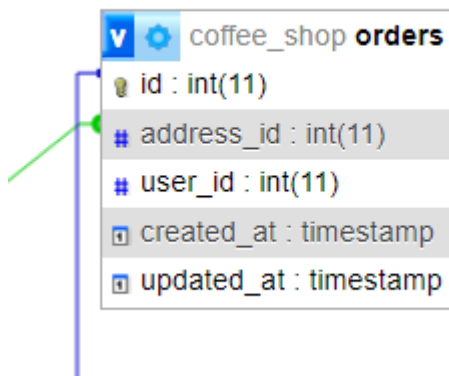
address (Lakcím): Az utca neve és a házszám, ahol a felhasználó él.

city (Város): A város vagy település, ahol a felhasználó lakik.

zip_code (Írányítószám): Az irányítószám, ami segít azonosítani a címet.

country (Ország): Az ország, ahol a felhasználó lakcíme található.

Orders tábla :



| | coffee_shop orders |
|------------|--------------------|
| id | : int(11) |
| address_id | : int(11) |
| user_id | : int(11) |
| created_at | : timestamp |
| updated_at | : timestamp |

Az orders tábla a leadott rendeléseket tárolja, és tartalmazza a rendelésekhez kapcsolódó információkat. Az alábbi mezők találhatók ebben a táblában:

id (Rendelés azonosító): Egy egyedi azonosító minden rendeléshez.

address_id (Lakcím azonosító): Ez a mező kapcsolódik a users_address tábla address_id mezőjéhez, így meghatározható, melyik lakcímhez tartozik a rendelés.

user_id (Felhasználó azonosító): Ez a mező kapcsolódik a users tábla id mezőjéhez, így azonosítani lehet, melyik felhasználó helyezte el a rendelést.

created_at (Létrehozás ideje): Az időpont, amikor a rendelés létrejött. Ezt az időpontot rögzíti a rendszer a rendelés leadásakor.

updated_at (Frissítés ideje): Az időpont, amikor a rendelés utoljára frissítve lett. Ezt az időpontot akkor rögzíti a rendszer, amikor a rendelés adataiban változás történik.

Product_orders tábla :

| coffee_shop | |
|----------------|-----------|
| product_orders | |
| id | int(11) |
| user_id | int(11) |
| order_id | int(11) |
| product_id | int(11) |
| quantity | int(11) |
| price | int(11) |
| total_price | int(11) |
| created_at | timestamp |
| updated_at | timestamp |

A product_orders tábla a rendelésekhez kapcsolódó termékek főbb adatait tárolja, és tartalmazza a termékek rendeléshez való hozzárendelését. Az alábbi mezők találhatók ebben a táblában:

id (termék rendelés azonosító): Egy egyedi azonosító minden rendeléshez.

user_id (Felhasználó azonosító): Ez a mező kapcsolódik a users tábla id mezőjéhez, így azonosítani lehet, melyik felhasználó helyezte el a rendelést.

order_id (Rendelés azonosító): Ez a mező kapcsolódik az előzően említett orders tábla id mezőjéhez, így meghatározható, melyik rendeléshez tartoznak a termékek.

product_id (Termék azonosító): Ez az azonosító mutatja, melyik terméket rendelték meg. Ezt az azonosítót használva hivatkozhatunk a termék részleteire, például a termék nevére vagy leírására.

quantity (Mennyiség): Ennek a mezőnek az értéke mutatja, hány darabot rendeltek meg az adott termékből.

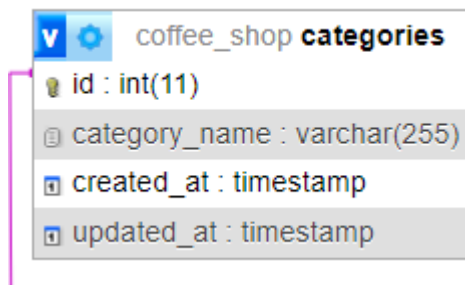
price (Ár): Ez az adat mutatja egy megadott termék árát.

total_price (Teljes Ár): Ez a mező számolja ki a rendelés során az adott termékből összesen megrendelt darabszám és az ár szorzatának az értékét ($\text{total_price} = \text{quantity} * \text{price}$).

created_at (Létrehozás ideje): Az időpont, amikor a rendeléshez kapcsolódó termék rekord létrejött. Ezt az időpontot rögzíti a rendszer, amikor egy terméket hozzárendelnek egy rendeléshez.

updated_at (Frissítés ideje): Az időpont, amikor a rendeléshez kapcsolódó termék rekord utoljára frissült. Ezt az időpontot akkor rögzíti a rendszer, amikor változás történik a termék rendelési részleteiben.

Categories tábla:



| coffee_shop categories | |
|------------------------|--------------|
| id | int(11) |
| category_name | varchar(255) |
| created_at | timestamp |
| updated_at | timestamp |

azonosításában.

A categories tábla a termékek főbb kategóriáit tárolja. Ezek a kategóriák segítenek az összes termék csoportosításában és kategorizálásában. Az alábbi mezők találhatók ebben a táblában:

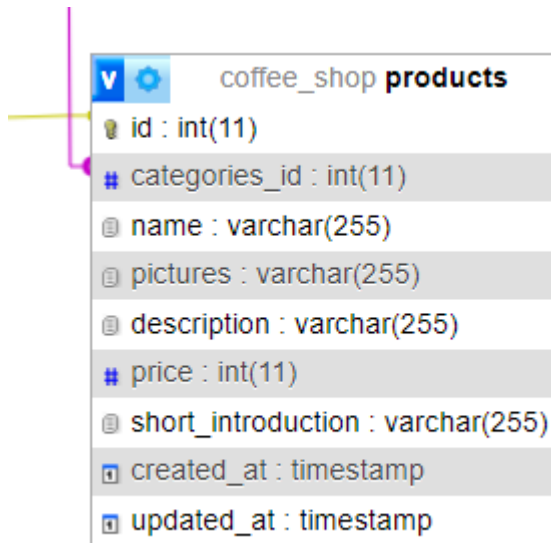
Id (Kategória azonosító): Egy egyedi azonosító minden kategóriához, ami segít azok egyedi

category_name (Név): A kategória neve, például "Blends," "Flavored_coffee," vagy "Single_origin_coffee."

created_at (Létrehozás ideje): Az időpont, amikor a kategória rekord létrejött. Ezt az időpontot rögzíti a rendszer, amikor egy új kategóriát hoznak létre.

updated_at (Frissítés ideje): Az időpont, amikor a kategória rekord utoljára frissült. Ezt az időpontot akkor rögzíti a rendszer, amikor változás történik a kategória adataiban.

Products tábla :



The screenshot shows a database table definition for 'coffee_shop products'. The table has the following columns: 'id' (int(11)) with a primary key icon, 'categories_id' (int(11)) with a foreign key icon, 'name' (varchar(255)), 'pictures' (varchar(255)), 'description' (varchar(255)), 'price' (int(11)) with a foreign key icon, 'short_introduction' (varchar(255)), 'created_at' (timestamp), and 'updated_at' (timestamp). A purple line points to the 'categories_id' column.

| Column | Data Type |
|--------------------|--------------|
| id | int(11) |
| categories_id | int(11) |
| name | varchar(255) |
| pictures | varchar(255) |
| description | varchar(255) |
| price | int(11) |
| short_introduction | varchar(255) |
| created_at | timestamp |
| updated_at | timestamp |

A products tábla a termékek részletes adatait tárolja, és kapcsolatban áll a kategóriákkal a categories_id mező segítségével. Az alábbi mezők találhatók ebben a táblában:

Id (Termék azonosító): Egy egyedi azonosító minden termékhez, ami segít azok egyedi azonosításában.

categories_id (Kategória azonosító): Ez a mező kapcsolódik a Categories tábla id mezőjéhez, így meghatározható, melyik kategóriához tartozik a termék.

name (Név): A termék neve, például "Blends."

Pictures (Kép): Egy kép vagy hivatkozás a termékre, amely megjeleníti a termék kinézetét.

description (Leírás): A termék részletes leírása, amely tartalmazza a termék részletes leírását

short_description (Rövid leírás): Egy rövid összefoglaló a termékről, amely gyorsan bemutatja a termék főbb jellemzőit.

price (Ár): Az adott termék ára.

created_at (Létrehozás ideje): Az időpont, amikor a termék rekord létrejött. Ezt az időpontot rögzíti a rendszer, amikor egy új terméket hoznak létre.

updated_at (Frissítés ideje): Az időpont, amikor a termék rekord utoljára frissült. Ezt az időpontot akkor rögzíti a rendszer, amikor változás történik a termék adataiban.

FONTOSABB FORRÁSKÓD RÉSZLETEK

A weboldalon több helyen is \$_SESSION használatával mentek adatokat.

Kosárkezelés a Cart.php-ban

A Cart.php -ban található meg a Session_cart osztályváltozó, amely minden kosárhoz tartozó függvényt és metódusokat tartalmaz, itt gondolok a termék kosárba való mentésére, törlésére és különböző adatok kinyerésére, pl: frissített termék mennyisége hozzáadást követően, a kosárba rakott termékek teljes összege. Két főbb függvényre térek ki ebben a részben.

```
public function getProductStructure($product, $qty, $price)
{
    return [
        'id' => $product[0]['id'],
        'name' => $product[0]['name'],
        'pictures' => $product[0]['pictures'],
        'price' => $price,
        'baseprice' => $product[0]['price'],
        'quantity' => $qty,
    ];
}
```

A képen látható metódus a kosár felépítésének a struktúráját tartalmazza.

Függvény: removeProductSession

Ez a PHP függvény felelős a kosárból egy adott termék eltávolításáért. A függvény egy paramétert vár, ami az eltávolítandó termék azonosítója (\$id). A funkció működése a következő lépésekből áll:

A \$_SESSION globális tömbben tárolt cart változó elemein iterálunk egy foreach ciklus segítségével.

Minden egyes elemnél ellenőrizzük, hogy a termék azonosítója (id) egyezik-e a megadott \$id értékkel.

Ha találunk egyezést, azaz a termék azonosítója megegyezik a megadott \$id értékkel, akkor eltávolítjuk azt a terméket a kosárból az unset funkcióval.

A függvény visszatér a kosár aktuális tartalmával, ami már a kiválasztott terméket nem tartalmazza.

```
public function removeProductSession($id)
{
    foreach ($_SESSION['cart'] as $key => $product_value) {
        if ($product_value['id'] == $id) {
            unset($_SESSION['cart'][$key]);
        }
    }
    return $_SESSION['cart'];
}
```

Termékek mentése a kosárba:

Ez a PHP függvény felelős a kosár tartalmának frissítésére. A függvény két paramétert vár: \$product és \$qty

A függvény működése a következő lépésekből áll.

Először is létrehoztam egy \$isExistProduct boolean változót, amelynek segítségével ellenőrzöm, hogy a termék már a kosárban van -e.

\$_SESSION['cart'] globális tömbön végig megyek a foreach ciklus segítségével és ellenőrzöm, hogy van -e már olyan termék a kosárban, amelynek megegyezik az azonosítója a megadott \$product[0]['id'] értékkel.

Ha a termék már megtalálható a kosárban (\$isExistProduct == true), akkor meghívja az updateProductQuantitySession függvényt a termék azonosítójával és az új mennyiségi értékkel.

Ha a termék még nem található a kosárban (\$isExistProduct == false), akkor kiszámolja az teljes árat (\$price = \$product[0]['price'] * \$qty) a megadott mennyiség alapján, majd hozzáadja a terméket a kosárhoz a getProductStructure függvény segítségével.

Ez a függvény lehetővé teszi a kosár tartalmának dinamikus kezelését és a felhasználóknak az aktuális vásárlás során történő könnyű termékek hozzáadását vagy frissítését.

```
public function setCartSession($product, $qty)
{
    $isExistProduct = false;
    foreach ($_SESSION['cart'] as $product_value) {
        if ($product_value['id'] == $product[0]['id']) {
            $isExistProduct = true;
        }
    }
    if ($isExistProduct == true) {
        $this->updateProductQuantitySession($product[0]['id'], $qty);
    }
    if ($isExistProduct == false) {
        $price = $product[0]['price'] * $qty;
        $_SESSION['cart'][] = $this->getProductStructure($product, $qty, $price);
    }
}
```

FELHASZNÁLÓI DOKUMENTÁCIÓ

A Weboldal célja

Az oldal célja, hogy az egyedi pörkölésű kávékat itthoni vásárlók számára elérhetővé tegye, így népszerűsítve a minőségi kávé élvezetét.

Hardware és software igény

Az oldal használatához szükséges egy számítógép, internetkapcsolat és egy webböngésző.

Weboldal rövid bemutatása

Az oldalon bejelentkezés nélkül használható dolgok:

- Coffe - Termékek böngészése és kosárhoz való hozzáadása

- About - Oldalon való rövidebb leírás a weboldalról

- Kosár – Kosár szerkesztése

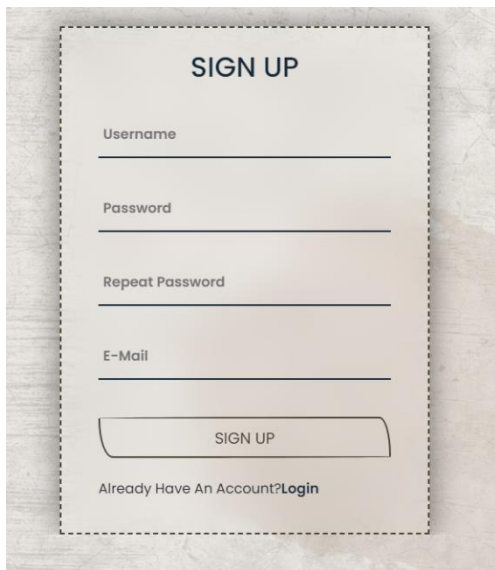
Az oldalon csak bejelentkezéssel elérhető funkciók:

- Felhasználói fiókon belül lakcím megadása


- Rendelés leadás

- Admin felület, amennyiben jogosultsággal rendelkezik.

Regisztráció, bejelentkezés



A registration form titled "SIGN UP" with a dashed border. It contains four input fields: "Username", "Password", "Repeat Password", and "E-Mail". Below the fields is a "SIGN UP" button. At the bottom, it says "Already Have An Account? [Login](#)".

A regisztráció elengedhetetlen része a weboldalnak, mivel a főbb funkciója a rendelés csak ezáltal érhető el. A hozzáférés a weboldal navigációs sávjának jobb oldalán található  ikonjával érhető el.



A login form titled "LOGIN" with a dashed border. It contains two input fields: "Username" and "Password". Below the fields is a "LOGIN" button. At the bottom, it says "Dont Have An Account? [Register](#)".

A Bejelentkezés rész segítségével a létrehozott felhasználóval beléphetünk a weboldalra és ezáltal megnyílik számunkra a teljes felhasználói rész.

Felhasználói felület

Test

Set Address

Check Orders

Shipping Address

Name

Test

Email

Test@Gmail.Com

Street Address

Test Utca 21

City

TestCity

Country

Test_megye

Zip/Postal Code

2022

Bejelentkezés után a felhasználói nevünkre való kattintás után elérhetővé válik számunkra a felhasználói oldal.

Itt lehetőség van a **Set address** funkcióval új lakcímet megadni vagy módosítani. A **Check orders** – el pedig megnézhetjük a jelenleg leadott rendeléseink státuszát.

Set Address

Test

Please Fill In Your Shipping Address!

Name

Email

Street Address


City

Country

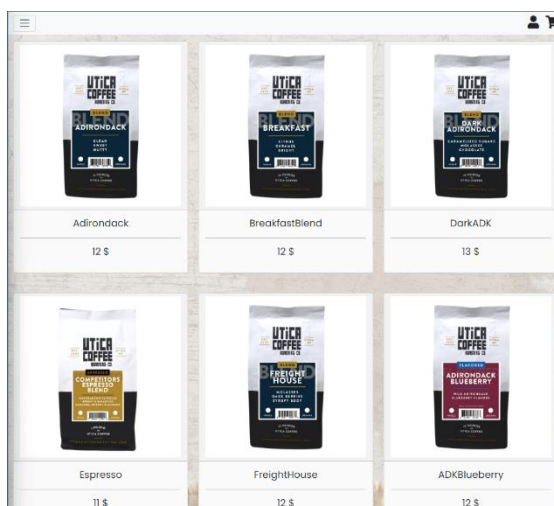
Zip/Postal Code

Save

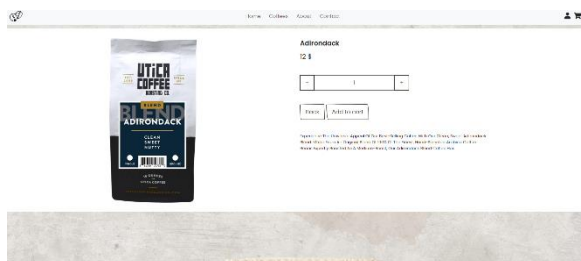
Check Orders

| Manage Orders | | | | | ← Back |
|---------------|---------------------------------|--------------|-------|--|--------|
| Order Time | Product And Quantity | Address | Total | Status | |
| 2023-09-03 | Adirondack (3),CoconutCream (1) | Test Utca 21 | 49 \$ |  Recieved Order | |

Termék kiválasztás

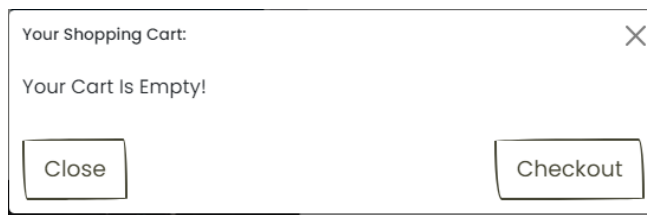



A Coffees menüpont segítségével böngészhetünk az aktuálisan elérhető kávék között és amennyiben bármelyik megtetszik számunkra, a termékre kattintva beléphetünk a belső menübe, ahol hozzá adhatjuk a kosárhoz.





Itt meghatározhatjuk a termék mennyiségét és hozzá adhatjuk a kosárhoz vagy visszaléphetünk a termékek böngészéséhez.

Kosár



A weboldalon a kosár funkció eléréshez a kosár gomb  megnyomásával léphetünk be.

Your Shopping Cart:

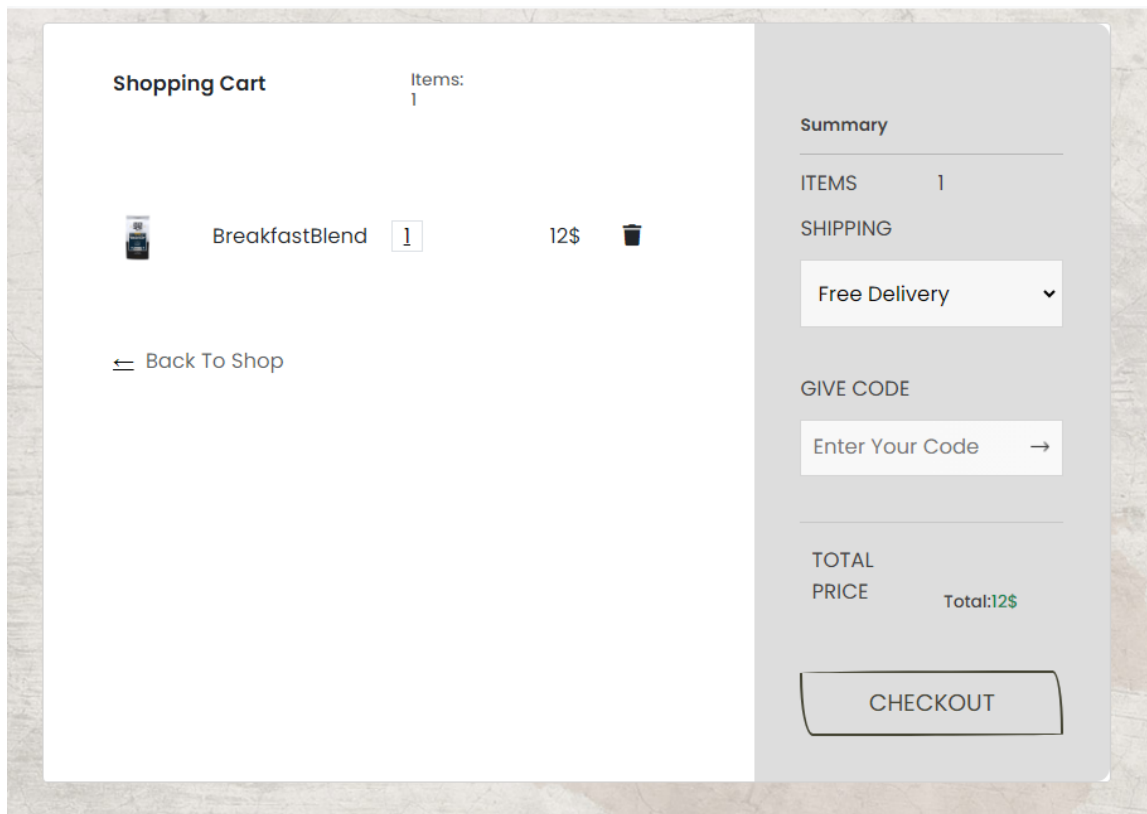
| | Name | Qty | Total | Action |
|---|--------------|-----|-------|---|
|  | Adirondack | 3 | 36\$ |  |
|  | CoconutCream | 1 | 13\$ |  |

Total:49\$



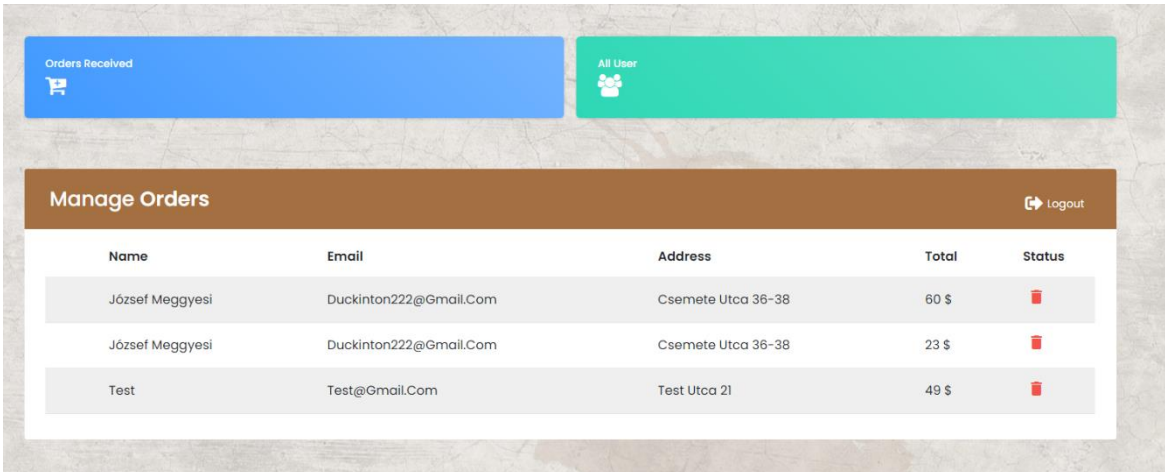
✕ Termékek hozzáadása után lehetőségünk van tovább haladni a (checkout) az – az rendelés leadás oldalra vagy tudjuk szerkeszteni a kosarat, törölni a nem kívánt termékeket.

Rendelés leadása






Amennyiben bejelentkezett a felhasználó és megadta a lakcímét, akkor a kosárban lévő checkout gombra kattintva átnavigál az oldal a megrendelés oldalra, ahol leadhatja a rendelését.

Admin felület



The screenshot displays the Admin Panel interface. At the top, there are two buttons: 'Orders Received' (blue) and 'All User' (teal). Below these is a 'Manage Orders' section with a 'Logout' link. The main content is a table with the following data:

| Name | Email | Address | Total | Status |
|-----------------|------------------------|--------------------|-------|---|
| József Meggyesi | Duckinton222@Gmail.Com | Csemete Utca 36-38 | 60 \$ |  |
| József Meggyesi | Duckinton222@Gmail.Com | Csemete Utca 36-38 | 23 \$ |  |
| Test | Test@Gmail.Com | Test Utca 21 | 49 \$ |  |

Amennyiben egyes `type_id` – val van ellátva a bejelentkezett felhasználó, akkor bejelentkezés után rögtön az admin felülethez kerül. Itt megtekintheti az összes leadott rendelést és amennyiben szükséges törölheti azokat.

About oldal

The History Of Pupa Coffee Roasting Co.



Once Upon A Time, In A Small Town Nestled Amidst Rolling Hills, There Lived A Passionate Coffee Enthusiast Named Sarah. Sarah Had A Deep Love For Coffee And A Dream Of Creating A Place Where People Could Experience The True Joy Of Freshly Roasted Beans.

With Unwavering Determination And A Heart Full Of Ambition, Sarah Embarked On A Journey To Turn Her Dream Into Reality. She Spent Countless Hours Studying The Art Of Coffee Roasting, Experimenting With Different Beans And Flavors, And Perfecting Her Craft. Her Dedication And Perseverance Paid Off As Her Coffee Creations Garnered Praise From Friends, Family, And Even Local Coffee Connoisseurs.

Ezen az oldalon egy rövid bemutatást olvashatunk a cég megalapulásáról.

Contact oldal

Contact Us

Name

Email

Message

Itt lehetőség van üzenetet küldeni az oldal tulajdonosának.

FEJLESZTÉSI LEHETŐSÉGEK

Contact oldal:

```
<?php
if (isset($_POST["submit"])) {
    $shopEmailAddress = 'mevama4122@tipent.com';
    $name = $_POST["name"];
    $email = $_POST["email"];
    $message = $_POST["message"];

    // $mail->isSMTP(); // Set mailer to use SMTP
    // $mail->Host = 'smtp1.example.com;smtp2.example.com'; // Specify main and backup SMTP servers
    // $mail->SMTPAuth = true; // Enable SMTP authentication
    // $mail->Username = 'user@example.com'; // SMTP username
    // $mail->Password = 'secret'; // SMTP password
    // $mail->SMTPSecure = 'tls'; // Enable encryption, 'ssl' also accepted

    // $mail->From = 'from@example.com';
    // $mail->FromName = 'Mailer';
    // $mail->addAddress('joe@example.net', 'Joe User'); // Add a recipient
    // $mail->addAddress('ellen@example.com'); // Name is optional
    // $mail->addReplyTo('info@example.com', 'Information');
    // $mail->addCC('cc@example.com');
    // $mail->addBCC('bcc@example.com');

    // $mail->WordWrap = 50; // Set word wrap to 50 characters
    // $mail->addAttachment('/var/tmp/file.tar.gz'); // Add attachments
    // $mail->addAttachment('/tmp/image.jpg', 'new.jpg'); // Optional name
    // $mail->isHTML(true); // Set email format to HTML

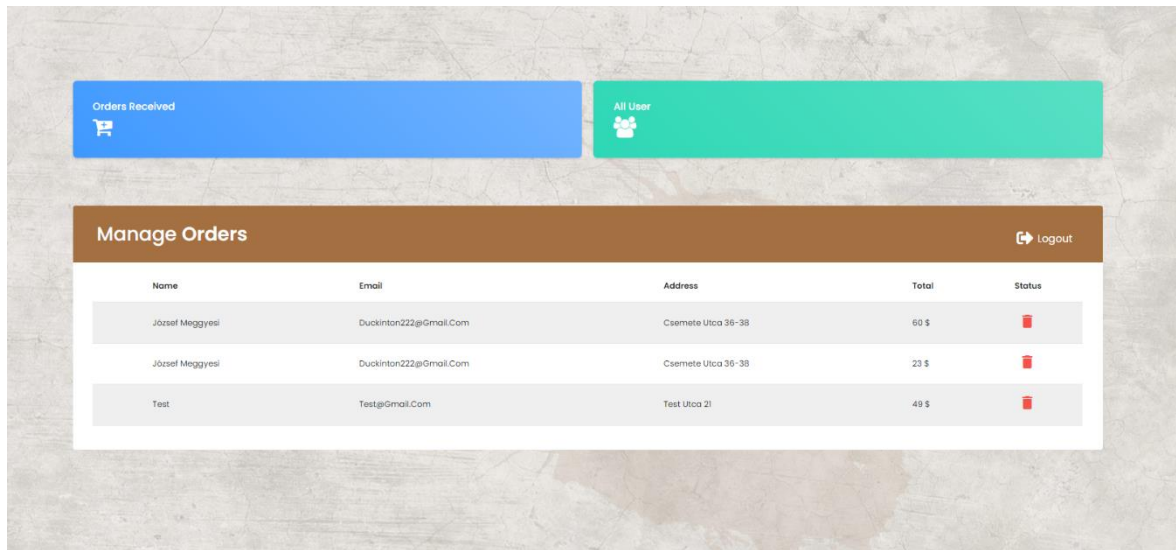
    // $mail->Subject = 'Here is the subject';
    // $mail->Body = 'This is the HTML message body <b>in bold!</b>';
    // $mail->AltBody = 'This is the body in plain text for non-HTML mail clients';

    // if (!$mail->send()) {
    //     echo 'Message could not be sent.';
    //     echo 'Mailer Error: ' . $mail->ErrorInfo;
    // } else {
    //     echo 'Message has been sent';
    // }

    header("location: http://localhost/webpage/?oldal=&error=YourMessageHasBennSend");
}
```

A jövőben tervezem bevezetni az email küldés lehetőségét. Amennyiben a felhasználó üzenetet szeretne küldeni a kapcsolat oldalon keresztül, az üzenet automatikusan továbbításra kerül majd az oldal tulajdonosa felé. Ehhez a funkcióhoz az alábbi kódrészletet fogom felhasználni.

Admin felület



Tervezem az admin felületen az orders recieved és az all users részekben megjeleníteni, hogy hány leadott rendelés van és mennyi felhasználó van regisztrálva az oldalon. Az admin page teljes mértékben reszponzívva tétele.

Regisztráció és bejelentkezés

Részletesebb visszajelzés és adatellenőrzés mindkét felületnél.

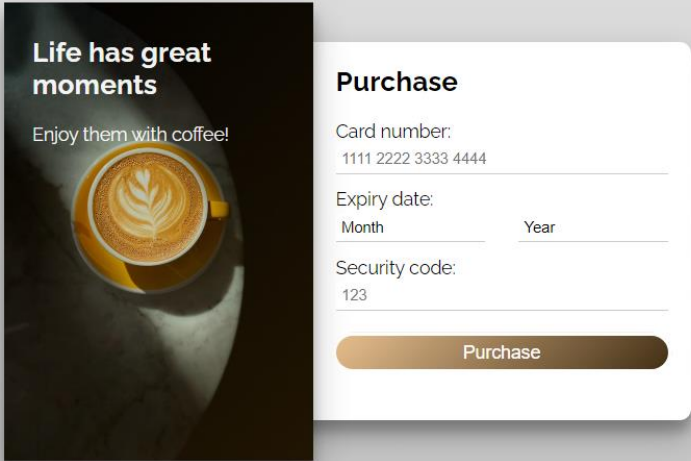
Teszt dokumentáció

Az oldal le lett tesztelve háromfajta böngészővel:

- Firefox
- Google Chrome
- Microsoft Edge

Megkértem 3 közeli ismerősome, a weboldal tesztelésére.

Az egyik fő megjegyzés az volt, hogy a rendelés leadás folyamata során hiányzott a bankkártya adatok bekérése a checkout oldalon. Ennek kijavítására egy új oldalt hoztam létre, amely a "checkout" gombra kattintva jelenik meg, és itt lehet megadni a bankkártya információkat.



The screenshot shows a checkout form titled "Purchase" on a light gray background. On the left is a promotional image for coffee with the text "Life has great moments" and "Enjoy them with coffee!". The form itself is white and contains the following fields:

- Card number:** 1111 2222 3333 4444
- Expiry date:** Month (dropdown) and Year (dropdown)
- Security code:** 123
- Purchase button:** A brown button with the text "Purchase".

IRODALOMJEGYZÉK

- <https://www.w3schools.com>
- <http://php.net/>
- <https://www.codecademy.com>
- <https://stackoverflow.com>