

PureLine

Tartalomjegyzék

- Bevezetés
 - Záródolgozat téma kiválasztása
 - Alkalmazás tervezése
 - Programozási nyelv kiválasztása, előkészítés
- Fejlesztői Dokumentáció
 - Adatbázis
 - Mappa struktúra és fájlok
- PHP Funkciók és Kódok
 - Fő funkciók
 - Adminisztráció

1. Bevezetés

1.1 Záródolgozat téma kiválasztása

A vizsgaremek elkészítése során háromfős fejlesztői csapatként dolgoztunk együtt egy komplex szoftveralkalmazáson, amely egy valós piaci problémára kínál megoldást. Olyan témát választottunk, amely lehetővé tette egy mindenki számára elérhető, webes alapú webshop létrehozását. A webshop rezponzív kialakításának köszönhetően minden eszközön, legyen az asztali számítógép, táblagép vagy mobiltelefon, zökkenőmentesen működik.

Az ötletet és inspirációt egy ismerősünk ruhaboltja adta, aki saját üzlete számára keresett modern, felhasználóbarát megoldást. Az alkalmazás egyik fő előnye, hogy a vásárlóknak nem kell fizikailag felkeresniük az üzletet, hiszen az online felület lehetőséget biztosít a termékek kényelmes böngészésére és megrendelésére akár otthonról is. Ezzel időt és energiát takarítanak meg, miközben az eladó számára is egyszerűbbé válik az adminisztráció és az ügyfélkapcsolatok kezelése.

A fejlesztés során különös figyelmet fordítottunk arra, hogy az alkalmazás megfeleljen a korszerű technológiai követelményeknek, miközben a felhasználói élményt is kiemelten kezeltük, hogy a rendszer egyszerre legyen praktikus és könnyen használható.

1.2 Alkalmazás tervezése

A projekt kezdetén a szoftveralkalmazás tervezése során az elsődleges célunk egy könnyen navigálható, rezponzív, modern és felhasználóbarát webshop létrehozása volt, amely minden eszközön optimálisan működik. A tervezési fázisban fontos szerepet kapott a célcsoport igényeinek feltérképezése, valamint a felhasználói élmény (UX) és a vizuális dizájn (UI) összhangja.

Első lépésként meghatároztuk a webshop alapvető funkcióit, amelyek közé tartozott a termékek könnyű kereshetősége, szűrhetősége, biztonságos és gyors vásárlási folyamat, valamint az egyszerű navigáció. A felhasználóknak egy átlátható, gyors és élvezetes vásárlási élményt szerettünk volna biztosítani, ezért a dizájn során kiemelt figyelmet fordítottunk a reszponzív megjelenítésre, hogy mobiltelefonokon, tableteken és asztali számítógépeken egyaránt kifogástalanul működjön.

A weboldal struktúráját úgy terveztük meg, hogy az egyszerűen bővíthető és módosítható legyen, figyelembe véve a jövőbeli igényeket, mint például új termékkategóriák hozzáadása, promóciók, vagy bármilyen új funkciók integrálása. A dizájn egy letisztult, modern arculatot kapott, amely lehetővé teszi a gyors navigációt, ugyanakkor esztétikus és vonzó a felhasználók számára.

A tervezés során külön figyelmet fordítottunk a webshop adminisztrációs felületére is. Az adminisztrációs oldal egy könnyen kezelhető felületet biztosít a termékek kezelésére, az egyes rendelések nyomon követésére, valamint a felhasználói adatok, illetve termékek kezelésére. Az admin felület is reszponzív, hogy bárholnan elérhető és kényelmesen használható legyen.

A felhasználói élmény mellett az alkalmazás technikai oldalát is megterveztük. Az adatbázis felépítése során figyelembe vettük a webshop működéséhez szükséges összes adatot, például a termékek, vásárlók, rendelési információk tárolását. Az adatbázis modell kapcsolódik a front-end és back-end rendszerekhez, biztosítva ezzel a megfelelő működést.

1.3 Programozási nyelv kiválasztása, előkészítés

A PureLine webshop fejlesztése során

Kiemelt figyelmet fordítottunk arra, hogy a frontend és backend technológiák optimális kombinációjával egy funkcionálisan stabil, gyors és felhasználóbarát rendszert hozzunk létre. A PureLine célja nemcsak egy esztétikus és modern megjelenésű felület biztosítása volt, hanem egy olyan webshopé, amely egyszerűen navigálható, gördülékenyen működik, és megfelelően kezeli mind az ügyféloldali, mind a háttérben zajló folyamatokat.

A fejlesztés során:

- **Frontend technológiákkal** letisztult dizájnt és reszponzív felhasználói felületet hoztunk létre, amely tökéletesen alkalmazkodik asztali és mobil eszközökhöz.
- **Backend technológiák** segítségével biztosítottuk az adatok biztonságos tárolását, gyors feldolgozását és a webshop alapvető funkcióinak, például a rendeléskezelés és a felhasználói bejelentkezés hitelesítés működését.

A PureLine webshop esetében az alábbi technológiák együttes alkalmazásával érhető el a kívánt eredmény:

1. Frontend technológiák (HTML, CSS, JavaScript, Bootstrap):

Ezek felelősek a vizuális megjelenésért és a felhasználói interakciókért, például a termékek böngészéséért, a kosár kezeléséért és az űrlapok kitöltéséért.

2. Backend technológiák (PHP):

A háttérben futó folyamatok felelősek az adatok biztonságos tárolásáért, feldolgozásáért és a logikák működéséért.

- A PHP biztosítja a szerveroldali logikát, például a rendelések feldolgozását, a felhasználói fiókok kezelését és az adatbázisokkal való kommunikációt.
- Az AJAX lehetővé teszi az aszinkron adatlekérést, amely gyors és dinamikus adatcserét tesz lehetővé a frontend és backend között a REST API-n keresztül, elkerülve az oldal teljes újratöltését.

Ez a kombináció hatékony működést biztosít a PureLine webshop számára, és lehetővé teszi a valós idejű, gördülékeny felhasználói élményt.

Az alábbiakban bemutatjuk a PureLine webshopnál használt frontend és backend technológiákat. Ez a struktúra biztosítja a hatékony működést, skálázhatóságot és a kiváló felhasználói élményt.

Frontend technológiák

A frontend az a réteg, amelyet a felhasználók közvetlenül látnak és használnak. Ez magában foglalja a vizuális elemeket, az interaktív funkciókat és a felhasználói élmény biztosítását.

HTML (HyperText Markup Language)

A HTML a weboldal alapvető vázát képezi, amely meghatározza az egyes elemek elhelyezkedését, tartalmát és szerkezetét.

- Jellemzők:
 - Statikus elemek felépítése (címek, képek, bekezdések, linkek).
 - Hipertext kapcsolatok és formázási struktúrák létrehozása.
 - Az összes böngésző támogatja.
 - Egyszerű és gyors integrálás más technológiákkal, például CSS-sel és JavaScript-tel.
-

CSS (Cascading Style Sheets)

A CSS biztosítja a weboldal megjelenését, az elrendezéstől kezdve a színek és animációk megadásáig.

- Jellemzők:
 - Elrendezési modellek, például rácsrendszer (grid) és rugalmas boxmodell (flexbox) támogatása.
 - Média lekérdezések segítségével reszponzív design kialakítása.
 - Könnyű testreszabhatóság.
-

JavaScript (JS)

A JavaScript az interaktív funkciókat és dinamikus működést biztosítja. Lehetővé teszi, hogy az oldal elemei valós időben változzanak, anélkül, hogy az oldalt újra kellene tölteni.

- Jellemzők:
 - Eseményvezérelt működés, például kattintások, görgetések, billentyűlenyomások kezelése.
 - DOM (Document Object Model) manipuláció a HTML tartalom módosításához.
 - Böngészőben futó kód, így nincs szükség telepítésre.
 - Számos keretrendszer támogatása (pl. React, Angular, Vue.js).
-

Bootstrap

A Bootstrap lehetővé teszi reszponzív weboldalak gyors létrehozását előre definiált stílusokkal és komponensekkel.

- Jellemzők:
 - Előre elkészített rácsrendszer (grid) a weboldalak elrendezéséhez.
 - Komponensek, például navigációs sávok, gombok, űrlapok gyors implementálása.
 - Mobilbarát kialakítás alapértelmezetten.
 - Könnyen tanulható és testreszabható.

Backend technológiák

A backend az alkalmazás háttérben működő része, amely kezeli az adatbázisokat, a szerveroldali logikát és a kliensoldalról érkező kéréseket.

PHP (Hypertext Preprocessor)

A PHP egy szerveroldali szkriptnyelv, amely az adatok feldolgozását, dinamikus tartalom generálását és a REST API működtetését végzi.

- **Jellemzők:**
 - Adatbázisokkal való közvetlen integráció (pl. MySQL, PostgreSQL).
 - Szerveroldali logikák kezelése, például hitelesítés, felhasználói adatok kezelése.
 - Nyílt forráskódú és széles körű támogatással rendelkezik.
 - Könnyen integrálható más technológiákkal, például HTML-lal és AJAX-szal.

AJAX (Asynchronous JavaScript and XML)

Az AJAX aszinkron adatkommunikációt biztosít a kliens és a szerver között, amely lehetővé teszi az oldalak frissítés nélküli adatlekérdezését.

- **Jellemzők:**
 - Adatok kezelése JSON vagy XML formátumban.
 - HTTP metódusok (GET, POST, PUT, DELETE) használata a REST API-val való kommunikáció során.
 - Gyors és hatékony adatfrissítés.
 - Kiválóan alkalmas valós idejű alkalmazásokhoz, például chat rendszerekhez vagy keresési javaslatokhoz.

1.4 Köszönetnyilvánítás

Köszönet a Fejlesztői Csoportnak

- **Projektvezető:**
Kiss Gábor, a folyamatok irányításáért és koordinálásáért.
- **Fejlesztők:**
[Kiss Máté Márk, Kiss Gábor, Takács Zoltán], akik a kódolásért és technikai megoldásokért feleltek.
- **UI/UX Tervezők:**
[Takács Zoltán, Kiss Gábor, Kiss Máté Márk], akik biztosították a felhasználóbarát és esztétikus megjelenést.

Oktató és Intézmény

Külön köszönetünket fejezzük ki **Bloch Tamás oktatónak**, aki szakmai irányításával segítette a projektünket.

Hálásak vagyunk a **KIÁKK intézménynek**, amely megfelelő kereteket és erőforrásokat biztosított a projekt létrejöttéhez.

Felhasználók

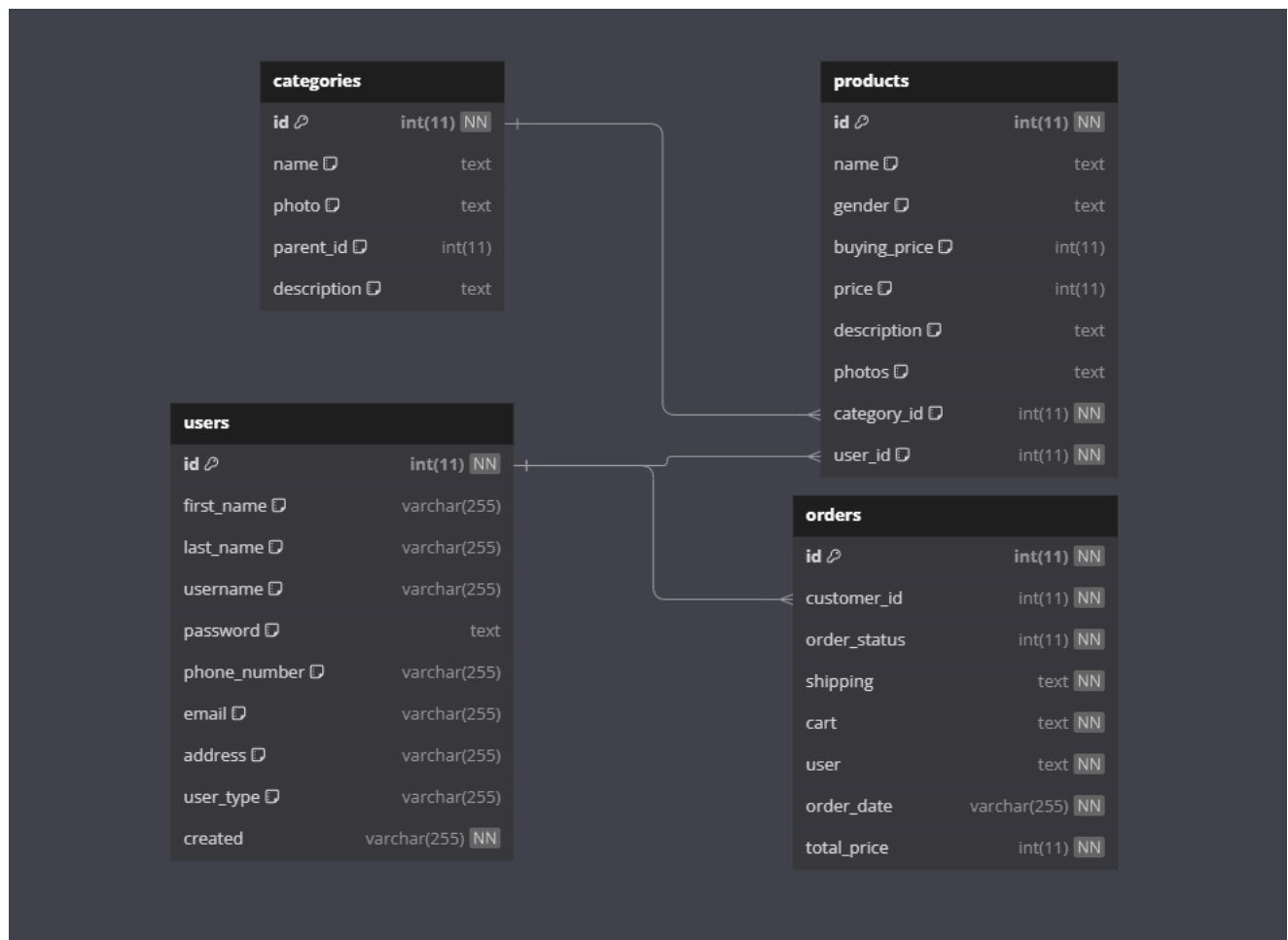
Hálásak vagyunk a felhasználóknak, akik visszajelzéseikkel és javaslataikkal segítették a projekt fejlesztését, valamint hozzájárultak ahhoz, hogy a végeredmény még jobb legyen.

2. Fejlesztői Dokumentáció

2.1 Adatbázis

Az alábbiakban bemutatjuk a shop adatbázis szerkezetét, tábláit, kapcsolatait és az adatbázis működéséhez szükséges fontos részleteket.

PureLine adatbázis diagram:



Kapcsolatok, amelyek az ábrát vezérelték:

1. `"categories"."id" < "products"."category_id"`

A termékek kategóriákhoz vannak rendelve, így ellenőrizni kell a kategóriákat termékek hozzáadásakor.

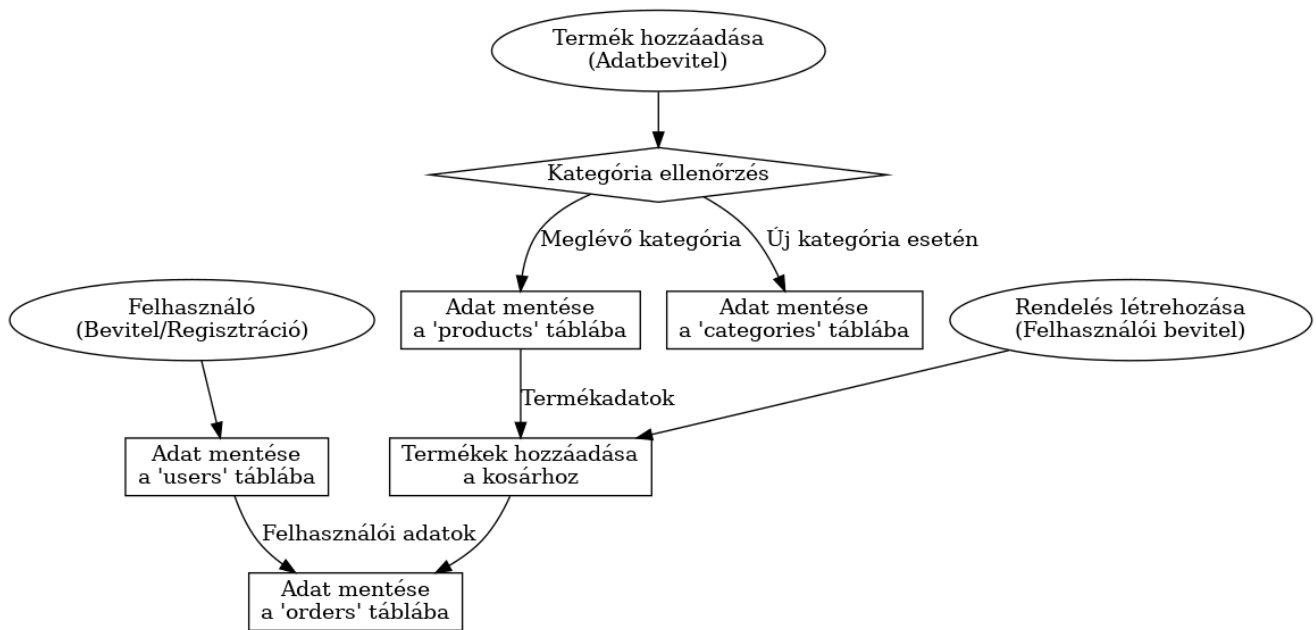
2. `"users"."id" < "orders"."customer_id"`

A felhasználók (vásárlók) adatai kapcsolódnak a rendelésekhez.

3. `"users"."id" < "products"."user_id"`

A termékekhez felhasználói azonosító (tulajdonos) tartozik.

PureLine adatbázis folyamatára:



Alakzatok jelentése a folyamatábrában:

Ovális: Folyamat kezdete vagy vége.

Téglalap: Adatok táblába mentése, műveletek végrehajtása.

Rombusz: Logikai döntés (pl. "Van-e ilyen kategória?").

Nyíl: Az adatok vagy műveletek sorrendje.

2.1.1 Általános információk

- **Adatbázis neve:** shop
- **Adatbázis-kezelő rendszer:** MariaDB 10.4.32
- **PHP verzió:** 8.2.12
- **Karakterkódolás:** utf8mb4_general_ci
- **Adatbázis-adminisztrációs eszköz:** phpMyAdmin 5.2.1

2.1.2 Tábla szerkezetek

2.1.2.1 "categories" tábla

Ez a tábla a kategóriák adatait tárolja.

- **Oszlopok:**
 - `id` (int): Egyedi azonosító, elsődleges kulcs, `AUTO_INCREMENT`.
 - `name` (text): A kategória neve.
 - `photo` (text): Kategória fotóinak JSON formátumú tárolása.
 - `parent_id` (int): Szülőkategória azonosítója (hierarchia támogatása).
 - `description` (text): Kategória leírása.
-

2.1.2.2 "orders" tábla

Az összes rendelés információját tárolja.

- **Oszlopok:**
 - `id` (int): Egyedi azonosító, elsődleges kulcs, `AUTO_INCREMENT`.
 - `customer_id` (int): A megrendelő azonosítója (users táblából).
 - `order_status` (int): Rendelés állapota.
 - **Ha 0:** Kiszállítva
 - **Ha 1:** Folyamatban
 - **Ha 2:** Késik
 - **Ha 3:** Törölve
 - `shipping` (text): Szállítási adatok JSON formátumban.
 - `cart` (text): Kosár tartalma JSON formátumban.
 - `user` (text): Felhasználói információk JSON formátumban.
 - `order_date` (varchar): Rendelés dátuma.
 - `total_price` (int): Rendelés összértéke.
-

2.1.2.3 "products" tábla

A termékek adatait tartalmazza.

- **Oszlopok:**
 - `id` (int): Egyedi azonosító, elsődleges kulcs, `AUTO_INCREMENT`.
 - `name` (text): Termék neve.
 - `gender` (text): Célcsoport (Man, Woman, Kid).
 - `buying_price` (int): Beszerzési ár.
 - `price` (int): Eladási ár.
 - `description` (text): Termék leírása.

- `photos` (text): Termék képei JSON formátumban.
 - `category_id` (int): Hivatkozás a `categories` táblára.
 - `user_id` (int): Hivatkozás a létrehozó felhasználóra (`users` tábla).
-

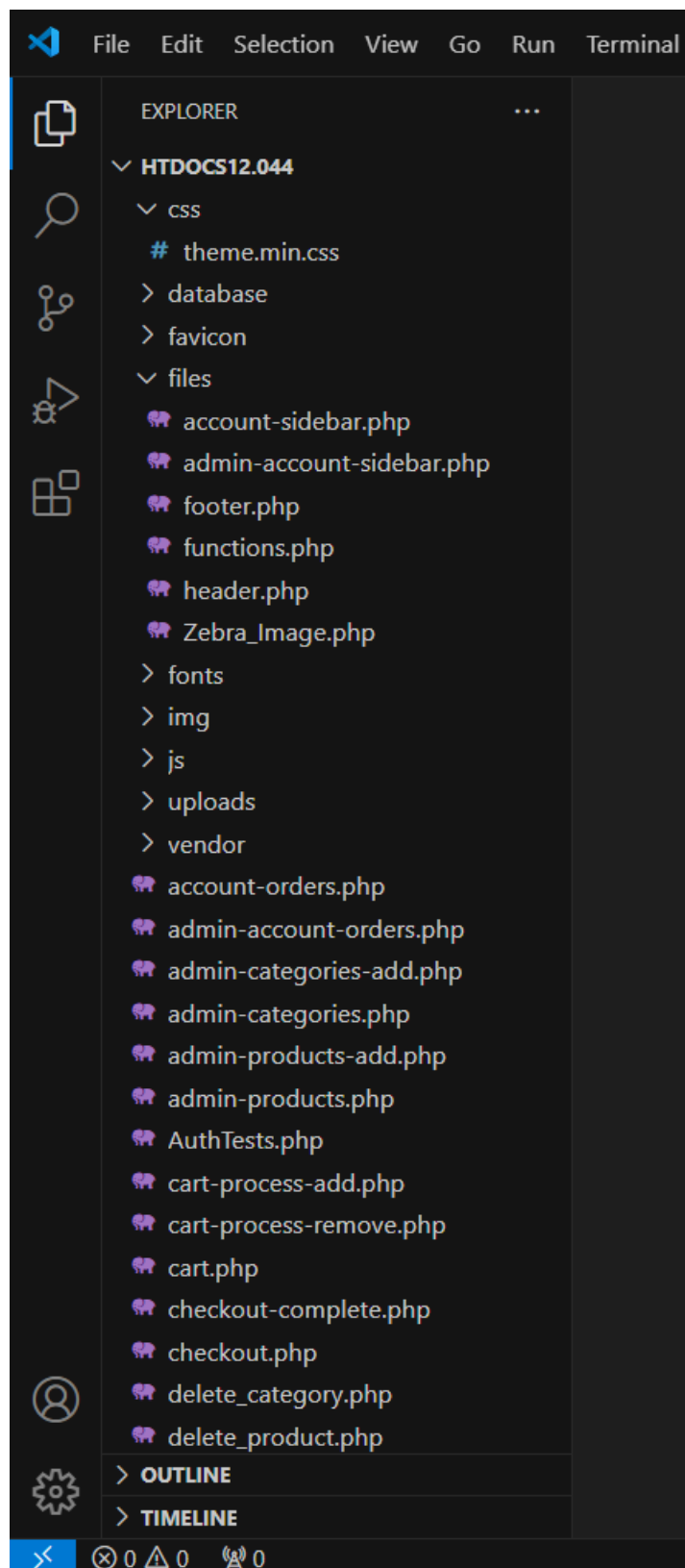
2.1.2.4 "users" tábla

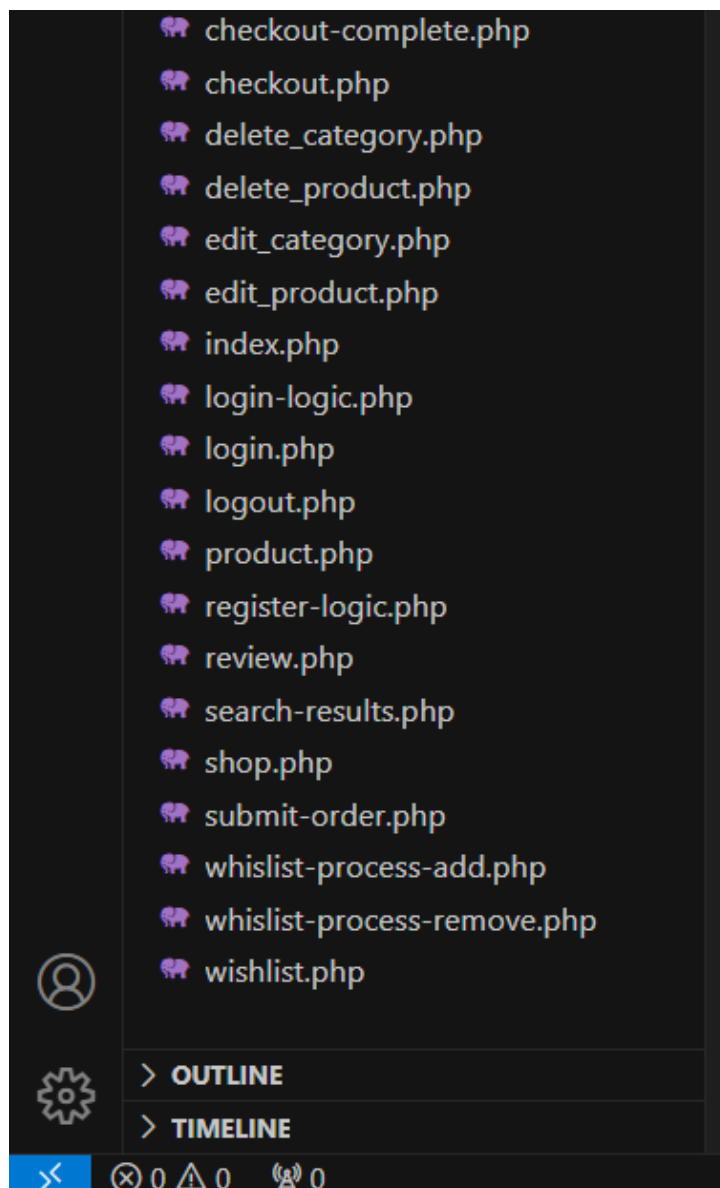
A felhasználói információkat tárolja.

- **Oszlopok:**
 - `id` (int): Egyedi azonosító, elsődleges kulcs, `AUTO_INCREMENT`.
 - `first_name` (varchar): Keresztnév.
 - `last_name` (varchar): Vezetéknév.
 - `username` (varchar): Felhasználónév.
 - `password` (text): Hash-elt jelszó.
 - `phone_number` (varchar): Telefonszám.
 - `email` (varchar): E-mail cím.
 - `address` (varchar): Cím.
 - `user_type` (varchar): Felhasználói típus (pl. `customer`, `admin`).
 - `created` (varchar): Fiók létrehozási dátuma.

2.2 *Mappa struktúra és fájlok*

Itt látható a PureLine Divat Webshop fájlrendszerének felépítése, amely különböző komponenseket tartalmaz az alkalmazás működtetéséhez. Íme a struktúra részletes leírása a főbb elemekkel:





2.2.1 Fájrendszer leírás

1. CSS

- Az alkalmazás stíluslapja található itt (CSS fájl), amely a weboldal megjelenését határozza meg.

2. Database

- A weboldal adatbázis fájlait tartalmazza.

3. Favicon

- Az itt lévő fájlok a weboldal faviconjait tartalmazzák különböző méreteken.

4. Files

A következő fájlokat tartalmazza:

- **account-sidebar.php**: Felhasználói oldalmenü megjelenítéséhez szükséges fájl.
- **admin-account-sidebar.php**: Admin oldalmenüjének sablonja.

- **footer.php**: Az oldal alsó részének (lábléc) megjelenítését kezelő fájl.
- **functions.php**: A webshop funkcióit tartalmazza.
- **header.php**: Az oldal felső részét (fejléc) tartalmazó sablonfájl.
- **Zebra_Image.php**: Egy képkezelő PHP fájl, amely képméretezési és optimalizálási funkciókat biztosít.

5. Fonts

- A webshop által használt betűtípusokat tartalmazza:
 - **cartzilla-icons435b.svg/ttf/woff**: Az alkalmazás harmadik féltől származó ikonfontjai SVG, TTF és WOFF formátumban.

6. Img

- Képfájlok mappája, például termékek, logók, egyéb médiaelemek.

7. JS

- Az alkalmazás Createx Studio által készített JavaScript fájlja található itt.

8. Uploads

- Felhasználók által feltöltött képeket tartalmazza.

9. Vendor

Harmadik féltől származó könyvtárak, pluginok és csomagok, amelyek az alkalmazás működéséhez szükségesek. Ezek a következőket tartalmazzák:

- **bootstrap**: A Bootstrap framework fájlljai, amelyek a front-end fejlesztést segítik.
 - **card**: Kártya-alapú UI komponensek megvalósításához szükséges eszközök.
 - **chartist**: Diagramok és grafikonok megjelenítéséhez használt könyvtár.
 - **drift-zoom**: Kép nagyításhoz kapcsolódó funkciókat biztosít.
 - **flatpickr**: Egy dátumválasztó (date-picker) JavaScript plugin.
 - **imagesloaded**: Képek betöltésének figyelésére szolgáló eszköz.
 - **lightgallery**: Galéria megjelenítéséhez használt plugin.
 - **nouislider**: Felhasználóbarát csúszkák készítéséhez használható plugin.
 - **prismjs**: Szintaxiskiemelő, amely a kódblokkok stílusozására használható.
 - **shufflejs**: Elemek keverésére és szűrésére használt könyvtár.
 - **simplebar**: Egyedi görgetősávok megvalósítását segítő plugin.
 - **smooth-scroll**: Egy JavaScript könyvtár, amely a sima görgetés (smooth scrolling) funkciót biztosítja.
 - **tiny-slider**: Egy könnyű és gyors csúszka (slider) plugin, amely képek és egyéb tartalmak lapozására szolgál.
-

2.2.3 PHP fájlok funkciói

1. **account-orders.php**

- Felhasználói rendelések megtekintésére szolgáló oldal.

2. **admin-account-orders.php**

- Adminisztrátor által kezelt rendeléseket listázza.

3. **admin-categories-add.php**

- Kategóriák hozzáadására szolgáló adminisztrátori felület.

4. **admin-categories.php**

- Kategóriák kezelése és listázása adminisztrátori oldalról.

5. **admin-products-add.php**

- Új termékek hozzáadása az admin felületen keresztül.

6. **admin-products.php**

- Az adminisztrátor által kezelt termékek listája.

7. **cart-process-add.php**

- Kosárhoz adási folyamatot végző logikai fájl.

8. **cart-process-remove.php**

- Termék eltávolítása a kosárból.

9. **cart.php**

- A vásárlói kosár megjelenítése.

10. **checkout-complete.php**

- Rendelési folyamat befejezéséhez tartozó oldal.

11. **checkout.php**

- Pénztár oldal a vásárlás véglegesítéséhez.

12. **delete_category.php**

- Kategóriák törlését kezelő fájl.

13. **delete_product.php**

- Termékek törlésére szolgáló logikai fájl.

14. **edit_category.php**

- Kategóriák szerkesztése.

15. **edit_product.php**

- Termékek szerkesztése.

16. **index.php**

- A webshop kezdőlapja.

17. **login-logic.php**

- Bejelentkezési folyamatot kezelő háttér fájl.

18. **login.php**

- Bejelentkezési oldal.

19. **logout.php**

- Kijelentkezési folyamatot végző fájl.

20. **product.php**

- Egyedi termék megjelenítésére szolgáló oldal.

21. **register-logic.php**

- Regisztrációs folyamatot kezelő háttérfájl.

22. **review.php**

- Termékértékelések megjelenítésére és kezelésére szolgáló fájl.

23. **search-results.php**

- A keresési eredményeket jeleníti meg az oldalon.

24. **shop.php**

- A termékek listája található.

25. **submit-order.php**

- A rendelési folyamat véglegesítését kezelő fájl.

26. **whishlist-process-add.php**

- Hozzáadja a kiválasztott terméket a kívánságlistához.

27. **whishlist-process-remove.php**

- Eltávolítja a kívánságlistáról a már korábban hozzáadott terméket.

28. **wishlist.php**

- A kívánságlista oldalt jeleníti meg.

2.3 Forráskód

Fontosabb tiszta kódnak megfelelő kódsorok, amelyek fontos funkciót látnak el a webshopunk működésében:

2.3.1 *protected_area()*

```
function protected_area(){  
    if(!isset($_SESSION['user'])){  
        alert('warning','Jogosulatlan hozzáférés, a folytatás előtt jelentkezzen be.');        header('Location: login.php');  
        die();  
    }  
}  
protected_area();
```

Ez a funkció biztosítja, hogy csak bejelentkezett felhasználók férhessenek hozzá egy adott oldalhoz. Amennyiben a felhasználó nincs bejelentkezve, figyelmeztető üzenetet kap, majd átirányításra kerül a bejelentkezési oldalra.

Működés:

1. Ellenőrzi, hogy az `$_SESSION['user']` létezik-e.
2. Ha nem létezik:
 - Figyelmeztető üzenetet küld az `alert()` funkcióval.
 - Átirányítja a felhasználót a `login.php` oldalra.
 - A script végrehajtása megszakad a `die()` hívással.

2.3.2 *logout()*

```
function logout(){
    if(isset($_SESSION["user"])){
        unset($_SESSION['user']);
    }
    alert('success','Sikeres kijelentkezés');
    header('Location: login.php');
    die();
}
logout();
```

Ez a funkció a felhasználó kijelentkeztetését végzi el. Ha a felhasználó be van jelentkezve, törli a kapcsolódó munkamenet-adatokat, majd értesíti a sikeres kijelentkezésről, és átirányítja a bejelentkezési oldalra.

Működés:

1. Ellenőrzi, hogy az `$_SESSION['user']` létezik-e.
2. Ha létezik:
 - Törli a munkamenetben tárolt felhasználói adatokat.
3. Értesítést küld az `alert()` funkcióval.
4. Átirányítja a felhasználót a `login.php` oldalra.

2.3.3 *is_logged_in()*

```
function is_logged_in(){
    if(isset($_SESSION["user"])){
        return true;
    } else{
        return false;
    }
}
is_logged_in();
```

Ez a funkció ellenőrzi, hogy a felhasználó be van-e jelentkezve, és ennek megfelelően ad vissza egy logikai értéket.

Működés:

1. Ellenőrzi, hogy az `$_SESSION['user']` létezik-e.
2. Ha létezik, `true`-t ad vissza.
3. Ha nem létezik, `false`-t ad vissza.

2.3.4 *alert(\$type,\$message)*

```
function alert($type,$message){  
    $_SESSION['alert']['type'] = $type;  
    $_SESSION['alert']['message'] = $message;  
}  
alert($type, $message);
```

Ez a funkció egy egyszerű figyelmeztetési rendszer, amely értesítéseket tárol az aktuális munkamenetben. Ezek az értesítések lehetnek figyelmeztetések, hibák vagy sikeres műveletek visszajelzései.

Működés:

1. Beállítja az `$_SESSION['alert']` tömb elemeit:
 - `type`: Az értesítés típusa (pl. warning, success, error).
 - `message`: Az értesítés szövege.
2. Ezek az adatok felhasználhatók a weboldalon értesítési üzenetek megjelenítésére.

2.3.5 *login_user(\$email,\$password)*

```
function login_user($email,$password){  
    global $conn;  
    $sql = "SELECT * FROM users WHERE email = '{$email}'";  
    $res = $conn->query($sql);  
  
    if ($res->num_rows < 1){  
        return false;  
    }  
    $row = $res->fetch_assoc();  
  
    if (!password_verify($password,$row['password'])) {  
        return false;  
    }  
  
    $_SESSION["user"] = $row;  
    return true;}  
}
```

Leírás:

A `login_user` funkció felelős a felhasználók bejelentkezéséért. Ellenőrzi az e-mail-cím és a jelszó helyességét, majd ha az adatok érvényesek, elmenti a felhasználói adatokat a munkamenetbe.

Paraméterek:

- `$email` (string): A felhasználó által megadott e-mail-cím.
- `$password` (string): A felhasználó által megadott jelszó.

Működés:

1. A funkció lekérdezi az adatbázisból az adott e-mailhez tartozó felhasználói adatokat.
2. Ha az e-mail-cím nem található az adatbázisban, a funkció `false` értéket ad vissza.
3. Ha az e-mail létezik, a `password_verify` funkcióval ellenőrzi, hogy a megadott jelszó egyezik-e az adatbázisban tárolt (hash-elt) jelszóval.
4. Ha a jelszó helyes, a felhasználói adatokat elmenti a munkamenetbe (`$_SESSION['user']`), és `true` értéket ad vissza.
5. Ha a jelszó nem egyezik, a funkció `false` értéket ad vissza.

2.3.6 *text_input(\$data)*

```
function text_input($data)
{
    $name = (isset($data['name'])) ? $data['name'] : "";
    $attributes = (isset($data['attributes'])) ? $data['attributes'] : "";

    $value = "";
    $error = "";
    $error_text = "";
    if (isset($_SESSION['form'])) {
        if (isset($_SESSION['form']['value'])) {
            if (isset($_SESSION['form']['value'][$name])) {
                $value = $_SESSION['form']['value'][$name];
            }
        }
    }

    if (isset($_SESSION['form'])) {
        if (isset($_SESSION['form']['error'])) {
            if (isset($_SESSION['form']['error'][$name])) {
                $error = $_SESSION['form']['error'][$name];
                $error_text = '<div class="form-text text-danger">'.$error.'</div>';
            }
        }
    }

    $label = (isset($data['label'])) ? $data['label'] : $name;
    $value = (isset($data['value'])) ? $data['value'] : $value;
    $error = (isset($data['error'])) ? $data['error'] : $error;
    return
    '<label class="form-label text-capitalize" for="'. $name. '">'. $label .'</label>
```



```

        <input name="'. $name.'" value="'. $value.'" class="form-control" type="text" id="'. $name.'"
placeholder="'. $label.'" '. $attributes.'>'
        . $error_text;
    }

```

Leírás:

A `text_input` funkció egy dinamikus űrlapelem-generátor, amely a bemeneti mező HTML kódját készíti el, figyelembe véve a munkamenetben tárolt értékeket és hibaüzeneteket.

Paraméterek:

- `$data` (array): Egy tömb, amely tartalmazza az űrlapelem attribútumait és megjelenítési információit:
 - `name` (string): Az input mező neve (kötelező).
 - `attributes` (string): Az input mező további attribútumai (pl. `required`).
 - `label` (string): A mező címkéje.
 - `value` (string): Az input mező alapértelmezett értéke.
 - `error` (string): Az adott mezőhöz tartozó hibaüzenet (opcionális).

Működés:

1. Ellenőrzi, hogy van-e korábbi érték (`$_SESSION['form']['value']`) a munkamenetben az adott mezőhöz. Ha van, az lesz az alapértelmezett érték.
2. Ha hibaüzenet van társítva a mezőhöz (`$_SESSION['form']['error']`), azt megjeleníti a mező alatt.
3. Létrehozza az input mező HTML kódját a megadott és a munkamenetből származó információk alapján.

2.3.7 *select_input(\$data, \$options)*

```

function select_input($data, $options)
{
    $name = (isset($data['name'])) ? $data['name'] : "";
    $attributes = (isset($data['attributes'])) ? $data['attributes'] : "";

    $value = "";
    $error = "";
    $error_text = "";
    if (isset($_SESSION['form'])){
        if(isset($_SESSION['form']['value'])){
            if(isset($_SESSION['form']['value'][$name])){
                $value = $_SESSION['form']['value'][$name];
            }
        }
    }

    if (isset($_SESSION['form'])){
        if(isset($_SESSION['form']['error'])){
            if(isset($_SESSION['form']['error'][$name])){
                $error = $_SESSION['form']['error'][$name];
                $error_text = '<div class="form-text text-danger">'. $error.'</div>';
            }
        }
    }
}

```

```

    }
  }
}

$label = (isset($data['label'])) ? $data['label'] : $name;
$value = (isset($data['value'])) ? $data['value'] : $value;

$select_options = "";
$selected = "";
foreach ($options as $key => $val) {
    $selected = "";
    if ($key == $value) {
        $selected = "selected";
    }
    $select_options .= '<option value="' . $key . '"' . $selected . '>' . $val . '</option>';
}

$select_tag = '<select name="' . $name . '" class="form-select" id="' . $name . '"
' . $attributes . '>' . $select_options . '</select>';

return
'<label class="form-label text-capitalize" for="' . $name . '">' . $label . '</label>'
.$select_tag
. $error_text;
}

```

A `select_input` funkció egy HTML `<select>` legördülő menü generálására szolgál. Figyelembe veszi az aktuális űrlapadatokat, hibaüzeneteket, valamint az előre definiált opciókat, és dinamikusan hozza létre a menüelemeket.

Paraméterek:

- **\$data (array):** Az űrlapelem attribútumait és megjelenítési információit tartalmazza:
 - `name (string):` A legördülő menü neve.
 - `attributes (string):` További HTML attribútumok (pl. `required`).
 - `label (string):` A mező címkéje.
 - `value (string):` Az alapértelmezett érték.
- **\$options (array):** A legördülő menü opcióit tartalmazó tömb, ahol a kulcs az opció értéke, az érték pedig a megjelenített szöveg.

Működés:

1. Ellenőrzi, hogy van-e korábban kiválasztott érték a munkamenetben (`$_SESSION['form']['value']`).
2. Ellenőrzi, hogy van-e a mezőhöz társított hibaüzenet (`$_SESSION['form']['error']`).
3. Létrehozza az opciók listáját, és az aktuális értékhez beállítja a `selected` attribútumot.
4. Generálja a `<label>` címkét és a `<select>` menüt, valamint megjeleníti a hibaüzenetet, ha van.

2.3.8 *get_category_name(\$category_id)*

```
function get_category_name($category_id) {  
    global $conn; // Az adatbázis kapcsolathoz  
  
    $stmt = $conn->prepare("SELECT name FROM categories WHERE id = ?");  
    $stmt->bind_param("i", $category_id);  
    $stmt->execute();  
    $result = $stmt->get_result();  
  
    return $result->fetch_column(); // Visszaadja a kategória nevét  
}
```

Leírás:

A `get_category_name` funkció egy adott kategória nevét adja vissza az adatbázisból, az adott kategóriaazonosító alapján.

Paraméterek:

- **\$category_id (int):** A kategória egyedi azonosítója az adatbázisban.

Működés:

1. Használja az adatbáziskapcsolatot (\$conn), hogy előkészített SQL lekérdezést futtasson.
2. Paraméterként átadja a kategóriaazonosítót a lekérdezésnek (bind_param).
3. Lefuttatja a lekérdezést, és visszaadja az eredményül kapott kategórianév oszlopot.

Visszatérési érték:

- A kategória neve (string), vagy null, ha az adott azonosító nem található.

2.3.9 *product_item_ui_1(\$pro)*

```
function product_item_ui_1($pro)  
{  
    $thumb = get_product_thumb($pro["photos"]);  
    // Kategória neve lekérdezése  
    $category_name = get_category_name($pro["category_id"]);  
    $format_price = number_format($pro["price"]);  
  
    $str = <<<EOF  
  
    <div class="col-md-4 col-sm-6 px-2 mb-4">  
        <div class="card product-card">  
            <form style="display: flex; justify-content: flex-end;" action="whislist-process-  
add.php" method="post">  
                <input type="hidden" name="id" value="{ $pro['id'] }">  
                <button class="btn-wishlist btn-sm" type="submit" data-bs-toggle="tooltip" data-bs-  
placement="left" title="Hozzáadás a kívánságlistához">  
                    <i class="ci-heart"></i>  
            </form>  
        </div>  
    </div>  
    EOF
```

```

        </button>
    </form>
    <a class="card-img-top d-block overflow-hidden text-center" href="product.php?id=
    {$_pro["id"]}">
        
    </a>
    <div class="card-body py-2">
        <a class="product-meta d-block fs-xs pb-1" href="javascript:;">{$_category_name}</a>
        <h3 class="product-title fs-sm"><a href="product.php?id={$_pro["id"]}">{$_pro["name"]}</a></h3>

        <div class="d-flex justify-content-between">
            <div class="product-price"><span class="text-accent">{$_format_price} Ft</span>
        </div>

        <div class="star-rating">
            <i class="star-rating-icon ci-star-filled active"></i>
            <i class="star-rating-icon ci-star-filled active"></i>
            <i class="star-rating-icon ci-star-filled active"></i>
            <i class="star-rating-icon ci-star-filled active"></i>
            <i class="star-rating-icon ci-star"></i>
        </div>
    </div>
</div>
<div class="card-body card-body-hidden">
    <div class="text-center pb-2">
        <div class="form-check form-option form-check-inline mb-2">
            <input class="form-check-input" type="radio" name="size_{$_pro['id']}"
            id="s_{$_pro['id']}_s">
            <label class="form-option-label" for="s_{$_pro['id']}_s">S</label>
        </div>
        <div class="form-check form-option form-check-inline mb-2">
            <input class="form-check-input" type="radio" name="size_{$_pro['id']}"
            id="m_{$_pro['id']}_m">
            <label class="form-option-label" for="m_{$_pro['id']}_m">M</label>
        </div>
        <div class="form-check form-option form-check-inline mb-2">
            <input class="form-check-input" type="radio" name="size_{$_pro['id']}"
            id="l_{$_pro['id']}_l">
            <label class="form-option-label" for="l_{$_pro['id']}_l">L</label>
        </div>
        <div class="form-check form-option form-check-inline mb-2">
            <input class="form-check-input" type="radio" name="size_{$_pro['id']}"
            id="xl_{$_pro['id']}_xl">
            <label class="form-option-label" for="xl_{$_pro['id']}_xl">XL</label>
        </div>
    </div>
    <form action="cart-process-add.php" method="post">
        <input type="hidden" name="id" value="{$_pro['id']}">
        <button class="btn btn-primary btn-sm d-block w-100 mb-2" type="submit">
            <i class="ci-cart fs-sm me-1"></i>Hozzáadás a kosárhoz
        </button>
    </form>
</div>
<div>
    <hr class="d-sm-none">
</div>
EOF;

```

```
    return $str;
}
```

A `product_item_ui_1` funkció a termék kártya felületének HTML kódját generálja. A kártya tartalmazza a termék képét, nevét, árát, kategóriáját, méretválasztó opcióit, valamint gombokat a kívánságlistára és kosárhoz való hozzáadáshoz.

Bemeneti Paraméterek

- **\$pro (array):**
Egy tömb, amely a termék adatait tartalmazza. A fontos kulcsok:
 - `id`: A termék egyedi azonosítója.
 - `name`: A termék neve.
 - `photos`: A termék képeinek elérhetősége.
 - `category_id`: A termékhez tartozó kategória azonosítója.
 - `price`: A termék ára.

Működés

1. Termékkép lekérése:

A `get_product_thumb` funkció segítségével lekéri a termék fő képének elérési útját.

2. Kategória neve:

A `get_category_name` funkció segítségével lekérdezi a termék kategóriájának nevét az adatbázisból az `category_id` alapján.

3. Ár formázása:

A termék árát a `number_format` funkcióval forintban formázza, hogy olvashatóbb legyen.

4. HTML kód generálása:

Egy Bootstrap-alapú kártyát hoz létre a következő elemekkel:

- **Kép:** A termék képe középre igazítva.
- **Kategória neve:** Link nélkül megjelenítve.
- **Termék neve:** Hivatkozásként megjelenítve, ami a termék részletes adatlapjára vezet.
- **Ár:** A termék ára forintban.
- **Értékelés:** Statikus csillagokkal megjelenítve.
- **Méretválasztás:** Négy méret opció (S, M, L, XL) rádiógombokkal.
- **Gombok:**
 - **Kívánságlista gomb:** A terméket hozzáadja a kívánságlistához.
 - **Kosár gomb:** A terméket hozzáadja a kosárhoz.

Visszatérési Érték

- **HTML kód:**

Egy Bootstrap-alapú HTML kód, amely a termék kártyáját jeleníti meg a webáruházban.

2.4 PHP funkciók

A `functions.php` a webshopunk funkcióinak alapját képezi. Több segédfüggvényt, adatbáziskapcsolatot és képkezelési lehetőséget definiál. Az alábbiakban röviden bemutatjuk a fontosabb részeit:

2.4.1 Alapértelmezett beállítások és kapcsolatok

- **Zebra_Image osztály betöltése:** A `require_once 'Zebra_Image.php';` sor a Zebra Image PHP könyvtárat tölti be, amelyet a képek átméretezéséhez és feldolgozásához használnak.
 - **Session indítása:** Ellenőrzi, hogy fut-e már PHP session, és ha nem, elindítja.
 - **Adatbáziskapcsolat:**
 - Az `mysqli` osztállyal létrehozza az adatbáziskapcsolatot a következő hitelesítő adatokkal:
 - Host: `sql213.infinityfree.com`
 - Felhasználónév: `if0_37627594`
 - Jelszó: `S4ZftgwwR8pKh`
 - Adatbázis: `if0_37627594_shop`
-

2.4.2 Fake termékek generálása

- **`fake_products_generator()`:**
 - Véletlenszerű termékeket generál nevek, árak, leírások és kategóriák alapján.
 - A generált termékeket a `db_insert()` függvénnyel menti az adatbázisba.
-

2.4.3 Termékmenedzsment

- **`get_product($id)`:**
 - Egy adott termék adatait lekéri az adatbázisból azonosító (ID) alapján.
 - Kiegészíti a termék adatait a kategória adataival is.
- **`get_product_photos($json)`:**

- A termékhez tartozó képek adatait dolgozza fel JSON formátumból. Ha nincs kép, egy alapértelmezett képet ad vissza.
 - **get_product_thumb(\$json):**
 - A termék elsődleges képének bélyegképét adja vissza.
-

2.4.4 Adatbázis-kezelő függvények

- **db_select(\$table, \$condition = null):**
 - Az adott táblából lekérdezi az adatokat opcionális feltétellel.
 - Tömbben adja vissza a találatokat.
 - **db_insert(\$table_name, \$data):**
 - Adatok beszúrására szolgál az adatbázis egy adott táblájába.
 - Dinamikusan generálja az SQL beszűrő lekérdezést.
-

2.4.5 Képkezelés

- **create_thumb(\$source, \$target):**
 - A Zebra Image osztállyal átméretez egy képet, megőrizve az arányokat.
 - A célméret alapértelmezetten 1000x1000 pixel.
 - Visszaadja az új kép elérési útját.
 - **get_jpeg_quality(\$size):**
 - Meghatározza a JPEG képminőséget a fájl méret alapján:
 - Nagyobb méret → alacsonyabb minőség
 - Kisebb méret → magasabb minőség
-

2.4.6 Fájlfeltöltés

- **upload_images(\$files):**
 - Több fájl feltöltésére alkalmas.
 - Feldolgozza a fájlokat, ellenőrzi az alapvető attribútumokat (name, type, tmp_name, error, size).
 - Létrehozza a cél elérési útvonalat a fájlokhoz, és bélyegképet generál a create_thumb() függvénnyel.
 - Visszaad egy tömböt, amely tartalmazza a feltöltött fájlok elérési útvonalait és bélyegképeit.
-

2.4.7 URL és hozzáférés kezelés

- **url(\$path):**
 - A megadott relatív útvonalból teljes URL-t készít a `BASE_URL` alapján.
 - **protected_area():**
 - Ellenőrzi, hogy a felhasználó be van-e jelentkezve. Ha nem, átirányítja a bejelentkezési oldalra és figyelmeztetést küld.
 - **logout():**
 - Kijelentkezteti a felhasználót, és átirányítja a bejelentkezési oldalra egy sikeres kijelentkezési üzenettel.
 - **is_logged_in():**
 - Ellenőrzi, hogy a felhasználó be van-e jelentkezve. Igaz/hamis értéket ad vissza.
-

2.4.8 Felhasználói kezelés

- **login_user(\$email, \$password):**
 - Ellenőrzi a felhasználó hitelesítő adatait az adatbázisban.
 - Ha az e-mail cím és a jelszó helyes, a felhasználót bejelentkezteti (session-t hoz létre).
 - Visszatérési érték:
 - `true`: Sikeres bejelentkezés.
 - `false`: Sikertelen bejelentkezés.
 - **alert(\$type, \$message):**
 - Figyelmeztető üzenetet ad hozzá a session adatokhoz (type: üzenet típusa, pl. siker, hiba; message: megjelenítendő üzenet).
-

2.4.9 Űrlapkezelés

- **text_input(\$data):**
 - Szövegbeviteli mező generálása.
 - Dinamikusan beállítható attribútumok, alapértelmezett értékek és hibakezelés.
 - **select_input(\$data, \$options):**
 - Legördülő menü generálása.
 - Beállítja az alapértelmezetten megjelenő értékeket és hibakezelést.
 - **Űrlaphibák kezelése:**
 - Az űrlapokhoz társított hibák (error) és előre kitöltött értékek (value) megjelennek a session adatain keresztül.
-

2.4.10 Adatbáziskezelés

- `get_category_name($category_id)`:
 - Egy kategória nevét adja vissza az azonosítója alapján.
 - Előkészített lekérdezést (prepare) használ, hogy biztonságos legyen a lekérdezés.
-

2.4.11 Termékek megjelenítése

1. `product_item_ui_1($pro)`

- **Funkció:** Dinamikusan létrehoz egy termékkártyát, amely a termék részleteit, kategóriáját, képét, árát és különböző interakciós lehetőségeket tartalmaz.
- **Főbb elemek:**
 1. **Kívánságlista hozzáadás:**
 - A terméket a kívánságlistához adhatja a felhasználó egy POST kéréssel.
 2. **Termékkép és kategória:**
 - A kártya tartalmazza a termékképet (a `get_product_thumb()` használatával generálva) és a kategória nevét (a `get_category_name()` használatával lekérdezve).
 3. **Termék neve és ára:**
 - A terméknév és az ár formázott formában jelenik meg.
 4. **Méretválasztási opciók:**
 - A felhasználó kiválaszthatja a kívánt méretet rádiógombok segítségével (S, M, L, XL).
 5. **Kosárba helyezés:**
 - A termék kosárhoz adható egy POST kérésen keresztül.

2. `product_item_ui_static($pro)`

- **Funkció:** Statikus termékkártyát hoz létre, amely elsősorban egyedi megjelenítést biztosít a kisebb képernyőkre korlátozás nélkül.
 - **Főbb elemek:**
 1. **Termékkép és kategória:**
 - Ugyanúgy megjeleníti a termékképet és a kategória nevét, mint az első függvény.
 2. **Termék neve és ára:**
 - A terméknév és ár reszponzívan jelenik meg, a szokásos módon formázva.
 3. **Csillagos értékelés:**
 - A termék értékelését vizuálisan jelzi a csillagos rendszerrel (Bootstrap ikonok használatával).
 4. **Interakció korlátozása:**
 - Statikus megjelenítés, nem tartalmaz méretválasztási lehetőséget vagy kosárba helyezési gombot.
-

2.5 Adminisztrátorként való regisztráció/bejelentkezés

Alapértelmezett Adminisztrátorként való jelentkezés

A rendszerben egy alapértelmezett adminisztrátori fiók áll rendelkezésre, amely az alábbi hitelesítési adatokkal érhető el:

- **E-mail:** admin@gmail.com
- **Jelszó:** 1234

Ez az adminisztrátori fiók kezdeti tesztelésre és karbantartásra használható.

Adminisztrátorként való regisztráció

Adminisztrátorként kizárólag akkor lehet regisztrálni, ha az e-mail cím tartalmazza az "admin" szót. Például az alábbi e-mail címek elfogadhatók:

- admin45@gmail.com

Amennyiben az e-mail cím nem tartalmazza az "admin" szót, a regisztráció nem lesz érvényes adminisztrátori státuszra, hanem egyszerű felhasználói fiókként kerül rögzítésre.

Funkcionalitás részletei

1. Regisztráció ellenőrzése

- A rendszer a regisztráció során ellenőrzi az e-mail címben az "admin" szó meglétét.
- Amennyiben az "admin" szó hiányzik, a regisztráció sikeresen lezajlik, de az új felhasználó nem adminisztrátori jogosultságot kap.

2. Bejelentkezés folyamata

- Az adminisztrátorok és a normál felhasználók bejelentkezési felülete azonos.
- A rendszer a bejelentkezési adatok (e-mail) alapján különbözteti meg az adminokat a normál felhasználóktól.

2.6 Admin Panel

PureLine Divat Webshop Admin Panel

A **PureLine Divat Webshop** tartalmaz egy admin panelt, amely az adminisztrátorok számára biztosítja a webshop kezeléséhez szükséges funkciókat. Az admin panel kialakítása és átláthatósága lehetővé teszi az adminisztrátor számára a termékek, rendelési adatok és kategóriák hatékony kezelését. Az adminisztrátorok itt kezelhetik a webshop működéséhez kapcsolódó alapvető feladatokat.

Admin panel elérése

Adminisztrátorként kell bejelentkeznünk, majd felül a navigációs részen, jobb oldalt a „**Fiókom**” szövegre kattintva érhetjük el az Admin panelt.

Funkciók:

Írányítópult (Bal oldalon található az oldalsávon):

- Itt találhatóak az admin funkciók, amit csak az adminok látnak.

Ügyfélszolgálat +36 30 355 9753

Ingyenes szállítás 10 000 Ft felett

Rendelés nyomkövetése

HUF / Ft

PureLine

Termék keresése

Q

♡

Szia, Gábor

Fiókom

Kosaram

0.00 Ft

Főoldal

Bolt

Férfiak

Nők

Gyerekek

Admin Panel

Kezdőlap > Fiók > Rendelések története

Gábor Kiss

admin@gmail.com

2

Írányítópult

Termék létrehozása

Termékek

Termékkategória létrehozása

Termékkategóriák

Saját Fiók

Rendelések

Kijelentkezés

Rendelések szűrése: Összes

Kijelentkezés

Rendelési szám	Rendelés dátuma	Állapot	Összesen	Műveletek
5	2024-11-08 11:05:21	Kiszállítva	7,812.00 Ft	Kiszállítva Törölés

Rendelések kezelése az Admin Panelen

Ha belépünk az Admin panelre, vagy rákattintunk a bal oldali oldalsávon található **„Rendelések”** menüpontra, a középső részen megjelenik az összes beérkező rendelés. Ezek állapotát négyféleképpen módosíthatjuk:

1. **Kiszállítva**
2. **Folyamatban**
3. **Késik**
4. **Törölve**

Ezen kívül a rendeléseket véglegesen törölhetjük az adatbázisból a **„Törölés”** gomb segítségével.

Rendelések szűrése:

A **„Rendelések szűrése:”** lehetőségnél ki tudjuk választani, hogy melyik rendelési állapotot jelenítse meg. Ha nem található rendelés, akkor a következő üzenet jelenik meg:

Ügyfélszolgálat +36 30 355 9753

< Ingyenes szállítás 10 000 Ft felett >

Rendelés nyomkövetése

HUF / Ft

PureLine

Termék keresése

Szia, Gábor

Fiókom

Kosaram

0.00 Ft

FőoldalBoltFérfiakNőkGyerekek

Rendeléseim

Kezdőlap > Fiók > Rendelések története

Gábor Kiss

admin@gmail.com

Irányítópult

Termék létrehozása

Termékek

Termékkategória létrehozása

Termékkategóriák

Saját Fiók

Rendelések

Kijelentkezés

Rendelések szűrése: Késik

Kijelentkezés

Nincsenek elemek

Úgy tűnik, még nem adott le rendelést. Fedezze fel termékeinket és kezdje el a vásárlást!

Vásárlás kezdése

1. Termékek kezelése:

- **Termék létrehozása:** Új termékek tudunk hozzáadni a webshopunkhoz a következők kitöltésével:



Gábor Kiss
admin@gmail.com

Irányítópult

Termék létrehozása

Termékek

Termékkategória létrehozása

Termékkategóriák

Saját Fiók

Rendelések

Kijelentkezés

Új termék hozzáadása

Név

Név

Kategória

jukhjh

Beszerzési Ár

Beszerzési ár

Eladási Ár

Eladási ár

Nem

Válaszd ki a nemet

Termékfotó 1

Fájl kiválasztása

Nincs fájl kiválasztva

Termékfotó 2

Fájl kiválasztása

Nincs fájl kiválasztva

Termékfotó 3

Fájl kiválasztása

Nincs fájl kiválasztva

Termékfotó 4

Fájl kiválasztása

Nincs fájl kiválasztva

Termékfotó 5

Fájl kiválasztása

Nincs fájl kiválasztva

Termékfotó 6

Fájl kiválasztása

Nincs fájl kiválasztva

Leírás

BEKÜLDÉS

Termékek: Az összes meglévő termék megtekintése, módosítása vagy törlése, illetve statisztikai adatokat is ezen a felületen láthatjuk:

- A termék árát,
- Az adott termékből értékesített mennyiséget,
- Az adott termékből származó bevételt:

Ügyfélszolgálat +36 30 355 9753

< Ingyenes szállítás 10 000 Ft felett >

Rendelés nyomkövetése

HUF / Ft

PureLine

Termék keresése

♡

Szia, Gábor
Fiókom

0


Kosaram
0.00 Ft

FőoldalBoltFérfiakNőkGyerekek

Termékek

Főoldal > Fiók > Admin Termékek

2



Gábor Kiss
admin@gmail.com

Irányítópult

Termék létrehozása

Termékek

Termékkategória létrehozása


Termékkategóriák

Saját Fiók

Rendelések

Kijelentkezés

Termékeid19



Kényelmes Kötött Overall


37,118.00 Ft

Értékesítések: 5

Bevételek: 185,590.00 Ft

✎

🗑



Virágmintás Maxiruha

26,380.00 Ft

Értékesítések: 0

Bevételek: 0.00 Ft

✎

🗑

Így néz ki, ha módosítani szeretnénk egy terméket (terméknél a kék ikonra kattintva tudjuk ezt megtenni):

Ügyfélszolgálat +36 30 355 9753

< Ingyenes szállítás 10 000 Ft felett >

Rendelés nyomkövetése

HUF / Ft

PureLine

Termék keresése

♡

Szia, Gábor
Fiókom

0


Kosaram
0.00 Ft

FőoldalBoltFérfiakNőkGyerekek

Termékek

Főoldal > Fiók > Admin Termékek

2



Gábor Kiss
admin@gmail.com

Irányítópult

Termék létrehozása

Termékek

Termékkategória létrehozása

Termékkategóriák

Saját Fiók

Rendelések

Kijelentkezés

Termék szerkesztése

Termék neve

Kényelmes Kötött Overall

Ár


Termék fénykép

Fájl kiválasztása

Nincs fájl kiválasztva

Mégsem

Módosítások mentése



Kényelmes Kötött Overall


37,118.00 Ft

Értékesítések: 5

Bevételek: 185,590.00 Ft

✎

🗑



Virágmintás Maxiruha

26,380.00 Ft

Értékesítések: 0

Bevételek: 0.00 Ft

✎

🗑

2. Termékkategóriák kezelése:

- **Termékkategória létrehozása:** Új kategóriák hozzáadása a termékek rendszerezéséhez:

Ügyfélszolgálat +36 30 355 9753

< Ingyenes szállítás 10 000 Ft felett >

Rendelés nyomkövetése

HUF / Ft

PureLine

Termék keresése

Szia, Gábor

Fiókom

Kosaram

0.00 Ft

Főoldal

Bolt

Férfiak

Nők

Gyerekek

Termékkategória létrehozása

Kezdőlap > Fiók > Rendelések kategóriája

2

Gábor Kiss

admin@gmail.com

Irányítópult

Termék létrehozása

Termékek

Termékkategória létrehozása

Termékkategóriák

Saját Fiók

Rendelések

Kijelentkezés

Új kategória hozzáadása

Név

Név

Kategória képe

Fájl kiválasztása

Nincs fájl kiválasztva

Leírás

Küldés

Termékkategóriák: Az összes meglévő kategória megtekintése, módosítása vagy törlése, illetve statisztikai adatokat is ezen a felületen láthatjuk:

- A kategóriák eladások számát,
- Az adott kategóriából származó bevételt.

Ügyfélszolgálat +36 30 355 9753

< Ingyenes szállítás 10 000 Ft felett >

Rendelés nyomkövetése

HUF / Ft

PureLine

Termék keresése

Szia, Gábor

Fiókom

Kosaram

0.00 Ft

Főoldal

Bolt

Férfiak

Nők

Gyerekek

Termékkategóriák

Kezdőlap > Fiók > Termékkategóriák

2

Gábor Kiss

admin@gmail.com

Irányítópult

Termék létrehozása

Termékek

Termékkategória létrehozása


Termékkategóriák

Saját Fiók

Rendelések


Kijelentkezés

Kategóriáid



Kissg

Eladások: 6 Bevételek: 233,383.00 Ft



kjkjh

Eladások: 4 Bevételek: 76,927.00 Ft

4. Saját fiók:

Itt tudjuk megnézni „Rendelések” lehetőséget, illetve itt tudunk kijelentkezni a „Kijelentkezés” lehetőséggel.

Ügyfélszolgálat +36 30 355 9753

< Ingyenes szállítás 10 000 Ft felett >

Rendelés nyomkövetése

HUF / Ft

PureLine

Termék keresése

Q

♥

Szia, Gábor
Fiókom

Kosaram
0.00 Ft

Főoldal

Bolt

Férfiak


Nők

Gyerekek

Rendelésem

Kezdőlap > Fiók > Rendelések története

2



Gábor Kiss
admin@gmail.com

Irányítópult

Termék létrehozása

Termékek

Termékkategória létrehozása

Termékkategóriák

Saját Fiók

Rendelések

Kijelentkezés

Rendelések szűrése: Összes

Kijelentkezés

Rendelési szám	Rendelés dátuma	Állapot	Összesen	Műveletek
2	2024-10-30 11:36:11	Folyamatban	126,626.00 Ft	Folyamatban Törlés
3	2024-10-30 17:00:12	Folyamatban	2,894.00 Ft	Folyamatban Törlés
4	2024-11-10 14:48:22	Késik	50,687.00 Ft	Késik Törlés
6	2024-12-04 05:46:35	Késik	26,955.00 Ft	Késik Törlés
7	2024-12-04 06:30:01	Folyamatban	185,590.00 Ft	Folyamatban Törlés

2.7 Jövőbeli Fejlesztési Lehetőségek

A PureLine webshop fejlesztése közben számos ötlet jutott eszünkbe, amelyekkel a jövőben továbbfejleszthetjük az oldalt, hogy még élvezetesebb és felhasználóbarátabb legyen:

- Továbbfejlesztett keresőmotor:** Kategóriák és szűrők bevezetése, például szín, méret alapján.
- Hűségprogram:** Pontgyűjtő rendszerek és exkluzív kedvezmények a visszatérő vásárlók számára.
- E-mail funkciók bevezetése:** Az alapvető e-mail támogatás megvalósítása, például jelszó-visszaállítási lehetőség és rendelés-visszaigazolások küldése, amely később kiterjeszthető kosárelhagyási értesítésekre és szezonális ajánlatokra is.
- Dinamikus termékértékelés és vélemények:** Lehetőség a vásárlók számára, hogy véleményüket közvetlenül a termékoldalon hagyják, és ezek valós időben megjelenjenek az oldalon.
- Többnyelvű támogatás:** A webshop különböző nyelveken való elérhetősége a nemzetközi piacok elérése érdekében.
- Többféle fizetési opció támogatása:** Bankkártyás fizetés, utánvét, digitális pénztárcák (pl. PayPal, Revolut).
- Szállítási opciók bővítése:** Gyorsított kézbesítés, személyes átvétel, csomagautomata szolgáltatások.
- Csomag nyomon követése:** Integráció futárszolgálatokkal, hogy a vásárlók valós időben láthassák rendelésük állapotát.
- Egyszerű regisztráció és belépés:** E-mail, Facebook vagy Google-fiókkal.
- Kategóriák és alkategóriák bővítése:** Könnyebb navigáció érdekében.
- Raktárkészlet valós idejű megjelenítése:** Termékek elérhetőségének pontos állapota.

12. **Méretválasztási lehetőség:** Olyan funkció bevezetése, amely lehetővé teszi a vásárlók számára, hogy a termékekhez elérhető méretek közül könnyen kiválasszák a számukra megfelelőt a termékoldalon.

2.8 Felmerült Kihívások

A PureLine webshop fejlesztése során rengeteg kihívással kellett szembenéznünk, amelyek rengeteg időt emésztettek fel. A kihívások megoldása nagyon fontos volt a projekt előrehaladása szempontjából. Az alábbiakban részletezzük a legjelentősebb problémákat:

1. Felhasználói élmény optimalizálása

- **Probléma:** Az oldal mobilbarát kialakítása és reszponzív megjelenése kezdetben nem volt tökéletes, ami negatívan befolyásolta a mobilos böngészést.
- **Megoldás:** A Bootstrap keretrendszer használatával sikerült reszponzívvá alakítanunk a design-t.

2. Bejelentkezés és regisztráció

- **Probléma:** Kezdetben a felhasználói bejelentkezési és regisztrációs folyamat nem volt kellően biztonságos és felhasználóbarát, ami a vásárlók számára kényelmetlenséget okozott.
- **Megoldás:** A bejelentkezési és regisztrációs rendszert biztonságosabbá tettük titkosított jelszókezeléssel és egyszerűsítettük a felhasználói élményt.

3. Adatbázis kezelése

- **Probléma:** Az adatbázis kezdetben nem volt optimálisan felépítve, ami lassította az oldal betöltési idejét és a keresési funkciókat.
- **Megoldás:** Az adatbázis struktúráját optimalizáltuk, így gyorsabb adatlekérést és jobb teljesítményt érhattünk el.

4. Kosár és vásárlási folyamat

- **Probléma:** A vásárlási folyamat nem volt kellően gördülékeny, és a kosár nem frissült automatikusan.
- **Megoldás:** A kosár funkció dinamikus frissítésével és a vásárlási folyamat egyszerűsítésével sikerült megoldani. A vásárlók most már könnyedén módosíthatják kosarukat, és gyorsan átválthatnak a pénztárhoz.

3.9 Teszt Dokumentáció
