

Bazy Danych

Project

Mykyta Korotych

15.04.2024

WSTĘP

Projekt bazy danych. Utworzony przy użyciu HTML / CSS / BOOTSTRAP / JQuery / PHP. Projekt nazywa się „iApple” – jest to sklep internetowy umożliwiający zamawianie telefonów i akcesoriów do nich.

W tej dokumentacji postaram się pokrótce opisać jak działa moja strona.

STRUKTURA PROJEKTU

Projekt jest podzielony na foldery dla poszczególnych potrzeb.

- **index.php**

W katalogu głównym projektu znajduje się plik index.php, który zawiera wszystkie główne elementy strony. Importuje następujące elementy: header, main, footer. Zawiera także pliki stylów i skrypty JS.

- **404.php**

Strona błędu 404 z podstawowymi znacznikami i przekierowaniem użytkownika na stronę główną.

- **.htaccess**

Potrzebne do przekierowania użytkownika na stronę 404, jeśli adres URL jest nieprawidłowy.

- **_BD-Plan.txt | _TO-DO.txt**

Pliki informacyjne.

- **Folder JSON**

Zawiera plik **iapple_shop.sql** dla importu bazy danych do phpMyAdmin. **product_data.json** zawiera przykład tego, jak wyglądają dane które zawierają zdjęcia.

- **Folder components**

Zawiera folder **pages** i pliki z głównymi częściami strony które są podłączane na stronę.

- **dbconfig.php**

Plik jest służy do przechowywania danych w celu połączenia z bazą danych.

- **Folder css**

Zawiera folder z **bootstrap** i pliki ze stylami css.

- **Folder images**

Ma w sobie ico do story i zdjęciem do banera na stronie głównej.

- **Folder js**

Zawiera folder z **bootstrap** i pliki js ze skryptami do stron.

STRUKTURA GŁÓWNYCH PLIKÓW

Omówię tylko parę plików głównych. Strukturę i co one zawierają.

- **index.php**



W tym miejscu podłączam home.php który jest stroną główną. Element z id="main" jest miejscem gdzie są podłączane wszystkie inne pliki pojedynczo kiedy użytkownik klika na produkty albo przechodzi na inną stronę. Footer ta header są niezmiennie i zawsze są na stronie.

- app.js

Jest to plik służący do podłączania EventListener do wszystkich interaktywnych elementów z którymi użytkownik może oddziaływać.



```
1 // Adding to the favorites list
2 $(".favorite-icon").click(function () {
3     var favoriteProductId = $(this).attr("data-product-id");
4
5     if ($(this).hasClass('in-favorites')) {
6         removeFavorite(favoriteProductId);
7     } else {
8         addToFavorite(favoriteProductId);
9     }
10 });
```

Jest to przykładowy odcinek kodu który przedstawia podłączanie do wszystkich elementów z klasą **favorite-icon** podsłuchiacza na click, otrzymuje id produktu na którym był kliknięty ten element. Później sprawdza czy do tego ten produkt nie był dodany. Jeśli tak, to usuwa z favorites, jeśli nie to dodaje do favorites.



```
1 function addToFavorite(favoriteProductId) {
2     $.ajax({
3         url: 'components/header.php',
4         type: 'POST',
5         data: {
6             favoriteProductId: favoriteProductId
7         },
8         success: function (response) {
9             $("#header").html(response)
10        }
11    })
12 }
```

To jest funkcja do której odwołujemy się przy dodawaniu produktu do listy ulubionych. Tworzymy AJAX zapytanie, które wysyła dane na url 'components/header.php' który jest drogą do pliku. Type = POST dla tego że wysyłamy, a nie przyjmujemy. W data oddajemy zmienne. Przy sukcesie wywołujemy funkcję anonimową która przeładowuje **header**.

- **header.php**

```

1 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
2     if (isset($_POST['favoriteProductId'])) {
3         $dbConnection = mysqli_connect($db_server, $db_user, $db_pass, $db_name);
4
5         $sessionId = $_SESSION['user_id'];
6         // Pobieranie aktualnej tablicy ulubionych z bazy danych
7         $query = "SELECT favorites FROM users WHERE session_id = '$sessionId'";
8         $result = mysqli_query($dbConnection, $query);
9
10        if ($result) {
11            $row = mysqli_fetch_assoc($result);
12            $favorites_json = $row['favorites'];
13
14            // Konwersja ciągu JSON na tablicę PHP
15            $favorites = json_decode($favorites_json, true);
16
17            // Dodanie nowego elementu do tablicy
18            $new_item = $_POST['favoriteProductId'];
19            $favorites[] = $new_item;
20
21            // Konwersja zaktualizowanej tablicy na ciąg JSON
22            $updated_favorites_json = json_encode($favorites);
23
24            // Zaktualizuj wpis w bazie danych o zaktualizowaną wartość ulubionych
25            $update_query = "UPDATE users SET favorites = '$updated_favorites_json' WHERE session_id = '$sessionId'";
26            mysqli_query($dbConnection, $update_query);
27        }
28        $dbConnection->close();
29    }
30 }

```

To jest odcinek kodu który obrabia zapytania **POST**. W danym przypadku sprawdza czy jest to zmienna która zawiera **ID** produktu. Jeśli tak to łączymy się z bazą danych. Później zapisujemy session id użytkownika, i tworzymy zapytania **SQL** w którym wyszukujemy użytkownika za jego session id. Jeśli zapytanie było opracowane poprawnie, przetwarzamy JSON na tablicę, dodajemy do tej tablicy nowy id produktu i kodujemy tablicę z powrotem na JSON. Tworzymy zapytanie SQL w którym aktualizujemy dane tego użytkownika. Po tych wszystkich działaniach zamykamy połączenie z bazą danych.

LINKI

Projekt jest do pobrania na stronie: <https://github.com/NikitaBenzin/BazyDanych-Project>

- <https://chat.openai.com/>
- <https://www.apple.com/pl/>