

Webshop Informatikai Cikkek Árusítására

Ilyó-Kovács Levente

NLFUA8

Oktató: Dr. Bednarik László

Tartalomjegyzék

A feladat leírása.....	2
Struktúra szerkezete	2
A felhasználói felület bemutatása	3
Befejezés	6

A feladat leírása

A projekt egy online webshop, amelynek célja informatikai termékek, mint például számítógépes alkatrészek, kiegészítők és perifériák értékesítése. Az alábbiakban részletesen kitérek a webshop három fontos aspektusára: a célja, a termékek adatkezelése, és a frissítési folyamat.

A webshop célja:

A webshop elsődleges célja az, hogy egy könnyen kezelhető, felhasználóbarát online platformot nyújtson, ahol a vásárlók hozzáférhetnek a legfrissebb informatikai cikkek széles választékához. A webshop lehetővé teszi a felhasználók számára, hogy kényelmesen böngésszenek, keressenek és vásároljanak termékeket otthonukból. Az ilyen típusú webshopok célja, hogy megkönnyítsék az informatikai termékek piacának hozzáférhetőségét, miközben javítják a vásárlási élményt a kényelmes és gyors tranzakciókkal.

A termékek adatainak kezelése:

A webshop a termékek adatait egy központi JSON fájlból olvassa ki, amely egy külső szerveren van tárolva, ebben az esetben a render.com szolgáltatását használja. A JSON (JavaScript Object Notation) egy könnyen olvasható adatformátum, amely ideális adatcsere-re webes alkalmazások és szerverek között. Ez a megoldás lehetővé teszi, hogy a termékatok könnyen frissíthetőek és skálázhatóak legyenek. Az adatok központi tárolása azt is jelenti, hogy bármilyen változás azonnal érvényesül minden felhasználó számára, aki a webshopot használja, így biztosítva az adatok konzisztenciáját és naprakészségét.

Frissítési folyamat:

A termékek adatainak frissítése automatizált módon történik. Amikor a webshop kezelői új terméket adnak hozzá vagy módosítják a meglévő termék információt, ezeket a változásokat GitHubon keresztül commitálják. A GitHub egy verziókezelő rendszer, amelyet széles körben használnak szoftverfejlesztésre. A commitok után egy automatizált CI/CD (Continuous Integration/Continuous Deployment) pipeline fut le, ami a változtatásokat automatikusan alkalmazza, és frissíti azt a render.com-on hostolt szerveren. Ez a folyamat biztosítja, hogy a webshop termékkínálata minden naprakész legyen, és a vásárlók az új termékeket is azonnal megtekinthessék és megvásárolhassák.

Ez a megközelítés nemcsak hatékonyabbá teszi a webshop működését, hanem növeli a felhasználói elégedettséget is, mivel a vásárlók minden a legfrissebb termékinformációkhöz férhetnek hozzá. Az automatizált frissítési folyamat csökkenti az emberi hiba lehetőségét, és gyorsabbá teszi az új termékek piacra dobását.

Struktúra szerkezete

A projekt mappaszerkezete a következő:

- css (mappa)
 - style.css
- html (mappa)
 - aboutus.html
 - buttonSlider.html
 - cart.html
 - company.json
 - contact.html
 - index.html
 - submit.php

- images (mappa)
 - about (almappa)
 - 1.mp4
 - aboutus.webp
 - homepage (almappa)
 - banner.jpg
 - components.webp
 - computeraccessories.webp
 - computers.webp
 - laptops.webp
 - printers.webp
 - smartphones.webp
 - pay (almappa)
 - app.jpg
 - pay.png
 - play.jpg
 - logo.png
 - mesmall.jpg
 - smalllogo.png
- js (mappa)
 - cart.js
 - script.js

A felhasználói felület bemutatása

Menük és oldalak

A webshopban a következő főbb oldalak és menük találhatók:

- **Főoldal (index.html):** Itt jelennek meg a kiemelt termékek és akciók. A felhasználók itt találhatnak gyors hivatkozásokat a legnépszerűbb vagy újonnan hozzáadott termékekre.
- **Termékek oldala (buttonSlider.html stb.):** Külön oldalakon jelennek meg a különböző kategóriákban szereplő termékek, például hardver vagy szoftver kategóriák.
- **Kosár (cart.html):** A felhasználók itt láthatják a kosárba helyezett termékeket, módosíthatják a mennyiséget vagy eltávolíthatják a termékeket.
- **Kapcsolat (contact.html):** Információkat nyújt a cég elérhetőségéről, és lehetőséget biztosít a felhasználóknak, hogy üzenetet hagyjanak.
- **Rólunk (aboutus.html):** A cég rövid leírásáról ad információt.

Felhasználói interakciók

A felhasználói interakciók közé tartozik:

- **Termékekre kattintás:** A felhasználó a termékre kattintva részletesebb termékinformációkat kaphat.
- **Termék hozzáadása a kosárhoz:** A termékek oldalán minden termék mellett egy "Kosárba" gomb található, amelyre kattintva a termék hozzáadódik a vásárló kosarához.
- **Contact us form kitöltés:** Email cím validált form kitöltés lehetősége mely AJAX requestet küld egy submit.php oldalnak.
- **Videó lejátszás és kontrolálás:** Az about us page-en található videó manipulálása.

JavaScript funkcionálisok:

- **Form validáció JavaScript segítségével:** A kapcsolat és vásárlási oldalakon a formok kitöltésekor a JavaScript azonnal ellenőrzi a beírt adatokat, és hibás kitöltés esetén megjeleníti a hibaüzenetet.
- **Kosár műveletek kezelése:** A kosár oldalon a JavaScript figyeli a "Mennyiségi módosítása" és "Eltávolítás" gombok eseményeit, így a felhasználók azonnal láthatják a változásokat a kosár összegében és tartalmában.
- **AJAX a kontaktinformációk dinamikus betöltésére:** Az oldalak JavaScript segítségével AJAX kéréseket indítanak a JSON fájlhoz, amely a kontaktadatokat tartalmazza. A megjelenő konzol hibaüzenet a böngésző CORS policyra vonatkozó házirendjének köszönhető, amely korlátozza, hogy az egyik forrásból betöltött dokumentum vagy szkript hogyan léphet kapcsolatba egy másik forrásból származó erőforrásokkal. Az mi esetünkben, mivel a HTML-fájlt közvetlenül a fájlrendszeréből nyitjuk meg (a file:/// protokoll használatával), a helyi JSON-fájlhoz intézett AJAX-kérelem blokkolva van.

Ezek a funkcionálisok biztosítják, hogy a webshop felhasználóbarát, gyors és interaktív legyen, miközben az adatokat hatékonyan és biztonságosan kezeli.

Példa kódok:

```
function fetchData() {
    showLoading();

    let products = [];

    let data = localStorage.getItem("productData");
    fetch("https://jsonserver-ihm9.onrender.com/products/search?q=" + data)
        .then((res) => res.json())
        .then(console.log);

    fetch("https://jsonserver-ihm9.onrender.com/products/search?q=" + data)
        .then((res) => res.json())
        .then((data) => {
            console.log("Array :" + Array.isArray(data));
            products1 = data.products.map((product) => {
                return {
                    category: product.category,
                    discount: product.discountPercentage,
                    image: product.images[0],
                    price: product.price,
                    id: product.id,
                    title: product.title,
                    description: product.description,
                };
            });
        })
        .then(function (data) {
            hideLoading();
            showProductData(data);
        })
        .catch(function (error) {
            hideLoading();
            showError(error.message);
        });
}
```

A termékek lekérése a JSON szerverről.

```

function addToCart(data) {
    showAlert("Added to cart successfully!");

    cartItems.push(data);
    console.log("addToCart " + data);
    updateCart();
    saveCartItems();
}

```

Termék hozzáadása a kosárhoz.

```

function updateCart() {
    var cartItemsList = document.getElementById("cart-items");
    cartItemsList.innerHTML = "";
    var total = 0;

    for (var i = 0; i < cartItems.length; i++) {
        (function(index) {
            fetch("https://jsonserver-ihm9.onrender.com/products/" + cartItems[index].id)
                .then((res) => res.json())
                .then((data) => {
                    var listItem = document.createElement("li");
                    listItem.classList.add("cart-item");

                    var productImage = document.createElement("img");
                    productImage.src = data.images[0];
                    productImage.alt = data.title;
                    listItem.appendChild(productImage);

                    var detailsContainer = document.createElement("div");
                    detailsContainer.classList.add("cart-item-details");

                    var productName = document.createElement("h4");
                    productName.textContent = data.title;
                    detailsContainer.appendChild(productName);

                    var productPrice = document.createElement("span");
                    productPrice.textContent = "$" + data.price + ".00";
                    detailsContainer.appendChild(productPrice);
                    total = total + data.price;
                    console.log("total: " + total);
                    var removeButton = document.createElement("button");
                    removeButton.innerHTML = "Remove";
                    removeButton.addEventListener("click", function () {
                        removeFromCart(index);
                    });
                    document.getElementById("cart-total").innerHTML =
                        "Total: $" + total + ".00";
                    detailsContainer.appendChild(removeButton);

                    listItem.appendChild(detailsContainer);
                    cartItemsList.appendChild(listItem);
                })
        })(i);
    }
    if (cartItems.length == 0) {
        document.getElementById("cart-total").innerHTML = "Cart is Empty!";
    }
}

```

Kosár tartalmának frissítése.

```
<script>
| var video = document.getElementById("myVideo");
|
| function playVideo() {
| | video.play();
| }
|
| function pauseVideo() {
| | video.pause();
| }
|
| function stopVideo() {
| | video.pause();
| | video.currentTime = 0; // Reset the video to start
| }
|
| function speedUp() {
| | video.playbackRate += 0.5;
| }
|
| function slowDown() {
| | video.playbackRate -= 0.5;
| }
|
| function resetSpeed() {
| | video.playbackRate = 1;
| }
</script>
```

Videó manipulálása

Befejezés

A projekt során létrehoztunk egy webshopot informatikai cikkek értékesítésére, amelyet modern webtechnológiák felhasználásával implementáltunk. A fejlesztés során különös figyelmet fordítottunk a felhasználói interakcióra, az adatkezelésre és az oldal reszponzivitására. A termékadatok kezelése egy külső JSON fájlon keresztül történik, amelyet egy automatizált rendszer frissít, ha új termékek kerülnek hozzáadásra a GitHub repositoryhoz.